1997

The Effects of Network Latency on Multimedia Applications

James Beach

College of Saint Benedict/Saint John's University

The Effects Of Network Latency On Multimedia Applications

A THESIS
The Honors Program
College of St. Benedict / St. John's University

In Partial Fulfillment
of the Requirements for the Distinction "All College Honors"
and the Degree Bachelor of Arts
In the Department of Computer Science

by
James Beach
May 1997

THE EFFECTS OF NETWORK LATENCY ON MULTIMEDIA APPLICATIONS

Approved by:

James A Schnepf

Professor of Computer Science

J. Andrew Holey

Assistant Professor of Computer Science

E. R. Diedrich

Professor of Economics

Noreen Herzfeld

Chair, Department of Computer Science

Director, Honors Thesis Program

Director, Honors Program

Contents

1 Introduction

Every user of the Internet has become impatient with a web page or program download at one time or another. Many users wonder why some Internet sites and activities are relatively quick while others are painfully slow. Until recently, the response time of the Internet has not been a critical issue and has only been a factor in a user's tolerance for slow Internet browsing. In as little as one year, however, the Internet's response time, as well as the way in which it responds, has quickly become a matter of major importance.

The response time and data delivery style of the Internet have become huge factors in the relatively new networking area of Internet multimedia applications, such as video conferencing. Currently, Internet sites communicate in a connectionless manner, setting up and destroying communication lines for each individual communication between sites. This style is inconsistent with the needs of multimedia applications, and the Internet needs to be converted to connection-oriented networks, which only set up and destroy communication lines once for the entire duration of communications between Internet sites. In addition to changing the communication style of the Internet, the response time of the Internet needs to be improved. Compared to the theoretical limits for Internet response time, today's Internet is pathetically slow. The main cause of the Internet's slow response time is network latency. Understanding the components of network latency, in order to lessen this latency, and transforming current networks into connection-oriented networks will pave the way for Internet multimedia applications such as video conferencing.

2 Network Latency

Network latency, simply stated, is the amount of time it takes one piece of data to move across the network from one computer to another. If network latency did not exist, as soon as an Internet user clicked on a link to an Internet site, that site would instantaneously be fully downloaded. In a stricter sense, network latency is the sum of the time for each component of network communication to be completed. There are six components generally considered to be central parts of network communication, although some components can be eliminated at the cost of adding others of essentially equal time.

Five of the six main components of network latency are Time on the Wire latency, Controller latency, Control / Data Transfer latency, Vectoring the Interrupt latency and Interrupt Service latency (Thekkath 6). Together, the time required for these five latencies can account for nearly all of the total network latency time. The sixth component of network latency is Propagation latency. This latency includes the time taken for light, representing data, to travel through the actual network cables. This component is listed separately from the other five components for two reasons: (1) Propagation latency is proportional to the total distance the data must travel through the network, causing this time to vary much more than any other latency, and (2) Propagation latency is near the theoretical limits imposed by the speed of light, leaving very little room for improvement, while the other latencies can be greatly improved upon by better hardware and software in the network.

2.1 Time on the Wire Latency

Time on the Wire latency is the amount of time it takes a packet of data, once transmitted, to reach point B from point A. This latency does not include any time taken to disassemble or reassemble the packet, which is a significant amount of time if the network uses ATM (Asynchronous Transfer Mode) to transport data. Also, the time taken for reading in the packet, both from the host machine to the network and from the network to the receiving machine, is not included in this amount of time. This latency only accounts for the time the packet spends on the network between point A and point B, without including the additional time from Propagation latency. A significant factor in the amount of time this latency is responsible for is the number of packet switching devices, such as network hubs and routers, the data must travel through.

The Harvard University Network Device Test Lab conducted benchmark tests in 1995 on several network packet switching devices. One of the major tests performed by Harvard was a latency test. In terms of a network switching device, latency is the time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port. The results of the Harvard latency test show that the lowest latency achievable, by the particular packet switching devices tested, is about 62 μseconds between any two ports (CrossComm). Every network packet switching device encountered by the data will add at least this much latency to the overall network latency.

2.2 Controller Latency and Control / Data Transfer Latency

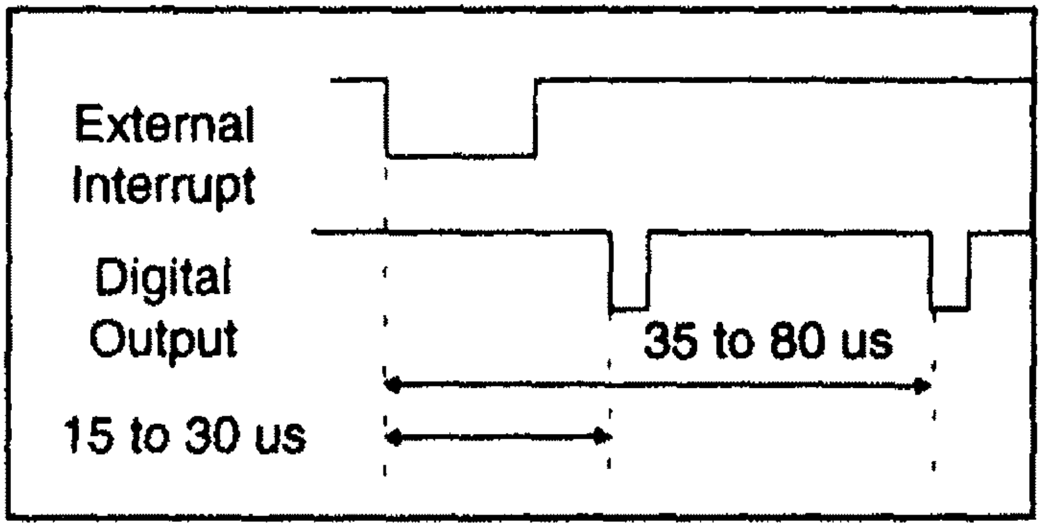
Controller latency is the time taken by the sending controller to start transferring data once the sending host has made the data available to it. Also included in this latency is the time taken by the receiving controller to take the data off the network wire and make it available to the receiving host. This latency is determined by the controller hardware involved. Control / Data Transfer latency includes the time taken for the data to be moved from the host machine's memory to the network over the host machine's data bus. Data must be moved at least once from the host machine's memory to the network, and most likely the data must be moved from the network to the receiving machine's memory, significantly increasing the time involved in this latency. This latency is also closely tied to the hardware used, but improving low level software involved with this latency can lessen the time required. Some controllers are capable of handling this transfer of data from the host machine's memory to the network through DMA (Direct Memory Access). The DMA transfer of data by the controller allows the CPU to have no data transfer overhead, but this latency decrease in data transfer overhead for the CPU is recaptured in Controller latency because the controller must now handle the transferring of data.

2.3 Vectoring the Interrupt Latency and Interrupt Service Latency

Vectoring the Interrupt latency includes the time taken by the receiving host to send a packet arrival message to the interrupt handler in the device driver. How much time this adds to the total network latency depends on the CPU architecture. Interrupt Service latency involves the time relating to the use of an interrupt in the receiving host machine. Host software must perform some essential tasks related to the interrupt before the interrupt can be dismissed (Thekkath 6). These two latencies can best be lessened by improving the way interrupts are acknowledged and dismissed by the operating system and coordinated with the device driver.

A study of Vectoring the Interrupt and Interrupt Service latencies was performed in an experiment named SNAP (Simulator Network Analysis Project) designed to measure dynamic latencies of networks used in simulations. SNAP was developed by the Control Integration and Assessment Branch, which is part of Wright Laboratory at Wright Patterson Air Force Base, Ohio (Bryant). The study was performed using the iRMX for Windows operating system. In the experiment, a signal generator produced a square wave input to an external interrupt on a National I/O board. The rising edge of the square wave was used to produce the interrupt. The interrupt handler would toggle the state of the digital output and the interrupt task would again toggle the state of the digital output. The relationship between the toggling digital output and the signal generator input to the system shows the latency from the interrupt being asserted to the interrupt handler running. This latency is Vectoring the Interrupt latency. The relationship also shows the latency from the interrupt handler running to the interrupt task beginning. Interrupt Service latency is included in this time.

The first falling edge of the digital output, demonstrating Vectoring the Interrupt latency, occurred between 15 μseconds and 30 μseconds after assertion of the interrupt by the signal generator. The second rising edge of the digital output, demonstrating Interrupt Service latency, occurred between 20 μseconds and 50 μseconds after the interrupt handler was running. The test demonstrates a combined upper bound, on the particular equipment tested, for both Vectoring the Interrupt latency and Interrupt Service latency of 80 μseconds (Bryant). The following diagram depicts the relationship between the toggling digital output and the signal generator input to the system.



2.4 Total Component Latency

Many network latencies, excluding Propagation latency, can vary depending on the type of architectures used in the network. An experiment was conducted by Chandramohan A. Thekkath and Henry M. Levy at the University of Washington to measure overall network latency and attempt to separate the total latency time into individual components. Thekkath and Levy attempted to measure overall network latency, and its components, using combinations of hardware platforms (DECSTATION and SPARCSTATION) and different controllers and network types, such as ATM and Ethernet. The experiment involved sending multiple packets of varying sizes across a network and measuring the number of μseconds necessary to complete the transmission of the packets. For the Ethernet and FDDI networks, packets were sent in sizes of 60 bytes and 1,514 bytes. For the ATM network, packets were sent in sizes of 53 bytes and 1,537 bytes. The number of μseconds consumed by each component of network latency was then recorded for each network type, hardware platform and packet size. Some of the results of their experiment are presented in the following table:

Component	**Round-Trip Time (μseconds)**					
	Packet Size in bytes (send / recv)					
	Ethernet (DEC)		**FDDI (DEC)**		**ATM (DEC)**	
	60/60	1514/1514	60/60	1514/1514	53/53	1537/1537
Time on the Wire	115	2442	13	245	5	159
Controller Latency	51	53	97	230	16	161
Control / Data Transfer	40	600	40	253	17	458
Vectoring the Interrupt	25	25	25	25	25	25
Interrupt Service	26	26	92	140	9	20
Sum of Components	257	3146	267	893	72	823

Hardware-Level Round-Trip Packet Exchange Times in μseconds.

The total latency times, i.e. the Sum of Components in the above table, for Ethernet, FDDI and ATM show an important difference between throughput and latency. If low latency for small packets is the goal of the network, then DECSTATION Ethernet should be used to achieve the 257 μsecond round-trip of a 60 byte message. Despite its tenfold bandwidth advantage, FDDI on similar hardware takes 10 μseconds longer to accomplish the same task. Although the ATM network is capable of accomplishing single cell transfers in 72 μseconds, this is an unreasonable lower bound. For the ATM network in this experiment, switching delays and the cost of checking whether the cell is part of a larger message that needs fragmentation and reassembly have been ignored. This omission alters the timing accuracy of small packets more than it alters the timing accuracy of large packets. The software in this experiment needed to fragment and reassemble cells for the ATM network requires 11 μseconds per cell, adding additional latency time of as much as 319 μseconds to the total network latency time.

Higher bandwidth does not make FDDI and ATM faster at transmitting small packets, but it certainly makes FDDI and ATM faster than Ethernet at transmitting large packets. As shown in the above table, both FDDI and ATM can transmit packets of 1,514 bytes and 1,537 bytes, respectively, in less than 1,000 μseconds. Ethernet, however, requires over 3,000 μseconds to transmit a packet of size 1,514 bytes. This trend further improves in favor of FDDI for packet sizes larger than 1,514 bytes (Thekkath 8).

2.5 Propagation Latency

One main factor in overall network latency was not included in the experiment conducted by Thekkath and Levy. Their experiment, due to the extremely short length of network cable used, excluded Propagation latency. This latency is the time taken for light or electricity, representing data, to travel through the actual network cables. The speed of light in optical fiber, or the speed of electricity in copper wire (which is nearly the same), is about 200,000 kilometers per second. The distance from Stanford to Boston is 4,320 kilometers. The amount of time necessary for light to travel from Stanford to Boston in optical fiber is 21.6 milliseconds. The time necessary for a round-trip to Boston and back to Stanford is, therefore, 43.2 milliseconds. This means if a packet is sent from Stanford to Boston, each byte of the packet will take 21.6 milliseconds just to go through the optical fiber. This time is called the transcontinental delay and does not include any other latencies besides Propagation latency (Cheshire).

Furthermore, this transcontinental delay can barely be improved, even if humans were capable of transmitting data at the true speed of light instead of only the speed of light in optical fiber. The speed of light is about 300,000 kilometers per second in a vacuum (compared to 200,000 kilometers per second for light in optical fiber) and produces a delay of 14.4 milliseconds from Stanford to Boston. This is a slight improvement, but attempting to overcome the slowed speed of light in optical fiber is obviously not worth the effort. Currently, the amount of time it takes for a very small message to be sent from Stanford, echoed back by Boston and received again at Stanford is about 85 milliseconds (Cheshire). The process of sending a small message to a network site and having it echoed back to the sending site is called a ping. The ping from Stanford to Boston shows the hardware in the backbone of the Internet can currently achieve speeds within a factor of two of the speed of light in optical fiber.

2.6 Network Latency Experiments

Studying and discussing the six main components of network latency on an individual basis does not necessarily present a complete picture of total network latency times. It is important to isolate each component of network latency and attempt to measure the amount of time required for each component, but it is also important to study how much time is required to actually accomplish tasks over the Internet. A study was conducted to measure the amount of time taken to ping twenty-five different Internet sites on six different continents. Packet sizes of 16 bytes and 128 bytes were sent to the sites. The study was conducted during relatively low Internet usage hours, Monday at 10:00 P.M. and Tuesday at 1:00 A.M., from a Unix workstation at the College of St. Benedict. In all, 25,000 pings were performed and the minimum, maximum and average times for each site and each packet size were recorded. The percentage of packets lost for each site was also recorded. Results of the study are presented in the **Appendix**.

For the twenty-five Internet sites chosen, many in foreign countries, the study shows the average minimum time of a ping is 172 milliseconds, the average maximum time is 489 milliseconds and the overall average time to complete a ping to these sites is 232 milliseconds. Another experiment was conducted by TASC, Inc., using the ping function on a Unix workstation for the same purpose of measuring network latency. The sites chosen by TASC, Inc., were located in North America, significantly shortening the ping distance compared to the previous study. The TASC study revealed an average network latency time of 80 milliseconds and a minimum network latency time of 30 milliseconds (TASC). These two studies show the network latency for the backbone of the Internet using highspeed connections. Not all network connections enjoy the relatively low latency times displayed in these studies.

A typical Ethernet connection, like the ones used in the previous two studies, usually has a latency of about 0.3 milliseconds. A typical ISDN (Integrated Services Digital Network) connection usually has a latency of about 10 milliseconds. A typical modem link, on the other hand, usually has a latency of about 100 milliseconds. This makes the latency of a modem about 300 times worse than the latency of Ethernet. Sending ten characters over the Internet with an Ethernet connection takes less than 2 milliseconds in transmission time, including the latency of the connection. Using a 33 kbit / second modem to send the same ten characters should take 2.4 milliseconds for transmission time (80 bits / 33000 bits per second). The actual transmission time for this modem, however, is 102.4 milliseconds because of the 100 millisecond latency produced by the modem itself. If a large amount of data is sent through this modem, such as 100,000 bytes, then the transmission time is about 25 seconds, and the 100 millisecond latency isn't very noticeable, but if a smaller amount of data is sent, such as 100 bytes, then the latency time is more than the transmission time (Cheshire).

Comparing modems to Ethernet, and the backbone of the Internet, also reveals large shortcomings when the travel time of data is combined with the transmission latency of data. As was shown before, the backbone of the Internet can currently achieve speeds within a factor of two of the speed of light in optical fiber. At the speed of light in optical fiber, the latency to travel 18 kilometers should be 0.1 milliseconds: 18000 / (180 x 10^6 m/s). The true latency, using a modem, is over 100 milliseconds. Modems can currently only achieve speeds which are 1000 times worse than the speed of light in optical fiber, while Ethernet connections are only two times worse (Cheshire).

3 Network Bandwidth and Connection Speeds

In addition to the six main causes of network latency already discussed, an additional factor which affects the total network communication time is the bandwidth of the network. The bandwidth of a network is the volume of information, or data, it can handle in a given unit of time. The bandwidth of a network varies with the type of connection used. A brief overview of connection types, and their data carrying capacity, will serve as a reference point for future discussion. Although more connection types exist, the ten most common connection types used today, or at least speculated to be commonly used when they are available in the future, are presented in the following table:

Connection Type	Data Carrying Capacity
Modem	56 Kbps
ISDN	112 Kbps
T-1	1.5 Mbps
Ethernet	10 Mbps
T-3	45 Mbps
OC-3	155 Mbps
OC-12	622 Mbps
OC-48	2.5 Gbps
OC-192	10 Gbps
OC-768	40 Gbps

Current networks have a possible transmission rate of approximately 2.5 Gigabits per second, or an OC-48 connection possibility . This transmission rate would use only 0.1% of the potential bandwidth, or capacity, of a single-mode optical fiber. Some estimates indicate single-mode optical fibers could carry over three orders of magnitude more than the current amount of 2.5 Gigabits per second (McEachern 70). Other estimates indicate the achievable bandwidth using current fiber technology is in excess of 50,000 Gigabits per second (Tanenbaum 87). With this much bandwidth currently achievable, and a huge increase in bandwidth achievable in the near future, networks have passed from being capacity limited (during the pre-Gigabit networking era) to being latency limited (Kleinrock 38).

Furthermore, a study conducted in 1995 by Jeffrey C. Mogul of Digital Equipment Corporation Western Research Laboratory and Venkata Padmanabhan shows most retrievals over the Internet result in the transmission of relatively small amounts of data. An unscientifically chosen sample of 200,000 HTTP retrievals shows an average size of 12,925 bytes and a median size of only 1,770 bytes. If 12,727 retrievals of zero bytes are excluded from the study, the mean size is still only 13,767 bytes and the median size is 1,946 bytes (Mogul). These retrieval sizes indicate bandwidth-related delay does not account for much of the perceived network latency.

For example, transmission of 20,000 bytes over a T1 (1.544 Mbit / second) connection should take about 100 milliseconds (160000/1544), but as previously stated, simply sending a very small packet from Stanford to Boston takes 85 milliseconds. About 75% of the perceived network latency is completely unrelated to the amount of bandwidth in the network. Therefore, realizing the full potential of network bandwidth beyond the already achievable 2.5 Gbit / second capability, and solving any bandwidth related problems, will not significantly improve the response time of the Internet. The six main components of network latency remain the largest obstacles to acceptable response times for an interactive Internet.

4 Latency Sufficiency for Isochronous Media

Before network multimedia applications became necessary, networks did not need to deliver information within strict time constraints. With applications such as video conferencing, however, information must be delivered in a real-time fashion. To obtain a reasonable approximation of reality, an entity must be able to communicate all necessary information, to all other entities requiring the information, within a time frame such that humans perceive it as continuous. Latency sufficiency specifies the bounds on this time frame in terms of a maximum acceptable delay between host processors connected through a network (TASC). The latency sufficiency for full duplex communication is 500 milliseconds, so any transmission service with an end-to-end latency of greater than 500 milliseconds becomes problematic. For more natural communication to take place, end-to-end latency should not exceed 250 milliseconds ("Implementing Desktop Video").

The latency sufficiency for full duplex communication is important right now for one major reason: video conferencing follows a full duplex, peer-to-peer communications paradigm, much like standard telephony. Video conferencing is an excellent application to represent all multimedia applications because it suffers from nearly all the problems of any other multimedia application, including problems unique to video conferencing. If all problems with video conferencing can be solved, very few problems will remain in the entire realm of multimedia applications across the Internet. Video conferencing involves the transmission of two types of data, audio and video. Both audio and video are isochronous media, meaning they recur at regular intervals. Transmission of isochronous media presents special challenges and places strict limits on total network latency and on network latency jitter (the variation in the amount of latency from one packet transmission to the next). Maintaining low network latency in video conferencing is far more critical than in a one-way multicast for video distribution, e.g. CNN to the desktop. This is due to the fact video conferencing involves interactive human conversation, and not simply the playing back of human conversation ("Video and ATM").

5 Video Conferencing Obstacles

Video conferencing, in its current form, is unacceptable for many reasons. Video conferencing currently relies on full duplex communication, making it the most sensitive to network latencies and packet forwarding delays ("Implementing Desktop Video"). Video conferencing involves substantial digital signals of more than 100 Mbps. This situation is due to the fact all applications involving video start with analog signals roughly 5 MHZ in bandwidth, and these signals must be translated into digital signals much larger in size. As a result of the large digital signals, nearly all networked digital video applications rely on compression to reduce these data rates by factors of 100 to 1 or more ("Implementing Desktop Video").

Although using data compression schemes may seem like a relatively easy solution, changes in network latency can be problematic for certain video decompression algorithms which rely on a steady stream of data at the decoder. In the case of video conferencing, many processes contribute to end-to-end network latency. These processes include video / audio capture and compression, network access and transmission, video / audio decompression and re-synchronization ("Implementing Desktop Video"). With these additional processes adding network latency to the already existing network latency, it becomes very difficult to consistently present a steady stream of data to the decompression algorithm. Also, given the tight constraints on network latency, encoding and decoding video and audio more than once is not practical, so any proposed solutions which attempt to encode and / or decode data more than once will not work satisfactorily.

5.1 Network Delivery Schemes

The largest reason video conferencing is unacceptable right now is derived from the very nature of current networks. Current networks rely on a shared media, packet-based, connectionless delivery scheme to transmit data. A connection between two sites is established, all data, in the form of packets, is transmitted at one time, and the connection is closed. If more audio or video data needs to be transmitted between the same two sites, the entire process must be repeated, including establishing a connection between the two sites. This scheme, however, is completely inconsistent with the isochronous and streaming nature of audio and video. If more routers are added to the network, the situation worsens, due to the queuing nature of these devices, especially as the bandwidth and number of video channels increases ("Video and ATM").

To satisfactorily solve the problems associated with video conferencing, the current concept of networks must evolve toward supporting isochronous media types. One of the main problems with the transmission of media data is its volume: CD-quality sound runs at approximately 200 kbps, and even compressed full-motion video runs at 2 Mbytes per second or more. A general problem is the number of "standards to choose from," with most of them attempting to be general-purpose. None of the "standards" are particularly effective for any specific type of data (NGNM). Video conferencing can be improved if the standards involved in its development are targeted at a specific type of data.

5.2 Varying Network Jitter

In sending media data over a network, any variation in the speed of its transmission is perceptible to the user, unless the data is buffered in a memory region for the amount of time corresponding to the network latency jitter. The buffering of media data can be accomplished in two main ways. One method gives a quick response using a small buffer, but a larger possibility of network jitter exists. The other method ensures no perceptible network jitter using a large buffer, but the response time will be longer. The network must have a way to decide where in the network jitter spectrum it will be operating, the necessary response time and, therefore, how large of a buffer to use. One way to support this decision process is to implement multi-rate interfacing, where the system software must "sniff" the client and automatically, and transparently to the user, adapt to a wide range of different user equipment, such as various kinds of PCS. The system software must also adapt to various speeds of users' connections, ranging from T-3 speed and above, through ISDN and fast modem speeds, to even supporting low-speed connections (NGNM).

5.3 Data Compression

Another problem currently being solved is the extensive amount of bandwidth required by multimedia applications. Uncompressed video streams are unreasonably large in terms of bandwidth required, so alternatives to transmitting uncompressed video over the Internet are being implemented. The alternatives to transmitting uncompressed video are currently moving towards the use of data compression techniques. The current bandwidth of networks makes data compression necessary right now, and even though bandwidth is dramatically increasing, the huge amounts of uncompressed data transmitted in the future may very well be more than the available bandwidth for each user at the exact time a user wishes to transmit the data. Also, the complexity and detail of transmissions will increase as bandwidth increases, so the size of this data in an uncompressed form will continue to be a problem. As these data compression techniques improve and become standard, their continued use will be necessary to utilize available bandwidth in the most efficient way possible.

A codec (short for compressor / decompressor) is a software or hardware mechanism or algorithm which compresses and / or plays back digital video. Each codec has an associated data format. For example, in the CD-ROM multimedia industry, Apple's QuickTime is usually used with the Radius CINEPAK codec. For many codecs, compression takes longer than playback. For prerecorded video, this asymmetric compression / decompression is a positive characteristic of codecs because smooth real-time playback is the goal, and the video can be compressed, with no time constraints, long before playback. For video conferencing, on the other hand, a need for real-time compression exists. Currently, real-time compression is accomplish with some loss of picture quality, but as hardware gets faster, lengthy compression ratios will be reduced, and the picture quality can be improved.

The most common codecs used are MBone Multicast Services, MPEG, CINEPAK and Indeo 3.2, RealAudio from Progressive Networks, VDOLive from VDOnet and CU-SeeMe. The individual characteristics of each codec are not significant; the goal of codecs in general is to balance smooth playback (no dropped frames and no synchronization problems) and the best screen resolution for a given window size. Another goal is to achieve the best linear video playback at the lowest data rate. This goal usually involves playback from a CD-ROM, but at the development time of many codecs, single-speed CD-ROMs were dominate. These CD-ROMs provided at most 150 kbps, so digital video playback was usually optimized for 105 to 125 kbps. It may seem reasonable to upgrade the optimization of digital video playback to current CD-ROM capabilities, but most Internet connections have much lower bandwidth than even single-speed CD-ROMs (Fetik). As improved Internet connections become widespread, an upgrade of digital video playback speeds may take place. The most important codecs for the development of video conferencing style multimedia applications are the codecs which implement a data streaming system consistent with the streaming nature of isochronous media. VDOLive from VDOnet is a video streaming system and RealAudio from Progressive Networks is an audio streaming system.

6 Video Conferencing Solutions

The solutions for the obstacles to acceptable video conferencing show the direction the entire Internet must go. Attempting to leave the majority of underlying Internet structures intact will lead to the failure of integrating multimedia applications into the Internet. The Internet itself must undergo intense changes, as well as new network protocols and software being developed to take advantage of the improved Internet. The example shown by these solutions needs to be followed with more dedicated attempts at changing the Internet.

6.1 VDOLive from VDOnet

VDOLive is based on two core technologies: (1) a scalable compression algorithm that can compress video down small enough to run over small bandwidth sections of the Internet, and allows the quality of the video to increase with the size and quality of the connection, and (2) a communications protocol that maintains the integrity of the video as it travels through the Internet. The relative benefit of these combined technologies is VDOLive motion video can be delivered in real time even over a modem connection. VDOLive motion video is able to run in a small window, in real time, at between 10 and 15 frames per second with a 28.8 kbps modem connection (Fetik).

6.2 RealAudio from Progressive Networks

To solve the problems associated with transmitting audio data across the Internet, the RealAudio protocol and the RealAudio client-server architecture were created. A key underlying technology was a new protocol for isochronous data which supported bi-directional communication between clients and servers. This protocol enables RealAudio users to pause, fast forward, rewind and skip to particular tracks or particular sections quickly and reliably. The RealAudio system and protocol support both the TCP and UDP protocols, but in the vast majority of cases, the results are much better when the audio is delivered using the UDP protocol, very often resulting in a continuous presentation (RealAudio).

Both UDP and TCP protocols accomplish the task of moving packets across a network. UDP is called an unreliable protocol because it does not verify the receipt of a packet at the receiving end. TCP is called a reliable protocol because it requires an acknowledgment for each packet from the receiving end, which relieves the application code from the responsibility for such verification. The choice between the two protocols, however, is more complex than the amount of guaranteed reliability implies.

UDP is very reliable, with nearly 100 percent reliability in intraplatform communications and almost as good over small local networks. Therefore, packet verification is wasted overhead in some applications. Also, network latency is doubled for TCP due to the network latency for transmission and the additional latency during acknowledgment of packet receipt. Furthermore, TCP uses a timeout mechanism to determine when a packet has been lost. If the acknowledgment of packet receipt does not arrive in time, TCP assumes the packet is lost and retransmits it. When the network is heavily used and under severe loads, the possibility of packet loss, due to the acknowledgment taking too long to travel the network, doubles for TCP, even when the packets have been successfully transmitted. This causes TCP to retransmit packets which have successfully reached the receiving end. This unnecessary retransmission during heavy network loads has the double impact of increasing the network load when it is least acceptable and of presenting the receiving software with duplicate information which must be dealt with. The duplicate information is handled by the TCP layer and does not concern the application layer, but this increases system load (NWS).

The RealAudio system, because of the additional network latency added by packet verification under TCP, will use the UDP protocol if it is available. The time saved by not verifying packets is crucial when dealing with the transmission of audio data, and the benefits of the saved time outweigh the costs of not being guaranteed zero packet loss. To get around the problem of occasional packet loss, Progressive Networks developed a sophisticated loss correction system which, in essence, minimizes the impact of any given lost packet. Under this system, the receiving client is able to "recreate" the missing pieces of the audio signal. The system works very well under normal lossless conditions, degrades gracefully when packet loss is in the 2% - 5% range, and even works acceptably when packet loss is as high as 8% - 10%. The RealAudio Server is more efficient at sending audio data than a Web server, sending only as much information as the user needs plus a little extra for buffering purposes. This configuration supports a higher level of overall usage than a Web server alone would provide, enabling support of hundreds, thousands, and soon, even hundreds of thousands of simultaneous listeners (RealAudio).

6.3 Data Transfer Modes

The current state of networks is the largest inhibitor to video conferencing and other interactive multimedia applications realizing their full potential. One solution to this problem is to change the basic way data is transferred by the network. Although this solution may cause more large scale changes to the network structure than other solutions, it is vital if the Internet is ever going to greatly increase in capability. Understanding Synchronous Transfer Mode is necessary to grasp the concept of Asynchronous Transfer Mode, the solution which will change how the Internet transfers data.

6.3.1 Synchronous Transfer Mode (STM)

For the telecommunication backbone networks in use today, Synchronous Transfer Mode (STM) is used to transfer packets of voice and data across long distances. STM is a circuit switched networking mechanism, where a connection is established between two end points before data transfer begins and is torn down when the two end points are completely done. Therefore, the two end points are allocated the entire connection bandwidth for the duration of communications, even when they are not actually transmitting any data.

The way data is transported across an STM network is to divide the bandwidth of the STM link, most commonly a T-1 or T-3 link, into a fundamental unit of transmission called a time-slot or a bucket. These buckets are organized into a train containing a fixed number of buckets, and the train repeats every 125 μseconds, with the buckets always in the same position with the same label. On a given STM link, a connection between two end points is assigned a fixed bucket number on a fixed train, and data from that connection is always carried in that bucket number on the assigned train (Ebrahim).

An analogy to STM would involve a railroad train arriving at the same station once a day. Each car on the train is assigned to one specific person and is reserved for the same person every day. If a person has cargo to ship in his or her personal car on the train, that cargo is shipped in the appropriate car. If a person does not have any cargo to ship, that reserved car for the person with no cargo goes unused, and it cannot be filled by any of the other people waiting to ship cargo from that station. This situation may result in many wasted cars and slower shipping times for the cargo as a whole.

The wasted bandwidth produced by using STM links had not been an issue in the past, due to the relatively low bandwidth requirements of networks at that time. Now, however, telecommunications companies would like to have Gigabit per second networks to carry both real time traffic, such as voice and high resolution video, as well as non real time traffic, such as computer data and file transfers. STM links have problems accomplishing this task because the peak bandwidth requirement for high resolution video is quite high, but the duration for which the data is actually transmitted can be quite small. In other words, the data comes in bursts and must be transmitted at the peak rate of the burst, causing considerable waste of bandwidth if each connection is always reserved enough buckets for its peak bandwidth rate. A better approach would be to use any available buckets for pending connections, instead of needing to reserve buckets for specific connections and having them be empty much of the time. This is the approach taken by ATM.

6.3.2 Asynchronous Transfer Mode (ATM)

STM always identifies a connection by the bucket number the data arrives in, but ATM simply carries the connection identifier along with the data in every bucket. It also keeps the size of the bucket small, so if any one bucket gets dropped along the network path, due to congestion, not too much data will be lost, and in some cases it can easily be recovered. The two end points in an ATM network link are associated with each other by a connection identifier called the Virtual Circuit Identifier (VCI label) instead of by a time-slot or bucket number as in an STM network link. The VCI is carried in the header portion of the ATM packet. The entire ATM packet, or cell, is 53 bytes long. The header is comprised of five bytes, while the payload constitutes 48 bytes of the cell. The header and payload are specified as follows:

```
<------------- 5 bytes ----------------->|<----------------- 48 bytes ----------------->|
-----------------------------------------------------------------------------------------
|  VCI   | control | header checksum | optional adaptation layer |     payload     |
| 24 bits | 8 bits  |      8 bits     |          4 bytes          | 44 or 48 bytes |
-----------------------------------------------------------------------------------------
```

The 48 bytes of payload may optionally contain a 4 byte ATM adaptation layer and 44 bytes of actual data, or all 48 bytes may be data, based on a bit in the control field of the header. This enables fragmentation and reassembly of cells into larger packets.

The control field may also contain a bit to specify whether this is a flow control cell or an ordinary cell, or it may contain an advisory bit to indicate whether or not this cell can be dropped in the face of congestion in the network (Ebrahim) .

ATM improves upon STM by being a scalable, connection-oriented network capable of supporting high bandwidth, isochronous traffic, along with a quality of service guarantee. Once a connection is set up, data traverses the ATM network of switches and interfaces in about 100 μseconds. Since ATM is capable of establishing quality of service for a specific data type, it is possible to maintain a very high quality link between two end points which are passing isochronous video traffic. Furthermore, cell delay variations (network jitter) of 250 μseconds, or less, can be expected for certain classes of traffic. This variation is more than adequate for most networked digital media applications, including video conferencing. It is this combination of high bandwidth with low latency and quality of service that allows for the support of high quality desktop video applications ("Video and ATM").

Another advantage of ATM is in the switches used for moving data. Unlike routers or other shared media devices, ATM switches perform their primary data moving functions in hardware. This is feasible because all cells transmitted through ATM networks are the standard 53 bytes in length. The hardware switching of ATM allows for extremely low latency, even when forwarding LAN (Local Area Network) packets. These forwards can be performed in 50 μseconds, providing a much better solution for client / server and multimedia applications than software solutions can. The ATM switches implement the same function as a router, with the differences being in scalability and much lower latency, even across a large network (Newbridge).

ATM also improves the handling of multiple video streams. Each individual video stream can be handled by different virtual path or virtual channel combinations over the same physical ATM connection. At 155 Mbps, a single ATM connection can handle approximately 50 streams of MPEG-1 video, each running under 2 Mbps (“Implementing Desktop Video”). In addition to this, ATM takes on the role of a virtual router, where traffic is filtered between subnets and through switched virtual circuits. This role allows every point in the network to be one ATM "hop" away, eliminating many, if not all, traditional router bottlenecks which are so detrimental to latency sensitive applications such as video conferencing.

The ATM cells in a connection-oriented link (i.e. cells with the same VCI label) always arrive in order at the destination. This is because no storing and forwarding of cells occurs in the network, cells travel over a single virtual circuit path, the ATM switches do not switch the cells in the same VCI out of order and no retransmission of cells is done at any point in the ATM network. Cells may arrive out of sequence, however, in a connectionless link because multiple virtual circuit paths may be used to transmit all the data. In such a case, additional sequencing must be done in a higher layer service above the ATM datalink layer (Ebrahim).

Even though an ATM network guarantees cells will arrive in order when a connection-oriented link is used, it does not guarantee reliable delivery of the cells. The ATM layer never retransmits cells and no acknowledgments are sent for what has been received. This unreliability has the advantage of making ATM networks much faster, eliminating at least half of the network latency required for reliable delivery. In addition, reliable delivery services can be implemented as a layer on top of the basic connection-oriented ATM layer, where acknowledgment of received data and retransmission of missing data can be done for connections requiring reliable delivery (Ebrahim).

6.4 Integration of Solutions

If ATM networking was fully available today, it would still face some challenges before becoming widespread in use. Three key areas are critical to ATM becoming a mainstream LAN technology.

6.4.1 Integration of Existing LANs

A major obstacle for the widespread acceptance of ATM in mainstream LANs is the integration of existing Ethernet, Token Ring and FDDI network products. Approximately 40 million Ethernet nodes are installed today, and about 10 million new ones are being shipped every year (Newbridge). The installed base of Ethernet will continue to grow for some time, particularly given its very low cost, standardization, and new technology extensions such as switched Ethernet products. It is necessary to allow traditional LAN users to communicate with ATM network devices without having to install any new software or hardware on their workstations. Compatibility with all traditional LAN network devices, such as routers, bridges and LAN hubs, is also a critical need.

6.4.2 Scalability

Current networks are moving towards higher performance, but this does not directly lead to the use of ATM. A logical network migration path includes the interim step of dedicated LAN speeds to the desktop (e.g. switched Ethernet). This technology offers substantially more performance than shared media LANs and is much more cost-effective than direct ATM connections. It also extends the useful life of installed Ethernet adapter cards and media, and costs substantially less than direct ATM connectivity.

ATM switching can, however, provide an effective network backbone technology, even in networks where all end points are standard shared media connections. By allowing network managers to apply the appropriate price / performance option for each user, a smooth migration path is provided. These options include shared media LANs, dedicated or switched LAN segments, and ATM desktop connections ranging in speed from 51 to 155 Mbps on copper twisted pair and 100 to 622 Mbps on multimode fiber (Newbridge).

6.4.3 Improved Network Management

Simplified network management may be the primary reason for many users to migrate to ATM networks, even prior to performance requirements dictating the transition. One of the most significant problems with existing LAN networks is the complexity and administrative effort required for the configuration of large router-based LANs. If the following two management capabilities are addressed in ATM networks, the incorporation of ATM will be even more attractive.

Virtual LANs

By providing tools to allow network managers to view and manage nodes in a logical manner, regardless of physical location or point of connection, ATM can dramatically improve productivity. These same logical groups can be used for other aspects of management, such as billing.

Simplified Moves/Adds/Changes

Network virtualization should enable automatic discovery and configuration of new attachments or other physical network changes. The network should be able to identify the physical address of a new device and associate it with a network layer address based on prior assignment, without human intervention to the system or the end station. This improvement will also dramatically improve productivity, leaving network managers free to accomplish other, more meaningful tasks.

7 Conclusion

Network latency effectively illuminates the many shortcomings of today's networks, especially the error in the connectionless approach to transferring data which is vastly predominate in networks. Some methods of reducing network latency are available now, and more will undoubtedly be available in the future, but simply reducing network latency will only place a bandage over the open wounds networks currently exhibit. An entirely new approach to how networks operate is needed. This approach should result in connection-oriented networks capable of supporting high bandwidth, isochronous traffic, along with quality of service guarantees.

The ATM approach is headed in the right direction, but it needs further refinement and more time for research and implementation. Until ATM is fully implemented, connection-oriented protocols, such as the RealAudio system, will need to fill the void in today's networks. Once ATM is implemented, systems such as RealAudio will be the standard protocol, and they will work in perfect harmony with the ATM network. Any available time and money should be spent researching and implementing groundbreaking technologies, such as ATM and RealAudio, instead of temporarily fixing the superficial flaws in current networks.

APPENDIX

The following table presents the results of a study conducted to measure the amount of time taken to ping twenty-five different Internet sites on six different continents. Packet sizes of 16 bytes and 128 bytes were sent to the sites. The study was conducted Monday at 10:00 P.M. and Tuesday at 1:00 A.M. from a Unix workstation at the College of St. Benedict. Each packet size was sent 500 times to each site. In all, 25,000 pings were performed and the minimum, maximum and average times for each site and each packet size are recorded below. The percentage of packets lost for each site is also recorded. The final row of the table shows the average times and packet loss percentage for all twenty-five sites.

SITE LOCATION	PACKET SIZE (BYTES)	MINIMUM TIME (MS)	MAXIMUM TIME (MS)	AVERAGE TIME (MS)	PACKETS LOST (%)
154.33.66.142 JAPAN	16	201	416	263	6
154.33.66.142 JAPAN	128	221	410	274	9
152.65.165.2 ST. JOHN'S UNIVERSITY	16	2	46	33	0
152.65.165.2 ST. JOHN'S UNIVERSITY	128	3	47	33	0
193.124.133.134 RUSSIA	16	211	356	259	3
193.124.133.134 RUSSIA	128	215	410	265	3
205.214.192.208 BARBADOS	16	170	381	217	5
205.214.192.208 BARBADOS	128	155	555	207	2
196.7.0.150 SOUTH AFRICA	16	320	530	363	15

SITE LOCATION	PACKET SIZE (BYTES)	MINIMUM TIME (MS)	MAXIMUM TIME (MS)	AVERAGE TIME (MS)	PACKETS LOST (%)
196.7.0.150 SOUTH AFRICA	128	330	490	368	11
202.44.200.9 THAILAND	16	254	455	288	1
202.44.200.9 THAILAND	128	265	496	293	0
140.138.151.145 TAIWAN	16	320	650	419	42
140.138.151.145 TAIWAN	128	320	570	417	42
165.21.81.10 SINGAPORE	16	352	540	387	6
165.21.81.10 SINGAPORE	128	355	606	395	13
202.82.40.133 HONG KONG	16	217	581	255	0
202.82.40.133 HONG KONG	128	225	397	263	0
203.21.24.248 NEW ZEALAND	16	322	721	376	6
203.21.24.248 NEW ZEALAND	128	330	515	375	1
203.61.156.174 AUSTRALIA	16	265	755	311	1
203.61.156.174 AUSTRALIA	128	270	545	309	1
205.214.216.20 WEST INDES	16	186	1496	327	2
205.214.216.20 WEST INDES	128	200	1620	758	1
194.90.6.10 ISRAEL	16	411	940	476	0
194.90.6.10 ISRAEL	128	415	629	477	0
192.76.144.75 GERMANY	16	207	450	261	14
192.76.144.75 GERMANY	128	221	494	270	11
134.206.1.72 FRANCE	16	160	481	200	8
134.206.1.72 FRANCE	128	165	381	205	6
193.149.64.145 ENGLAND	16	154	321	195	11
193.149.64.145 ENGLAND	128	157	321	199	10
198.105.232.37 WASHINGTON	16	90	315	121	0
198.105.232.37 WASHINGTON	128	91	255	126	0

SITE LOCATION	PACKET SIZE (BYTES)	MINIMUM TIME (MS)	MAXIMUM TIME (MS)	AVERAGE TIME (MS)	PACKETS LOST (%)
199.86.65.253 ST. CLOUD, MN	16	10	310	40	0
199.86.65.253 ST. CLOUD, MN	128	24	301	44	0
204.70.186.6 CHICAGO, IL	16	26	311	64	0
204.70.186.6 CHICAGO, IL	128	26	181	50	0
204.70.1.233 SAN FRANCISCO, CA	16	70	440	106	0
204.70.1.233 SAN FRANCISCO, CA	128	76	360	110	0
204.70.1.37 DENVER, CO	16	54	340	107	1
204.70.1.37 DENVER, CO	128	54	346	116	4
204.70.3.145 SEATTLE, WA	16	90	321	145	4
204.70.3.145 SEATTLE, WA	128	90	520	146	6
206.157.77.49 ATLANTA, GA	16	53	500	94	2
206.157.77.49 ATLANTA, GA	128	53	387	101	1
204.70.104.65 WILLOW SPRINGS, IL	16	26	375	79	1
204.70.104.65 WILLOW SPRINGS, IL	128	30	730	136	2
204.70.2.1 WASHINGTON, D.C.	16	65	476	140	6
204.70.2.1 WASHINGTON, D.C.	128	74	375	128	2
AVERAGE	N/A	172.02	488.96	231.82	5.18

Works Cited

Ayre, Rick. "Browsing with Java." PC Magazine 12 March 1996: 124.

Bryant, Richard Barry, and Douglass, Capt. D. Scott, and Ewart, Ronald, and Slutz, Gary Jeff. "Dynamic Latency Measurement Using the Simulator Network Analysis Project (SNAP)." <http://www.wl.wpafb.af.mil/flight/fcd/figd/snaprep/snap_all.htm>.

Cheshire, Stuart <cheshire@CS.Stanford.EDU>. "It's the Latency, Stupid." <http://rescomp.stanford.edu/~cheshire/rants/Latency.html>. May, 1996.

CrossComm Corporation <http://www.crosscomm.com>. "CrossComm's Workgroup Switch is Top Performer at Harvard Benchmark Testing." <http://www.crosscomm.com/products/brad.htm>. 1997.

Ebrahim, Zahir. "A Brief Tutorial on ATM." <http://juggler.lanl.gov/lanp/atm.tutorial.html>. March 5, 1992.

Farber, Dan. "Microsoft gives nod to Sun's Java." PC Week 11 Dec. 1995: 138.

Fetik, Richard <fetik@interbiz.com>. "Multimedia X: Multimedia Meets the Internet." <http://www.sigs.com/publications/docs/xspot/9608/xspot9608.d.tren ds.html>.

Flynn, Jim. "How Java makes network-centric computing real." Datamation 1 March 1996: 42.

Hayes, Frank. "Spicing up Java: new products designed to boost enterprise capability." Computerworld 11 Dec. 1995: 79.

Hogan, Mike. "Java stirs up the Web." PC / Computing March 1996: 48.

Kleinrock, Leonard. "IEEE Communications Magazine." The Latency / Bandwidth Tradeoff in Gigabit Networks April 1992: 36-40.

McEachern, James A. "IEEE Communications Magazine." Gigabit Networking on the Public Transmission Network April 1992: 70-78.

Mogul, Jeffrey C., and Padmanabhan, Venkata <padmanab@cs.berkeley.edu>. "Improving HTTP Latency." <http://www.spyglass.com:4040/newtechnology/performance/mogul/HTTPLatency.html>. 1995.

Newbridge Networks Inc. <http://www.newbridge.com>. "ATM Virtual Router Architecture." <http://www.vivid.newbridge.com/documents/vivid-architecture.html>. 1996.

NGNM (Next-Generation Networked Multimedia). "The Next-Generation Networked Multimedia Project." <http://www.create.ucsb.edu/ngnm/ngnm.html>. Sept. 18, 1996.

Northcutt, J. Duane, and Eugene M. Kuerner. "Computer Communications." System support for time-critical applications Oct. 1993: 619-636.

Notess, Greg R. "Java creates new potential for Internet information." Online March-April 1996: 82.

NWS (National Weather Service). "WFO Advanced Preliminary Evaluation of DECmessageQ." <http://www.oso3.nws.noaa.gov/esahome/prc/DMQ.html>. Nov. 14, 1996.

Pfeiffer, Marc P. <webadmin@vivid.newbridge.com>. "Implementing Desktop Video Networks Over ATM LANs." <http://www.vivid.newbridge.com/documents/DesktopVideoNetwork_index.html>. April, 1996.

-----. "Video and ATM: A look at Networked Digital Media in ATM LANs ." <http://www.vivid.newbridge.com/documents/ATM-VideoArticle.html>. 1996.

RealAudio <http://www.realaudio.com>. "White Paper on RealAudio Client-Server Architecture." <http://www.realaudio.com/prognet/openarch/whitepaper.html>. 1996.

Tanenbaum, Andrew S. Computer Networks. 3rd ed. New Jersey: Prentice-Hall, 1996.

TASC, Inc. <http://www.tasc.com>. "A Distributed Interactive Simulation Intranet." <http://www.tasc.com/simweb/papers/disramp/latsuf.htm>. 1996.

Thekkath, Chandramohan A., and Henry M. Levy. Limits to Low-Latency Communication on High-Speed Networks. Seattle, WA: Department of Computer Science and Engineering, University of Washington.