WPI-CS-TR-98-18 July 1998

End-to-End Quality in Multimedia Applications

by

Mark Claypool and John Riedl

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

End-to-End Quality in Multimedia Applications

Mark Claypool
claypool@cs.wpi.edu
Worcester Polytechnic Institute
Computer Science Department

John Riedl
riedl@cs.umn.edu
University of Minnesota
Computer Science Department

August 12, 1998

1 Introduction

The tremendous power and low price of today's computer systems have created the opportunity for exciting applications rich with graphics, audio and video. These new multimedia applications promise to support and even enhance the work we do in teams by allowing users to collaborate across both time and space. In order to live up to their potential, these multimedia applications must meet the performance needs of the users they support.

Thus far, research in multimedia performance has been targeted mostly at the systems-level characteristics of performance. These characteristics, called Quality of Service (QoS), represent the set of quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application. Typical QoS parameters deal with system-level requirements, such as a bound on network delay. Resource reservation schemes are also tied to system-level capacities, such as guaranteed network throughput [ZDE93].

Application performance should strive to meet user-level requirements in addition to system level requirements. Users do not necessarily care about the throughput of the network or delay on network packet arrival times. Instead, users want defined image resolutions and audio clarity, a smooth playout of audio and video, and an upper bound on response time for interactive applications. Kalkbrenner had this in mind when he developed an interface to map user choices into system parameters [Kal94]. Users chose image size, resolution, and color for video and speech of telephone or CD quality for audio. These choices were then mapped into lower-level system parameters. Conversely, a measure of multimedia application performance should reflect how well the user choices are met by the system. From a user's point of view what is needed is a performance metric that accounts for user-level requirements and gives them a measure of end-to-end quality.

"End-to-end" refers to the performance from the source of a multimedia conversation (often a microphone, video codec or disk) to the destination (often a speaker or monitor). An end-to-end measure of quality accounts for all the system components between the source and the destination that affect the performance of the application. For example, the end-to-end quality of a video-on-demand system on the Internet would be determined by the server workstation (codec, disk, bus, processor, ram, etc.), network (protocol, bandwidth, routers, etc.) and client workstation (processor, ram, video card, monitor, etc.). Acceptable performance of one component may result in poor application quality if the performance bottleneck is in another component. For example, determining that a network is supplying the required bits/second for a video display does not guarantee that the user workstation is displaying those frames smoothly, or even that it is displaying them at all. An end-to-end measure of quality must take into into account all system components that affect the media.

Different media place different constraints on computer systems. For instance, human eyes can smooth over occasional glitches in a video stream more readily than human ears can smooth over breaks in an audio stream [Cor97]. Having the computer system place greater emphasis on preserving audio data than on preserving video data might be important for user satisfaction. What is needed is a performance metric that accounts for performance effects from different media.

In addition, the constraints for each type of media can vary from application to application. For example, the acceptable delay for an audio broadcast application such as a radio program may be far more than the acceptable delay for an audioconference. In an audioconference, users require low latencies so that the conversation is as life-like as possible. However, in an audio broadcast program, the users do not interact, allowing a larger delay to go unnoticed. You could imagine a user downloading an entire radio program overnight and then playing it back in the morning. In this case, a delay of over eight hours might be quite acceptable. What is needed is a performance metric that accounts for user perceptions of different media.

Although we often think of a multimedia application as a continuous stream of data, computer systems handle multimedia in discrete events. An event may be receiving an update packet or displaying a rendered video frame on the screen. The quantity and timing of these events give us measures that affect application quality. Based on previous multimedia application research [SW93, AFKN95, Roy94, IKK93, PSR93, RS94, MS94, Fer92, KN82, TJ94, Par94, FM76], we have identified three measures that determine quality for most multimedia applications:

- *Latency.* The time it takes information to move from the server through the client to the user we call latency. Latency decreases the effectiveness of applications by making them less like real-life interaction [Zeb93, IKK93, Roy94, DCJ93].
- *Jitter.* Distributed applications usually run on non-dedicated systems. The underlying networks are often packet-switched and the workstations are often running multiple processes. These non-dedicated systems cause variance in the latency, which we call jitter. Jitter can cause gaps in the playout of a stream such as in an audioconference, or a choppy appearance to a video display [JSTS92, RS94, JVS91].
- *Data Loss.* Any data less than the amount determined by the user requirements we call data loss. Data loss takes many forms such as reduced bits of color, pixel groups, smaller images, dropped frames and lossy compression [AFKN95, OOM95, MS94, SW93]. Data loss may be

done voluntarily by either the client or the server in order to reduce load or to reduce jitter and/or latency. When any component in the end-to-end link does not have sufficient capacity to transmit data at the rate required by the application, data loss occurs. For example, if the network has a maximum bandwidth of 5 Mbps and a videoconference requires 10 Mbps there will be 50% data loss.

The effects of latency on a user's perception of an application is well-understood and well-researched [Zeb93, IKK93, Roy94, DCJ93]. Human perception of latency is around 150 milliseconds. Generally, for interactive applications, latency must be under 500 milliseconds, often as low as 200 milliseconds.

Similarly, there is a clear relationship between data loss and application performance deterioration [AFKN95, OOM95, MS94, SW93]. Audio rates of 64 Kbits/second are acceptable for human voice. CD-quality sound requires around 1.5 Mbits/second. Frame rates of at least 3 frames per second have been found to be minimal for doing effective work. The human eye blends discrete images into motion at about 12 frames per second. Motion picture quality video is about 30 frames per second.

Methods to ameliorate the effects of jitter have been explored by many researchers [SJ95, RS94, Fer92, CHR97]. The tradeoff between buffering and jitter has also been explored [KN82, SR95].

There is no "one-size fits all" method for determining end-to-end multimedia performance. Previous research has defined metrics for components fundamental to most multimedia applications [WS95]. However, while there are fundamental requirements for most multimedia applications, some applications have unique user-level requirements. For example, a video display with audio has a user synchronization requirement for the sound and the display. Multimedia applications that have only audio or only video do not have a synchronization requirement. An end-to-end measure of quality must be flexible enough to incorporate application specific requirements in addition to the fundamental multimedia requirements.

The rest of this chapter is laid out as follows:

In Section 2, we present a performance metric for determining the end-to-end quality of multimedia applications. We describe the fundamental properties of a multimedia quality metric and propose out metric that satisfies these properties.

In Section 3, we present an example detailing the application of our quality metric to videoconferences. We introduce videoconferences, apply our metric and predict videoconference performance under a variety of system configurations, including faster processors and networks, increasing users and increasing system load.

In Section 4, we summarize the conclusions from this chapter. We also describe possibilities for future research including other quality metrics, media scaling and applications with changing requirements.

2 Perceptual Quality of Multimedia

Perceptual quality is a measure of the performance of a multimedia application based on the requirements expected by the user. If the user performance requirements are met, application

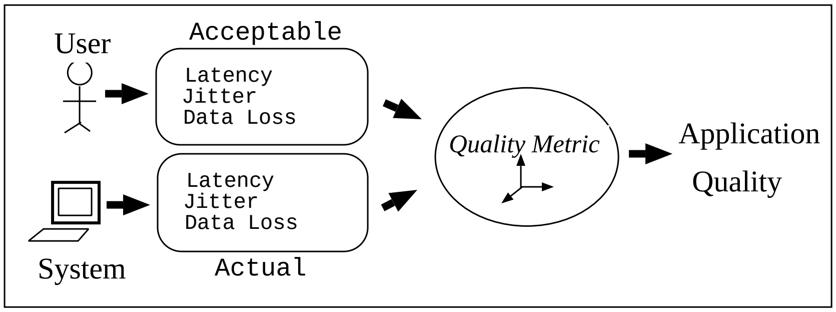

Figure 1: The Process for Computing Application Quality. The user defines the acceptable latency, jitter and data loss and the end-to-end system determines the actual values. Based on the acceptable values specified in the user requirements, a quality metric computes the application quality from the actual values.

quality will be acceptable. If the user performance requirements are not met, application quality will be unacceptable. A multimedia performance metric must account for components fundamental to multimedia applications. As described in Section 1, we have identified three fundamental measures that determine quality for most distributed multimedia applications: latency, jitter and data loss.

Ideally, we would like there to be no latency, jitter or data loss. Unfortunately, on a variable delay network and non-dedicated computer this can never be achieved. To compute the perceptual quality of the multimedia application, we use the above quality components in a process depicted in Figure 1. The user requirements for the application define the acceptable latency, jitter and data loss. The end-to-end system determines the actual latency, jitter and data loss. Acceptable and actual data are fed into a *quality metric* for the application. The quality metric is a function, based on the acceptable components and dependent upon the actual components, that computes the application quality.

In order to quantitatively evaluate application quality, we need a reasonable quality metric.[1] In the mathematical sense, given a space `S` with at least 3 elements `(x,y,z)` a metric is a real function of 2 variables `D(x,y)` such that:

- `D(x,y) >= 0` (non-negative)
- `D(x,y) = 0 iff x=y` (x and y are the same elements)
- `D(x,y) = D(y,x)` (symmetry)
- `D(x,y) + D(y,z) >= D(x,z)` (triangle inequality, which says you cannot gain by going through an intermediate point)

We further define a perceptual quality metric for multimedia as having several other important properties:

- It incorporates the three fundamental multimedia perceptual quality components: latency, jitter and data loss.

[1] A metric is sometimes called a measure of the distance between 2 points in any space.

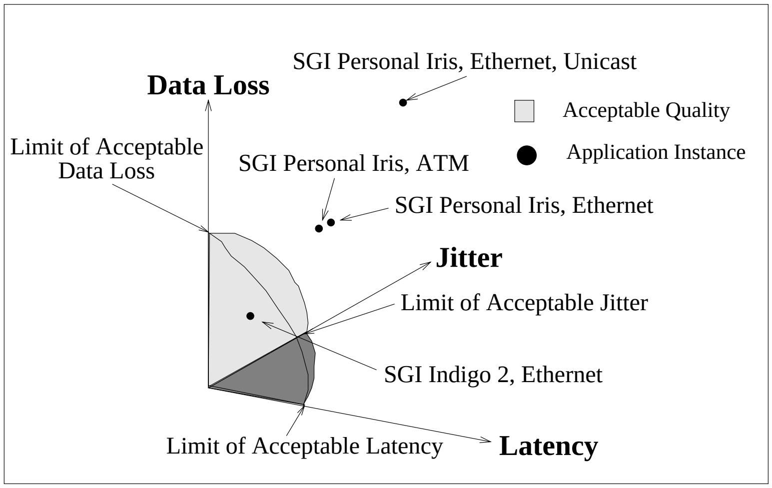

Figure 2: A Perceptual Quality Space for Multimedia Applications. The user defines the acceptable latency, jitter and data loss. These values determine a region of acceptable application quality, depicted by the shaded region. All points inside the shaded region have acceptable quality, while those outside the region do not. An instantiation of the application and the underlying computer system lies at one point in this space. Four application configuration instantiations are shown.

- It treats the fundamental components equally, which seems appropriate in the absence of user studies to the contrary.
- It produces a convex region of acceptable quality. This fits our intuition about changes in quality: the measure increases total quality with any increase in quality along one axis. There are no pockets of unacceptable quality within the acceptable quality region, nor can you move from unacceptable to acceptable by any combinations of increase along the axes.

We propose a quality metric extended from the work of Naylor and Kleinrock [KN82]. Naylor and Kleinrock developed a model for measuring the quality of an audioconference based on the amount of dropped frames and client-side buffering. We extend this model by using each quality component as one axis, creating a multi-dimensional quality space. We place the best quality value for each axis at the origin and normalize each axis so that the user-defined minimum acceptable values have an equal weight. An instantiation of the application lies at one point in this space. The location of the point is determined by our predictions of the amount of latency, jitter an data loss that would occur with the given system configuration. In order to satisfy the mathematical properties of a metric, we compute the application quality by taking the Euclidean distance from the point to the origin. All points inside the region defined by the user-defined minimums have acceptable quality while points outside do not. Figure 2 depicts an example 3-d perceptual quality space for multimedia applications.

Our metric attains all of the mathematical metric properties and multimedia metric properties listed above. There can be many possible quality metrics for a given application. In fact, there may be many quality metrics that agree with a user's perception of the application. Mean opinion score (MOS) testing can be used to determine if a metric agrees with users' perceptions. The MOS is a five-point scale where a MOS of 5 indicates perfect quality and a score of 4 or more represents high quality. MOS has been used extensively in determining the acceptability of coded speech. MOS testing has been beyond the scope of our research, so we cannot be certain our quality metric fits users' perceptions. However, if new metrics are developed and validated with MOS testing, they can be used in place of our quality metric.

One limitation to multimedia perceptual quality metrics is that after scaling, the upper limits on the axes have different characteristics. The "data loss" axis has a finite upper-limit of 100%, while the "latency" and "jitter" axes each have an infinite bound. Comparing application quality for two different configurations at the upper-limit of any of the axes may not match user perception. Fortunately, this limitation usually arises when comparing two very unacceptable configurations. The metric is most valuable for determining whether a configuration provides "acceptable" or "unacceptable" application quality and comparing configurations within the "acceptable" region.

3 An Example: End-to-End Quality for Videoconferences

In this section, we show how our perceptual quality metric for multimedia can be used to study the performance of videoconferences. We can learn a lot from videoconferences. Videoconferences incorporate both audio and video. Interactive videoconferences can have from two to tens of users, while videoconference broadcasts can have hundreds or perhaps even thousands of viewers. In addition, videoconferences are often integrated into larger multimedia applications. Determining end-to-end quality for various system configurations to support videoconferences is valuable for business' wishing to invest in videoconference technology.

In order to apply our quality metric to a videoconference under various system configurations, we must: 1) determine the region of acceptable videoconference quality; 2) determine jitter; 3) determine latency; and 4) determine data loss.

3.1 The Region of Acceptable Videoconference Quality

To determine the region of acceptable videoconference quality, we need to define acceptable limits for videoconferences along each of the latency, jitter and data loss axes. According to Jeffay and Stone, delays of 230 milliseconds or under are acceptable for a videoconference [JSS92]. For data loss, research in remote teleoperator performance has found that task performance is virtually impossible below a threshold of 3 frames per second [MS94]. We use 3 frames per second as the minimum acceptable frame rate.

The presence of jitter often presents an opportunity for a tradeoff among latency and data loss. Buffering, an application-level technique for ameliorating the effects of jitter, can compensate for jitter at the expense of latency. Transmitted frames are buffered in memory by the receiver for a period of time. Then, the receiver plays out each frame with a constant latency, achieving a steady stream. If the buffer is made sufficiently large so that it can hold all arriving data for a period of time as long as the tardiest frame, then the user receives a complete, steady stream. However, the added latency from buffering can be disturbing [Par94], so minimizing the amount of delay compensation is desirable.

Another buffering technique to compensate for jitter is to discard any late frame at the expense of data loss. Discarding frames causes a temporal gap in the play-out of the stream. Discarding frames can keep play-out latency low and constant, but as little as 6% gaps in the playout stream can also be disturbing [KN82]. In the case of audio speech, the listener would experience an annoying pause during this period. In the case of video, the viewer would see the frozen image of the most recently delivered frame.

Naylor and Kleinrock describe two policies that make use of these buffering techniques: the E-Policy (for Expanded time) and the I-Policy (for late data Ignored) [KN82]. Under the E-policy, frames are never dropped. Under the I-policy, frames later than a given amount are dropped. Since it has been observed that using a strict E-Policy tends to cause the playout latency to grow excessively and that dropping frames occasionally is tolerable [CSZ92, SJ95], we use the I-Policy as a means of examining needed jitter compensation for a multimedia stream.

The I-policy leads to a useful way to view the effects of jitter on a multimedia stream. Figure 3 depicts the tradeoff between dropped frames and buffering as a result of jitter. We generated the graph by first recording a trace of video frame interarrival times. We then fixed a delay buffer for the receiver and computed the percentage of frames that would be dropped. This represents one point in the graph. We repeated this computation with buffers ranging from 0 to 250 milliseconds to generate the curved line. The graph can be read in two ways. In the first, we choose a tolerable amount of dropped frames (the horizontal axis), then follow that point up to the line to determine how many milliseconds of buffering are required. In the second, we choose a fixed buffer size (the vertical axis), then follow that point over to the line to determine what percent of frames are dropped. In Figure 3, if we wish to restrict the amount of buffering to 100 milliseconds, then we must drop about 2% of the frames since that is how many will be more than 100 milliseconds late, on average. For an 2 Mbps video stream consisting of 33 6-Kbyte frames per second, this equates to dropping one frame every 1.5 seconds. On the other hand, if we wish to not drop any frames, we have to buffer for over 200 milliseconds.

3.2 Determining Jitter

Previous experiments measuring the effectiveness of several jitter reduction techniques give us the relationship between load and jitter for faster processors and networks [CHR97]. We also know the reduction in jitter due to real-time operating system priorities [CHR97]. We use these results as the basis for determining the jitter in the videoconference under various system configurations.

3.3 Determining Latency

We can predict the amount of latency from the jitter compensation buffer by using predictions on the amount of jitter. In addition to the buffering latency, there is the additional latency from the sender processing, the network transmitting and the receiver processing. In previous experiments, we measured the latency from recording and playing video [CR96]. From other previous experiments, we measured the latency attributed to sending and receiving packets [CR94]. We can compute the latency from the network based on the frame size and network bandwidth. To predict the total latency, we add the latencies from: recording the video frame; sending the video frame to the client; receiving the video frame from the receiver; buffering in the jitter compensation curve; and playing

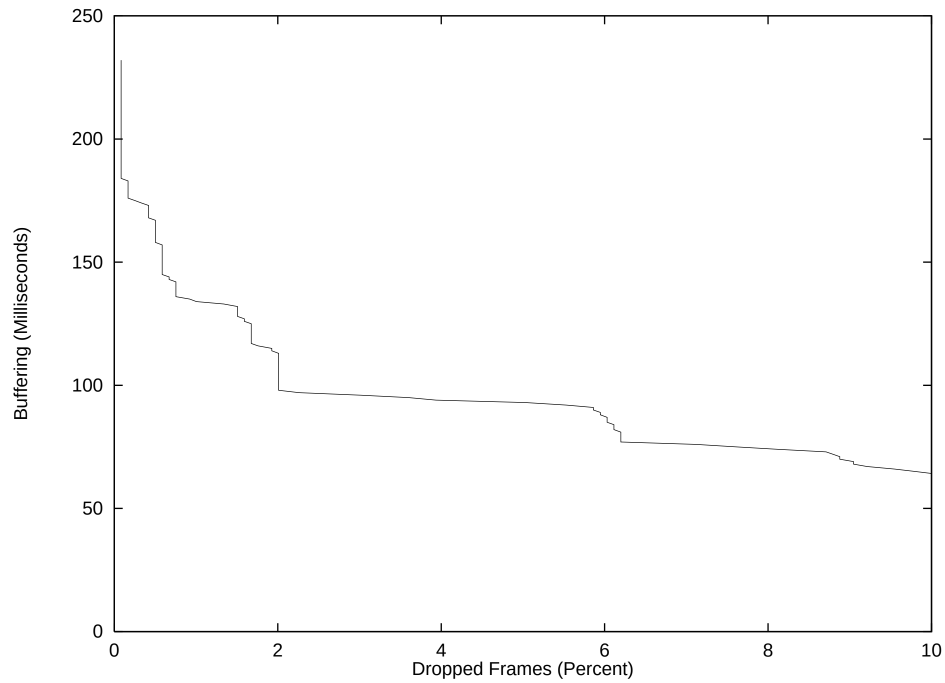

Figure 3: Jitter Compensation. This picture depicts the amount of buffering needed for a given number of dropped frames. The horizontal axis is the percentage of dropped frames. The vertical axis is the number of milliseconds of buffering needed.

the video frame.

3.4 Determining Data Loss

In order to predict data loss, we need to identify what form data loss may take and when data loss may occur. In general, data loss can take many forms such as reduced bits of color, jumbo pixels, smaller images, dropped frames and lossy compression. For a videoconference, we assume data loss only in the form of dropped frames or reduced frame rate. For a videoconference, we assume data loss under three conditions:

- *Voluntary.* As described in Section 3.1, an application may chose to discard late frames in order to keep playout latency low and constant. We assume the videoconference chooses to discard enough frames to achieve the best quality.
- *Saturation.* When either the network or the processor do not have sufficient capacity to transmit data at the required frame rate, data loss occurs. For example, if the network has a maximum bandwidth of 5 Mbps and the videoconference required 10 Mbps there will be a 50% data loss. We can compute when systems reach capacity based on our previous work measuring processor capacities [CR94, CR96] and theoretical network bandwidths.
- *Transmission Loss.* In previous experiments, we found that typically about 0.5% packets on the average are lost when the network is running under maximum load [CR96]. We assume

a maximum lost data rate of about 0.5% due to network transmission.

3.5 Determining End-to-End Quality

We can now use our metric to explore end-to-end videoconference quality under different system configurations. We can quantify how effectively today's computer systems support multi-person videoconferences. We can determine when today's systems will fail due to too many users or too much load on the processors or networks. We can see how much using real-time priorities will help videoconference quality. We can evaluate the benefits of expensive high-performance processors and high-speed networks before installing them. We can even investigate possible performance benefits from networks and processors that have not yet been built. Let's go exploring!

We determine application quality for three scenarios: 1) high-performance processors and high-speed networks; 2) increasing users; and 3) increasing system load.

For all of our videoconference quality predictions we assume:

- *Multicast.* Our previous work has found that multicast is crucial for many-person multimedia applications [CR96]. Using unicast routing, multi-person multimedia applications saturate existing networks for even a few participants. Multicast routing dramatically increases the user scalability of multi-person applications.
- *Specialized Hardware.* The processor load for processing video frames can be substantial [CRC+95]. We assume specialized hardware that does most of the computation required for video frame processing.

3.6 High-Performance Processors and High-Speed Networks

Our previous experimental results showed that both high-performance processors and high-speed networks reduce jitter [CHR97]. However, which reduces jitter more? And more importantly, which improves application quality more?

We assume we have five videoconference participants. In Subsection 3.7, we use our model to evaluate quality for a variable number of users, but here we evaluate a likely videoconference configuration that has interesting quality predictions. We compute quality under two different scenarios. In the first, processor load remains constant while the network bandwidth increases. In the second, network bandwidth remains constant while processor power increases. We use the Standard Performance Evaluation Corporation (SPEC) benchmarks to make predictions about end-to-end quality on more powerful workstations [Cor94]. Figure 4 shows these predictions. For five users, increasing the processor power to a SPECint92 of 40 or greater results in acceptable videoconference quality. At no time does increasing the network bandwidth result in an acceptable quality. In this scenario, we conclude that processor power influences videoconference quality more than does network bandwidth.

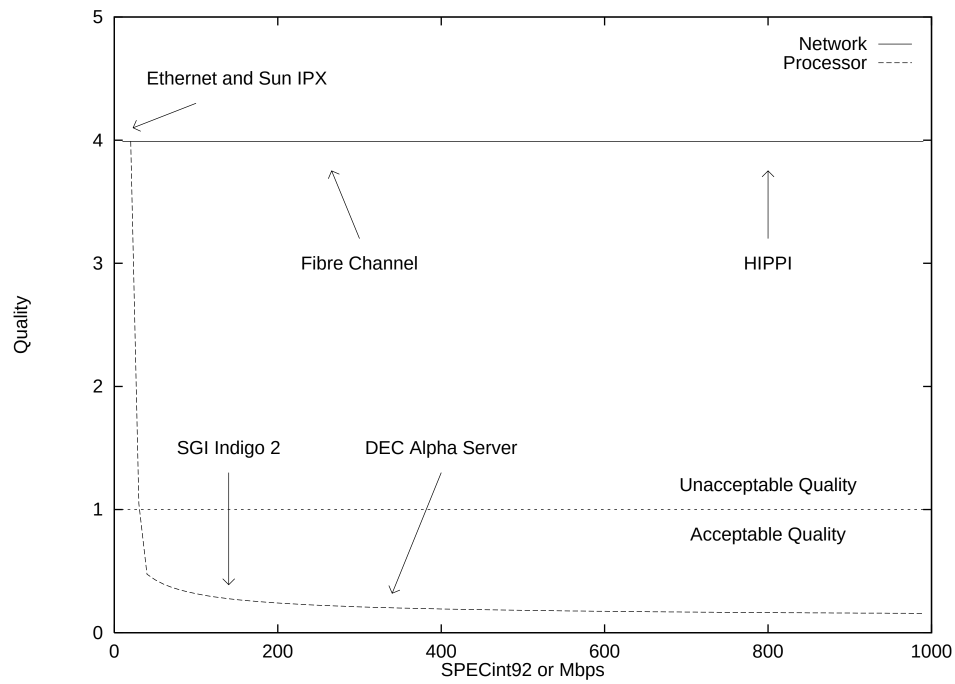

Figure 4: Videoconference Quality versus Processor or Network Increase. The horizontal axis is the SPECint92 power of the workstation or the network Mbps. The vertical axis is the predicted quality. There are two scenarios depicted. In the first, the processor power is constant, equivalent to a Sun IPX (SPECint92 = 22), while the network bandwidth increases. This is depicted by the solid curve. In the second scenario, the network bandwidth is constant, equivalent to an Ethernet (10 Mbps), while the processor power increases. This is depicted by the dashed curve. The horizontal line marks the limit between acceptable and unacceptable videoconference quality.

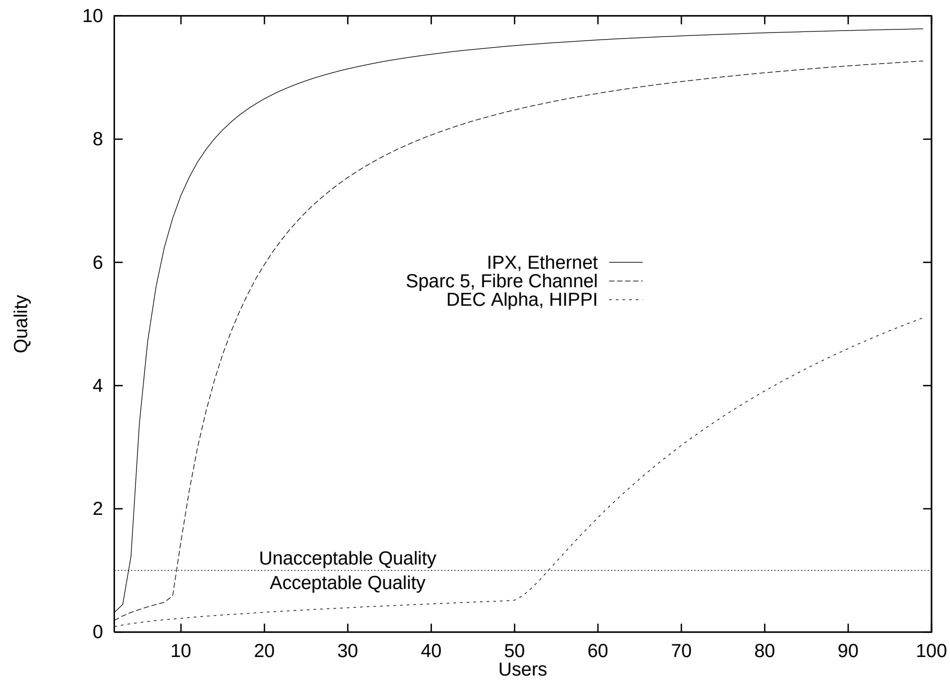

Figure 5: Videoconference Quality versus Users. The horizontal axis is the number of users. The vertical axis is the predicted quality. There are three scenarios depicted. In the first, the processors is a Sun IPXs connected by an Ethernet. In the second, the processors is a Sun Sparc 5s connected by a Fibre Channel. In the third, the processors are DEC Alphas connected by a HIPPI. The horizontal line marks the limit between acceptable and unacceptable videoconference quality.

3.7 Users

While today's computer systems may struggle to support even five videoconference participants, tomorrow's processor improvements promise to support more and more users. But how many more? How do more and more videoconference users affect application quality? Figure 5 depicts the predicted effects of increasing users on videoconference quality. We predict videoconference quality for three different videoconference configurations: a low-end workstation with a typical network (Sun IPX and Ethernet), a mid-range workstation with a fast network (Sun Sparc 5 and Fibre Channel), and a high-performance workstation with a high-speed network (DEC Alpha and HIPPI). As we saw in Subsection 3.6, today's low-end workstation and typical Ethernet network cannot support even five videoconference participants. However, workstations such as Sun Sparc 5s connected by fast networks such as a Fibre Channel can support up to 10 users. Very high-performance workstations such as DEC Alphas connected by a high-speed network such as a HIPPI can support over 50 users.

3.8 Processor and Network Load

Videoconferences are resource intensive, forcing processors and networks to run at heavy loads. In addition, videoconference streams are often integrated into larger distributed multimedia applica-

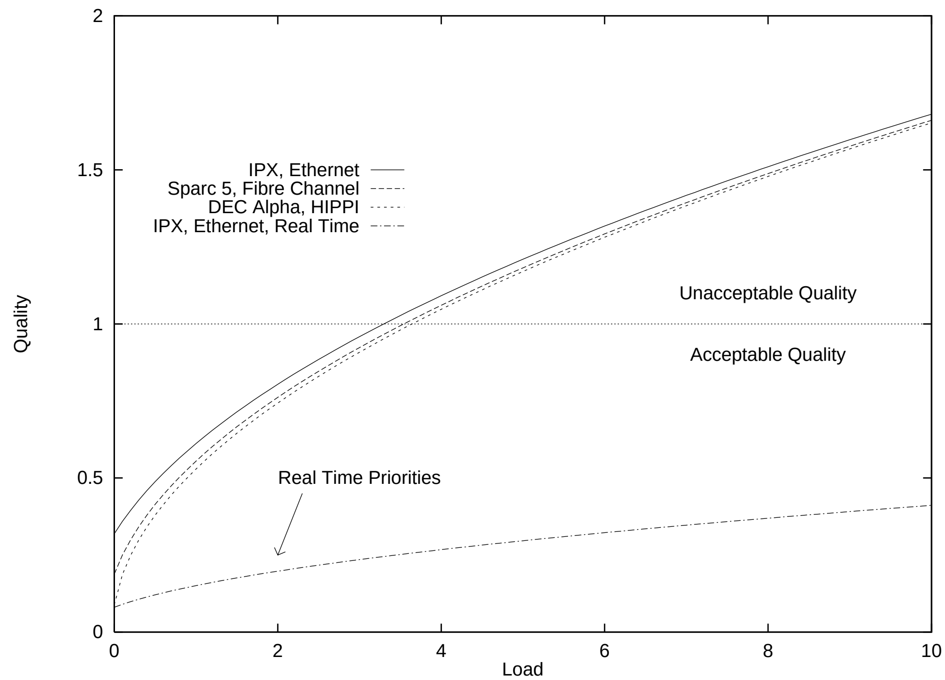

Figure 6: Videoconference Quality versus Load. The horizontal axis is the processor load. The vertical axis is the quality prediction. There are four system configurations depicted. In the first, the processors are Sun IPXs connected by an Ethernet. In the second, the processors are Sun Sparc 5s connected by a Fibre Channel. In the third, the processors are DEC Alphas connected by a HIPPI. In the fourth, the processors are again Sun IPXs connected by an Ethernet, but they are using real-time priorities instead of default priorities. The upper horizontal line marks the limit between acceptable and unacceptable videoconference quality.

tions. In the past, applications have tended to expand to fill (or surpass) available system capacity. As system capacities increase, videoconference users will demand higher frame rates and better resolution, making heavy-load conditions likely in the future. We predict the effects of increasing load on videoconference quality.

Figure 6 depicts the predicted effects of load on videoconference quality (remember, we are assuming specialized hardware for video processing and multicast routing). There are three classes of systems depicted. A traditional system has Sun IPXs connected by an Ethernet. A mid-range system has Sun Sparc 5 connected by a Fibre Channel. A high-end system has DEC Alphas connected by a HIPPI. The predictions for videoconference quality are almost identical for the three systems. We saw in Section 3.6 that the processor is more crucial than network for videoconference quality. Increasing processor load has a larger effect on decreasing videoconference quality than does improving the network speed and processor power.

Figure 6 also depicts Sun IPXs connected by an Ethernet but using real-time priorities instead of default priorities, shown by the bottom line. With real-time priorities, videoconference quality does not suffer from increased jitter from the processor as processor load increases. For conditions of increasing load, real-time priorities have a greater effect on improving quality than do faster processors and faster networks.

4 Summary

Today's explosive growth in fast networks and powerful workstations has provided the potential to support and even enhance group work through multimedia applications. Before realizing the real and potential benefits of multimedia, we must overcome several obstacles in designing multimedia applications and systems. Multimedia and multi-user applications are more resource intensive than traditional text-based, single-user applications. In addition, multimedia applications have different performance requirements than do text-based applications. Text-based applications are sensitive to latency and loss, while multimedia applications are sensitive to latency and jitter. The bottlenecks to text-based application performance might lie in those components that induce latency, while the bottlenecks to multimedia applications might lie in the those components that induce the jitter. New techniques must be developed to identify bottlenecks in the end-to-end perceptual quality of multimedia applications.

A measure of end-to-end quality must take into account the components fundamental to multimedia applications: latency, jitter and data loss. In addition, such a measure should allow the investigation of bottlenecks in quality by being adjustable to: number of users, applications, different quality metrics and alternate hardware and architectures. In this chapter, we present one such measure of multimedia application quality from the user perspective.

There are still many exciting areas for future work. As presented in this chapter, the fundamental perceptual quality component of "data loss" can be expanded upon. We have described the "percent" of lost data as if all bytes in a multimedia stream were equivalent. In reality, some parts of the stream are more important that others. For example, the silent parts between words in speech are less important for user intelligibility than the words themselves. Likewise, movie frames that have very little scene change from the previous frames are likely less important to the user than are frames which vary a lot. On the systems level, too, there may be bytes which are more important than other bytes. For example, the I frames in an MPEG video stream are more important to the system than the P frames since you cannot reconstruct subsequent P frames without the I frames. If possible, the data loss axis should be weighted to reflect the import of the bytes that are actually lost.

For some applications, there is potential for interaction effects among the quality events. For example, 3-d graphics applications have multiple factors affecting users' perception of objects and different combinations of requirements may yield satisfactory results. Such applications may even have a non-convex region of acceptable quality. Future research into new quality metrics appropriate for these applications may be required.

Applications that have changing user requirements present another challenge. For example, users doing remote problem-solving via a video link, may want to maximize frame rate at the expense of frame resolution while they are identifying the location of the problem. Once the problem is located, they may want to maximize frame resolution at the expense of frame rate (perhaps even wanting a still image) to best identify the problem. As presented, our metric does not allow specification of dynamic user requirements. One possible solution would be to apply a separate quality metric to each set of user requirements specified. The metric that had the poorest quality for a given system configuration could then be examined more closely to determine the application quality bottleneck within.

References

[AFKN95] Ronnie T. Apteker, James A. Fisher, Valentin S. Kisimov, and Hanoch Neishlos. Video acceptability and frame rate. *IEEE Multimedia*, pages 32 – 40, Fall 1995.

[CHR97] Mark Claypool, Joe Habermann, and John Riedl. The effects of high-performance processors, real-time priorities and high-speed networks on jitter in a multimedia stream. Technical Report TR-97-023, University of Minnesota Department of Computer Science, June 1997.

[Cor94] Standard Performance Evaluation Corporation. SPEC primer. July 1994. The SPEC primer is frequently posted to the newsgroup comp.benchmarks. SPEC questions can also be sent to spec-ncga@cup.portal.com.

[Cor97] American Technology Corporation. *HyperSonic Sound System (HSS) – (A New Method of Sound Reproduction)*. July 1997. The URL for this document can be found at http://www.atcsd.com/HTML/sound2.html.

[CR94] Mark Claypool and John Riedl. Silence is golden? The effects of silence deletion on the CPU load of an audio conference. In *Proceedings of IEEE Multimedia*, Boston, May 1994.

[CR96] Mark Claypool and John Riedl. A quality planning model for distributed multimedia in the virtual cockpit. In *Proceedings of ACM Multimedia*, pages 253 – 264, November 1996.

[CRC+95] M. Claypool, J. Riedl, J. Carlis, G. Wilcox, R. Elde, E. Retzel, A. Georgopoulos, J. Pardo, K. Ugurbil, B. Miller, and C. Honda. Network requirements for 3D flying in a zoomable brain database. *IEEE JSAC Special Issue on Gigabit Networking*, 13(5), June 1995.

[CSZ92] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in and integrated services packet network: Architecture and mechanism. *Computer Communication Review*, 22(4), July 1992.

[DCJ93] Spiros Dimolitsas, Franklin L. Corcoran, and John G. Phipps Jr. Impact of transmission delay on ISDN videotelephony. In *Proceedings of Globecom '93 – IEEE Telecommunications Conference*, pages 376 – 379, Houston, TX, November 1993.

[Fer92] Domenico Ferrari. Delay jitter control scheme for packet-switching internetworks. *Computer Communications*, 15(6):367–373, July 1992.

[FM76] J.W. Forgie and C.W. McElwain. Some comments on NSC note 78 'effects of lost packets on speech intelligibility.'. Technical Report Network Speech Compression Note 92, M.I.T., Lincoln Laboratory, March 1976.

[IKK93] Satoru Iai, Takaaki Kurita, and Nobuhiko Kitawaki. Quality requirements for multimedia communcation services and terminals – interaction of speech and video delays. In *Proceedings of Globecom '93 – IEEE Telecommunications Conference*, pages 394 – 398, Houston, TX, November 1993.

[JSS92] K. Jeffay, D. Stone, and F. Smith. Kernel support for live digital audio and video. *Computer Communications*, 15(6), 1992.

[JSTS92] K. Jeffay, D. L. Stone, T. Talley, and F. D. Smith. Adaptive, best-effort, delivery of audio and video data across packet-switched networks. In *3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, November 1992.

[JVS91] Saimin Jin, Dhadesugoor R. Vaman, and Divyendu Sina. A performance mangement framework to provide bounded packet delay and variance in packet switched networks. *Computer networks and ISDN Sytems*, pages 249 – 264, September 1991.

[Kal94] Kalkbrenner. Quality of Service (QOS) in distributed hypermedia systems. In *Proceedings of the 2nd International Workshop on Principles of Ducment Processing*, 1994.

[KN82] Kleinrock and Naylor. Stream traffic communication in packet switched networks: Destination buffering considerations. *IEEE Transactions on Communications*, COM-30(12):2527 – 2534, December 1982.

[MS94] Michael J. Massimino and Thomas B. Sheridan. Teleoperator performance with varying force and visual feedback. In *Human Factors*, pages 145 – 157, March 1994.

[OOM95] Judith S. Olson, Gary M. Olson, and David K. Meader. What mix of video and audio is useful for remote real-time work? In *Proceedings of CHI'95 – Proceedings of the Conference in Human Factors in Computing Systems*, pages 362 – 368, 1995.

[Par94] Craig Partridge. *Gigabit Networking*. Addison-Wesley, 1994.

[PSR93] Ketan Patel, Brian C. Smith, and Lawrence A. Rowe. Performance of a software mpeg video decoder. In *Proceedings of ACM Multimedia*, Anaheim, CA, 1993.

[Roy94] Radhika R. Roy. Networking contraints in multimedia conferencing and the role of ATM networks. *AT&T Technical Journal*, July/August 1994.

[RS94] Ramachandran Ramjee, Jim Kurose, Don Towsley and Henning Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies on Networking for Global Communciation. Volume 2*, pages 680–688, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[SJ95] Donald Stone and Kevin Jeffay. An empirical study of delay jitter management policies. *Multimedia Systems*, 1995.

[SR95] Michael V. Stein and John T. Riedl. The effects of transport method on the quality of audioconferences with silence deletion. Technical report, University of Minnesota Department of Computer Science, June 1995.

[SW93] Merryanna Swartz and Daniel Wallace. Effects of frame rate and resolution reduction on human performance. In *Proceedings of IS&T's 46th Annual Conference*, Munich, Germany, 1993.

[TJ94] T. Talley and K. Jeffay. Two-dimensional scaling techniques for adaptive, rate-based transmission control of live audio and video streams. In *Proceedings of the Second ACM International Conference on Multimedia*, pages 247 – 254, October 1994.

[WS95] Duminda Wijesekera and Jaideep Srivastava. Quality of service metrics for continuous media. *Protocols for Multimedia Systems*, pages 269 – 289, 1995.

[ZDE93] Lixia Zhang, Stephen Deering, and Deborah Estrin. RSVP: A new resource ReSerVation Protocol. novel design features lead to an Internet protocol that is flexible and scalable. *IEEE network*, 7(5):8–??, September 1993.

[Zeb93] J.A. Zebarth. Let me be me. In *Proceedings of Globecom '93 – IEEE Telecommunications Conference*, pages 389 – 393, Houston, TX, November 1993.