An Integrated Metric for Video QoS

Nalini Venkatasubramanian
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
nalini@cs.uiuc.edu

Klara Nahrstedt
Department of Computer Science
University of Illinois, Urbana-Champaign
Urbana, IL 61801
klara@cs.uiuc.edu

Abstract

In this paper, we address the issues in designing metrics that are important in evaluating the Quality of Service(QoS) of video transmission. We propose a new metric for video QoS called the *weighted cost-satisfaction ratio* based on requirements from two perspectives: the user and the service provider. To understand real video workload environments and user behavioral patterns, we obtained and analyzed empirical results from the VOSAIC (video-over-the-Web) system, a hierarchical video-on-demand (VOD) system and a remote VCR system. Based on these results, we define parameters of resource consumption (storage and network bandwidth etc.) and user satisfaction (jitter, synchronization skew) and derive analytical interrelationships among the metric parameters. We also draw an economic relationship between the user-satisfaction and resource consumption factors to solve metric optimization relations. *Keywords: QoS, video, resource consumption, user satisfaction, workload.*

1 Introduction

Current Web based systems are designed for the static class of document based information. In the next generation of information systems, digital video will form an important class of traffic. Emerging systems will need to handle multimedia applications that possess varying traffic characteristics and have Quality-of-Service(QoS) requirements in terms of bounded delay, jitter, loss rate, synchronization skew etc. [13]. In order to design and deploy these complex and evolving systems with proper cost/performance ratios, we must understand their expected behaviors and workloads. Large scale video service applications that are currently being deployed have some significant problems: (1) User dissatisfaction due to poor QoS. (2) Poor cost-performance ratios due to inefficient management of system resources, especially when guaranteed service is desired. In order to resolve these issues, we will initially need to identify bottlenecks in the system that are responsible for poor response times. We can then determine suitable mechanisms to obtain cost-effective QoS.

In this paper, we address the issues in designing metrics that are important in evaluating the QoS of video transmission. There has been little work in determining effective metrics of QoS for video transmission that characterize both cost (revenue generated or service demand) and guaranteed service. The metrics of analysis and comparison for video transmission must be determined as an end-to-end measure of QoS from video server to end-user(s). By developing these metrics, we hope to enhance the client, server and networking components of a system with monitoring capabilities to measure and evaluate video characterizations. This paper is organized as follows. In Section 2, we discuss a workload model for developing and understanding QoS metrics. Section 3 presents empirical studies and experimental justification for the metric selection based on the three systems - VOSAIC, hierarchical VOD and the remote VCR systems. Section 4 proposes a new integrated metric for measuring video QoS and the analytical framework to express the tradeoffs. We also propose a metric-based QoS architecture along with negotiation and reward protocols. In Section 5 we discuss related work and conclude with future research directions in Section 6.

2 QoS Metric Selection

QoS is defined as a set of perceivable attributes expressed in user-friendly language with parameters that may be subjective or objective. Objective values are parameters related to a particular service and are measurable and verifiable. Subjective values are based on the opinions of the end-users. The specification and enforcement of QoS presents interesting challenges in multimedia systems development.

2.1 QoS specification and enforcement

Typical application QoS parameters for images and video include image size, frame rate, startup delay, reliability etc. The application QoS profile can also include subjective factors such as the degree of importance of the information to the user and the overall cost-quality metric that the user desires. Network QoS parameters include bandwidth, delay, jitter and loss rate. End-system parameters include CPU load, utilization, buffering mechanisms and storage related parameters. Users express dynamic preferences for media quality through benefit functions, [21], for e.g. (1) frame rate benefit function which indicates that beyond a threshold frame rate, there is no additional benefit, (2) synchro-

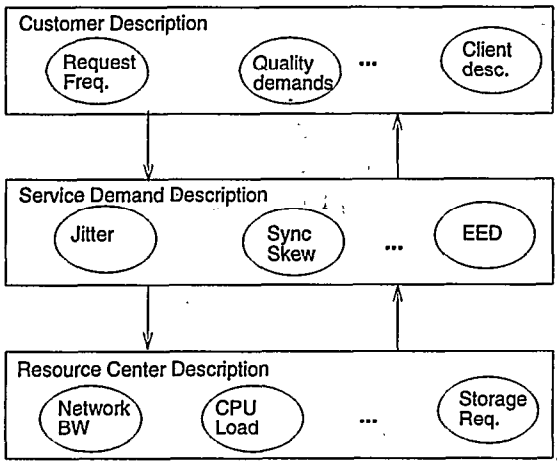

Figure 1: Components of a QoS Based Architecture

nization benefit function which indicates that the benefit is high only when the audio/video synchronization skew is low.

There are several challenges in delivering the specified QoS to video applications. The mapping between different sets of parameters at different levels in the system, the *QoS translation* process is one of the challenges in meeting end-to-end performance bounds. QoS parameters at the user level must be translated to quantitative parameters at the network and system level. This translation is required during the admission control process where resource availability at different levels must be queried and negotiated for. The admission control process represents a second challenge and consists of a variety of tests such as EED(end-to-end delay), buffer allocation and schedulability tests [16]. Since network and system conditions are likely to be dynamic, the optimal operating point is not constant and this represents yet another challenge in QoS enforcement. Several QoS parameters such as deadlines and packet losses [6] must dynamically adapt during processing and communication. Violation of the QoS bounds portrays itself as flickering video, distorted audio, blanks and frozen images.

2.2 The Workload Model

Selection criteria for video QoS metrics come from two perspectives - the end-user and the service provider. With a VOD application, the end-user requirements are dominant and stringent, the QoS is prespecified and is expected to be met accurately. In an unconstrained environment such as the Internet, both user and service-provider factors are variable. The end-user desires that the best approximation to the QoS criteria be satisfied while the service provider wants to ensure cost-effective utilization of resources. In order to satisfy the dual requirements of user-satisfaction and cost effectiveness, we need integrated metrics for determining the best possible QoS.

To determine measures of cost-effectiveness and user satisfaction, we must define a workload model. Our workload model consists of three elements (see Figure 1):

- **Customer description:** description of expected workloads and classes of customer requests and the correspondence of the workload components to customer classes. This information is typically obtained from empirical studies by observing customer request patterns and analyzing them.
- **Service demands:** description of interaction b/w customers and service (resource) centers (e.g. QoS requirements). This involves the translation of abstract customer descriptions obtained from empirical studies into more tangible parameters in an analytical model.
- **Service Center description:** a baseline description of the set of services that serve user requests and the correspondence of resources to service centers.

From the workload model it follows that selecting a unified metric will involve a choice of multiple parameters. This choice is based on empirical and statistical data which provides a base for analytical models with user requirements and resource allocations.

3 Empirical Workload Measurements and Analysis

Our study is based on an understanding of real video workload environments, user behavioral patterns and system responses to changes in workload. We start by identifying a baseline class of workload components and a simplified characterization of the workload. For instance, we expect the video workloads to be largely I/O bound and not CPU bound [1].

Our metrics and measurements are derived from different Video-on-Demand (VOD) systems implemented at the University of Illinois, Urbana-Champaign. The first VOD system considered is the VOSAIC web system and we will observe the user access log characteristics obtained over a 20 day observation period of the VOSAIC (video-over-the-internet) system [3]. The user access logs allow us to analyze the user access statistics and general user information important for modeling of *user satisfaction* (see section 4.1). The second system considered is a hierarchical VOD system with a hierarchical set of video servers [14]. In this system, we observe performance characteristics such as jitter and frame loss rate. The third system is a simple remote VCR system with a single VOD server and client that we use to observe synchronization skew performance characteristics. The jitter, loss rate and skew characteristics allow us to analyze the system access and resource dependencies important for modeling of *resource consumption* (see section 4.2). Before we discuss performance analysis results, we briefly present the main architectural features of the three VOD systems.

- VOSAIC is essentially a framework for the integration of audio-visual information into a standard hypertext document and Web environment. It transmits information on-demand in real-time over the current Internet. While traditional VOD systems are designed to operate in a constrained environment, VOSAIC is designed to operate in unconstrained environments. VOSAIC is currently designed as a client-server video playback system. The session protocol used in VOSAIC is VDP (Video Datagram Protocol), which uses a combination of TCP and UDP: the playback command and request from client to server are sent over TCP connections and actual video and audio data transmission from server to client are carried out over UDP connections. VDP uses a feedback scheme

[1] This might change if video applications start to use image pattern recognition, face recognition, scene recognition and other CPU intensive image processing tasks.

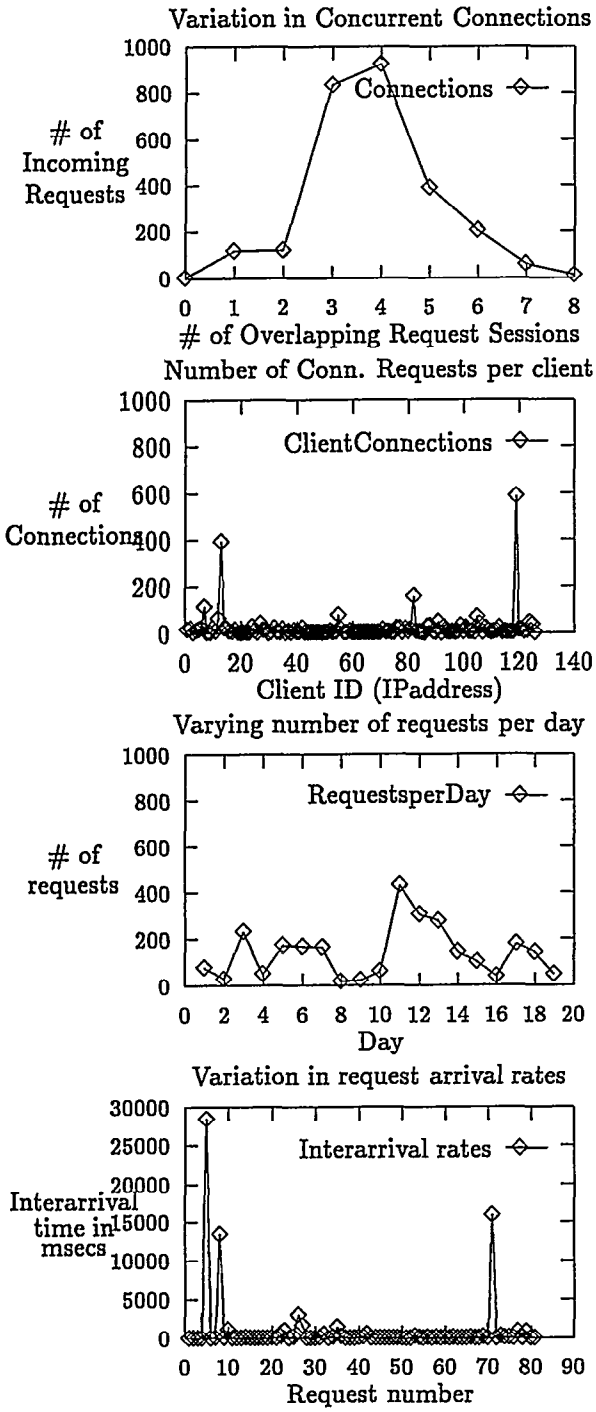

Figure 2: VOSAIC Data

to detect network congestion and automatically delete frames in response to the congestion. The HTTP protocol is used for text/data communications between the client and server while MPEG video/audio uses VDP.

- The hierarchical client-server VOD system utilizes a distributed server approach. The VOD system consists of a set of *Cache Servers*(CS) being on the same LAN with *VOD Clients*, a *Cache Agent* running on the CSs, and a *Primary Server* (PS) that is on an external network. The CS downloads the requested video streams from PS in non-realtime (using a best effort service) and provides real-time services to VOD clients. Although there is an initial startup delay while CS downloads the data for the first time, the VOD client does not have to cache the whole video as is the case with WWW applications. Once the video is cached on the CS, streaming for display and real-time services such as fast forward and rewind are available. The CA controls resources of the set of CSs on the same LAN to achieve good load balancing and better performance. The hierarchical VOD system consists of two protocols: (1) *Hierarchical Media Management Protocol*(HMMP) used by the CA, and (2) *Multimedia Stream Session Protocol*(MSSP). HMMP uses TCP/IP for management communication among the CSs. MSSP uses TCP/IP for (1) command control communication during connection setup and transmission, (2) downloading of a new video from PS to CS and (3) audio streaming from CS to VOD client. MSSP uses UDP/IP for video streaming from the CS to the VOD client. The CA with HMMP provides resource reservation during the connection setup, hence there is no adaptation activity during transmission over an admitted connection. In contrast, VOSAIC does not employ resource reservation during the connection setup, but supports feedback-oriented adaptation during video transmission.

- The remote VCR system consists of a single VOD client and VOD server utilizing an adaptive synchronization protocol [18]. The adaptive synchronization protocol synchronizes MPEG compressed videos. The connection setup and command control are performed over TCP/IP. The transmission component is built on top of the UDP/IP protocol stack. This protocol does not have any resource reservation, but it enforces adaptive synchronization to keep synchronization skews within acceptable bounds.

3.1 User Access Analysis

The data collected in the VOSAIC user access log files provides information on user access statistics and general user information such as IP address, access-time, access filename, total connection time and playback information. Since the workload is representative of an exclusive video storage repository, accuracy of data in predicting future access patterns is enhanced. However, in the VOSAIC logs, there is little information on jitter, response latencies to a user request and low level resource utilization. The study of the access logs indicates a large variation in request accesses. The study set consists of 150 video objects in the video server with 126 clients accessing these video objects over the 20 day period.

The graphs in Figure 2 depict some of the variations in user requests and connections in the sample VOSAIC data.

From Figure 2, it is evident that while the system can support a limited number of concurrent sessions (approximately 8), the typical operating point is lower (about 4-5 concurrent requests). This may be caused by typical access patterns or by admission control techniques that limit the number of concurrent sessions based on the resource utilization of each session. Traditional HTTP servers do not require admission control due to short bursty transfers, but this is not true for continuous media requests. The admission controller component in the VOSAIC server estimates the resource requirement of an incoming request and admits the request only if sufficient resources are available. The ascending section of the graph indicates that the utilization of the system is based on the availability of requests. The descending section of the graph depicts the resource bottleneck, i.e. as the number of overlapping requests increase, resource utilizations are heavy leading to either a larger number of rejects or longer response times. The graphs in Figure 2 indicate variations in client accesses (few clients request very frequent accesses) and variation in the number of requests per day over the period observed.

Based on the variations and results obtained from the VOSAIC empirical studies, we classify video workloads into multiple classes. We classify video objects into *long(L)*, *medium(M)* and *short(S)* objects based on the size of the video and the amount of storage occupied. Another level of classification is based on the degree of access to a video object. We define frequently accessed video objects to be *hot(H)*, moderately accessed video objects to be *warm(W)* and infrequently accessed videos to be *cold(C)*. Ensuring video QoS requires guarantees that cannot be supported with best-effort traffic. Hence, we classify service requests as *guaranteed(G)* vs. *best-effort(B)* requests and apply the results from the empirical studies to determine the guaranteed class. This categorization can be summarized into a few *QoS classes* that represent the degree(quality) of service that can be expected. The choice of values that determines boundaries among individual classes is based on empirical parameters, for example, those obtained from the VOSAIC system analysis. In the sample logs, out of 150 video objects, 8 have over 50 requests, giving a 5 percent ratio of hot videos in the system. Similarly, from the study conducted, we consider short videos to be those that last less than a minute and long videos to be ones that last longer than a minute. In general, the choice of boundary values for any system is dependent on the workload, user class and other parameters. Therefore, empirical studies of the system will indicate values for separation into QoS classes.

3.2 System Overhead Analysis

We obtained detailed performance measurements of two VOD systems (the hierarchical VOD system and the remote VCR system) to analyze system overheads and resource consumption dependencies. We use these results to model resource consumption and relations between user satisfaction and resource consumption.

From the hierarchical VOD system we gathered measurements of video jitter and frame loss rate. From the remote VCR system we gathered measurements of synchronization skews between MPEG-compressed audio and video streams. We first describe the experimental workload and then present the results and their conclusion on both systems. With the hierarchical VOD system two major experiments were conducted. The experimental setup was as follows: The experimental LAN was a 10 Mbps Ethernet,

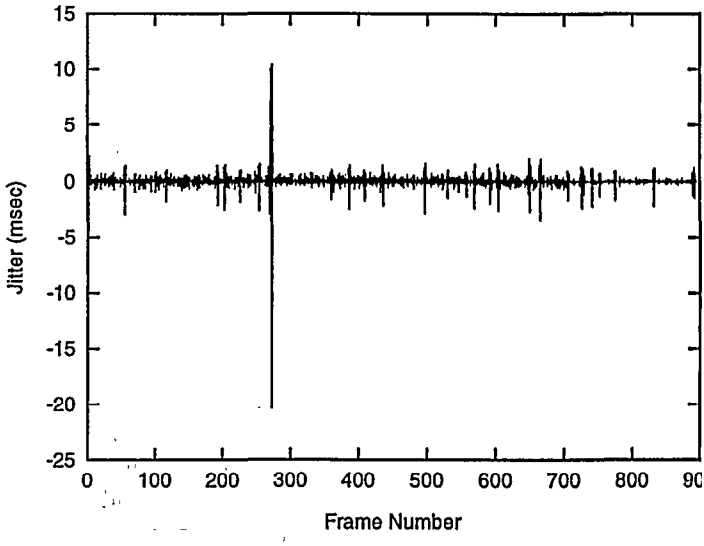

Figure 3: Jitter of Displaying Video Frames

the video files were JPEG compressed (320x240), recorded at a rate of 30 fps and displayed at a rate of 30 fps or less using the Parallax JPEG boards. Audio data were in SUN au format of telephone quality. In *Experiment 1*, we measured video jitter at the VOD client side when there was only one CS on the LAN and the CS served a video stream to a single VOD client. This experiment represented a very light workload. The resulting video jitters are shown in Figure 3. The CPU resource consumption of the individual threads at the VOD client and CS are shown in Table 1.

The results indicate that this system is capable of providing very high quality video streams for a single VOD client with a short connection setup delay, no out-of-sync errors, no frame losses, low video display jitter and low CPU utilization on the CS. The low CPU consumption on CS indicates the ability of the CS to serve many VOD clients. Higher CPU resource consumption at the VOD client side is largely due to the JPEG decoding thread.

The second experiment (*Experiment 2*) with the hierarchical VOD system consisted of supporting multiple VOD client streams from a single CS server. In this experimental setup, the network bandwidth was managed by the CA. We set the maximum bandwidth b_{max} maintained at CA to 7Mbps and 10Mbps. When CA assumes b_{max} = 10Mbps, it allowed CS's (using an admission control mechanism) to send data up to 10Mbps. Since it is difficult for a 10Mbps Ethernet to provide 100% bandwidth in practice, congestion was observed when the CS tried to send traffic at approximately 10Mbps. The experiment allocated 6 clients requesting the same 30 fps video stream which is already downloaded in CS. The results of this experiment concentrated on measuring the frame loss rate due to increased workloads on the CS. The results of the experiments measuring frame losses are shown in Table 2.

Table 2 shows that when the network is not congested, i.e., the total available bandwidth is bounded by 7 Mbps, then the configuration performs well with low frame loss rates. There are almost no video frame losses. However, the negotiated frame rate delivered to some clients is changed during the admission control process because the CS can-

Client

thread	average $\bar{e_i}$ (ms)	maximum (ms)
WriteAudioHead	2.54	3.45
WriteVideoDisplay	12.39	23.54
ReceiveAudioNet	4.14	66.14
ReceiveVideoNet	0.68	32.69
CPU Consumption	39.9%	

Cache Server

thread	average $\bar{e_i}$ (ms)	maximum (ms)
ReadAudioDisk	0.49	1.31
ReadVideoDisk	0.97	2.21
SendAudioNet	0.85	0.94
SendVideoNet	0.92	1.70
CPU Consumption	5.8%	

(in msec)

Note that most ReceiveAudioNet threads take about 2 msec, however, one long execution (66.14 msec) affects the average.

Table 1: CPU Consumption

	fps	loss	%	fps	loss	%
		skip	%		skip	%
client	b_{max} = 7Mbps			b_{max} = 10Mbps		
CL1 loss	23	0	0	30	589	65
CS loss		0	0		21	2.3
CL2	5	0	0	15	85	19
CS loss		0	0		0	0
CL3 loss	2	0	0	7	37	18
CS loss		0	0		0	0
CL4 loss	11	0	0	30	5	0.6
CS loss		0	0		0	0
CL5 loss	30	4	0.4	30	305	34
CS loss		0	0		36	4.0
CL6 loss	30	0	0	30	56	6.2
CS loss		0	0		0	0
total CL loss		4	0.1		1077	25
total CS loss		0	0		57	1.3

Table 2: Frame Losses in Experiment 2

not deliver the requested 30fps to each client. In the case where the available bandwidth is set at 10 Mbps, representing a fully congested network, many video frames are lost. Due to the burst losses, perceptual viewing quality of the video is unacceptably low in this configuration. This implies that the upper bounds for bandwidth resource allocation and consumption are important indicators of user satisfaction.

The third system, the remote VCR system, was analyzed for synchronization skew performance using the available adaptive synchronization protocol. In order to guarantee user satisfaction, the desired synchronization skew between the audio and video streams was required to be 80ms or lower. Perceptual studies [25] indicate that the skews within this boundary are acceptable to the user. Skews in the range between 80 and 160 ms are tolerable by the user. Skews larger than 160 ms are unacceptable and undesirable by the user. Experiments using the adaptive synchronization protocol within the remote VCR system and the corresponding results, shown in Figure 4, indicate that user acceptable or tolerable synchronization performance can be achieved [2].

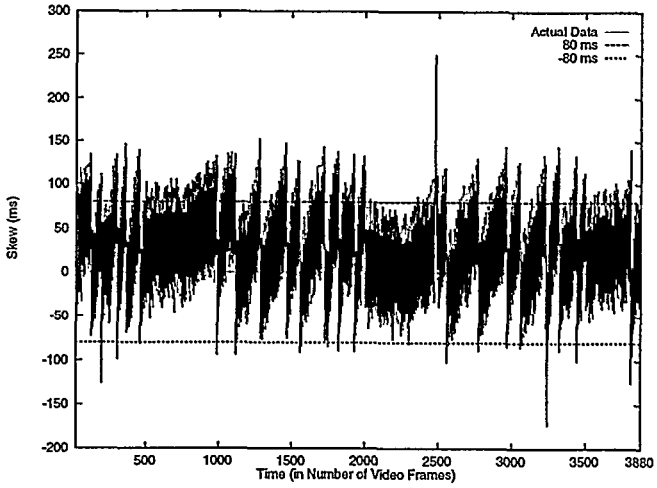

Figure 4: Synchronization Skew. The characteristics of the video clip are MPEG compressed video with Pattern:IBPBPBI, size 320x240, rate 7fps, MPEG compressed audio with layer II encoding and rate 44.1 kHz.

In summary, the hierarchical VOD system and the remote VCR system show that if sufficient resources are reserved, and/or adaptive mechanisms are enforced, user satisfaction can be provided. Based on the experimental results presented here, we see that a weighted correlation between resource consumption and user satisfaction exists; this relationship is modeled in the next section.

4 A New Unified Metric of Video QoS

From the empirical study and the workload model obtained, we learn that there are two kinds of factors that a QoS metric must represent. These are (1) resource consumption factors and (2) user satisfaction factors. *Resource consumption factors* are a measure of the cost incurred due to re-

[2] We present in this paper only results from one experimental setup. More experimental results are presented in [18].

source utilization in the system and include CPU, storage and network related parameters. User satisfaction factors quantify the QoS guarantees met and factors that affect the delivery of the desired response quality. Application QoS related parameters such as frame rate, frame width, frame height, color resolution, compression ratio, jitter for video and synchronization skew are good measures to determine the achieved quality and the deviation between the actual response and the desired quality.

We propose a new metric for measuring the effectiveness of video transmission in dynamic situations based on aggregate resource and response factors [19]. We call this metric the *weighted resource cost- user-satisfaction ratio* and it includes parameters that specify the various objectives of QoS. To quantify levels of service, we assign weights to user satisfaction (US) and the cost incurred due to resource consumption (RC) and construct a benefit function from the composition of the weighted objectives. Let the QoS metric that represents service to user i be denoted as M_i.

$$M_i \propto \frac{W_{US} * US_i}{W_{RC} * RC_i} \tag{1}$$

where W_{US} represents the weight of the user satisfaction component and W_{RC} represents the weight of the resources consumed component. In the following subsections we elaborate further on each factor.

4.1 User Satisfaction (US)

The user-satisfaction (US) is related to a number of parameters (success Suc, jitter $\jmath$, price P, end-to-end delay EED, synchronization skew σ, startup latency sL and loss-rate lR).

$$US \propto \frac{Suc}{f(\jmath, P, EED, \sigma, sL, lR)} \tag{2}$$

In each *QoS class*, the success of a request, Suc, is defined differently, since, for example deterministic services must meet stricter performance requirements (e.g. throughput and delay) than best-effort traffic. For example, in a system with admission control, success of a request Suc could be determined by whether the request has been admitted. With best effort traffic, a request is always admitted and hence success could be used to measure whether the request successfully ran to completion. For our study, we assume the presence of an admission control process, and focus on representing and measuring the QoS of a request once it has been admitted, hence Suc is a constant. While the startup latency sL impacts user-satisfaction, it is a non-continuous one-time factor, i.e., it is measured once per session or request and does not impact inter-frame displays. To simplify Equation 2, the parameters loss rate (lR) and end-to-end delay (EED) can be subsumed by the jitter parameter ($\jmath$). We will come back to the issue of price (P) in Section 4.3. The simplified relation is:

$$US \propto \frac{1}{f_{US}(\jmath, P, \sigma)} \tag{3}$$

where f_{US} is a function that represents the relationship between jitter, price and synchronization skew. Furthermore, there exists a strong relationship between the jitter ($\jmath$) and the synchronization skew (σ).

(1) Jitter $\jmath$ between two consecutive packet pairs is defined as follows:

$$\jmath^i = (\delta^i - \delta^{i-1}) \tag{4}$$

The objective is to minimize $\jmath$ and achieve $\jmath = 0$ at the destination d if possible.

(2) Delay D_l of a link: consists of a sum of propagation delay ($DPROP$), queueing delay ($DQUEUE$), and switching delay ($DSWITCH$).

$$D_l = DPROP + DQUEUE + DSWITCH \tag{5}$$

The propagation delay and switching delay are constant, hence the only variable is the queueing delay. The queueing delay can be calculated according to Little's formula $DQUEUE = \frac{n}{\lambda}$. The arrival rate λ at node i depends on jitter built up from the source till node $i-1$, $\jmath$ (i.e. $\lambda(\jmath)$), therefore queueing delay depends on jitter, $DQUEUE(\lambda(\jmath))$, hence delay on link D_l depends on jitter,

(3) Skew σ: Let $t^{i,a}$ and $t^{i,v}$ be time points of two packets from different media which should be synchronized in the same time interval.

$$\sigma = t^{i,a} - t^{i,v} \tag{6}$$

Let us assume $t_A^{i,a}, t_A^{i,v}$ as arrival times at the destination with $t_D^{i,a}, T_D^{i,v}$ as departure times at the source:

$$t_A^{i,a} = EED_a + t_D^{i,a}, t_A^{i,v} = EED_v + t_D^{i,v} \tag{7}$$

$$\sigma = \sum_{l \in R(s,d)} (D_l^a - D_l^v) + (t_D^{i,a} - t_D^{i,v}) \tag{8}$$

$$\sigma = \sum_{l \in R(s,d)} (D_l^a((\jmath)) - D_l^v((\jmath))) + (t_D^{i,a} - t_D^{i,v}) \tag{9}$$

This equation shows the dependency between jitter parameter and the synchronization skew.

4.2 Resource Consumption (RC)

The new metric for video QoS, specified in Equation 1, also depends on the resource consumption (RC), as illustrated in Section 3.2. We consider four factors in determining the resource overhead, i.e., CPU utilization (cpu), network bandwidth utilization (B^{nw}), buffer utilization (Buf) and storage bandwidth overhead (B^{st}). Benefit functions for each parameter can be used to indicate which resources may be more critical based on the design of the system.

$$RC = f_{RC}(cpu, B^{nw}, Buf, B^{st}) \tag{10}$$

An interesting observation is that the resource cost must consider the current set of available resources. As resources get used up to service requests, they become more critical and resource cost becomes more dependent on the fraction of the available resources utilized. When resource availability is low, admission control mechanisms must allocate resources to the high priority tasks.

We characterize the resource consumption of a server S_j due to a request R_i by a ratio that captures the utilization of resources, similar to the load-factor measure in [26]. $RC(R_i, S_j)$ is proportional to:

$$(max(\frac{cpu_i}{cpu_j}, \frac{B_i^{nw}}{B_j^{nw}}, \frac{Buf_i}{Buf_j}, \frac{B_i^{st}}{B_j^{st}})) \tag{11}$$

where r_i is the resource needed by request R_i and r_j is the resource available on server S_j. Different requests have different resource requirements and the RC factor may vary from one request to another. Lower the RC value, greater is the capacity of the system to service additional requests. Hence, the resource cost component of the QoS metric is essentially the cost of the bottleneck resource, since this constraining resource measures the degree of QoS delivered.

4.3 The Economic Relationship between US and RC

The relation M_i (Equation 1) between the resource consumption and user satisfaction factors can be complex and interdependent due to resource sharing. For instance, the user satisfaction component is dependent on the jitter encountered by the receiving client, which may be a result of insufficient resources in the pipeline from the disk to the client buffer. Examples of user satisfaction functions: (1) $U = x_{min} + \sqrt{x}$ (2) $U = ax^3 + bx^2 + cx + d$ where x represents the resource utilization. A QoS metric must capture how far we are from an *optimal* resource allocation. In the presence of conflicting requirements, we use an economic (pricing)framework to represent the near-optimal resource allocation. An economic paradigm is essential when quality must be ensured in a system with shared resources. The metric (weighted cost-satisfaction ratio) is a function that determines the degree of equilibrium of the system, for example the supply-demand equilibrium in economics. The cost-satisfaction ratio can be used to generate the desired trading profile to satisfy the service provider goal of operating in an equilibrium region.

Users of a multimedia system view QoS differently. Without proper pricing mechanisms, users will always request the best available quality and this will result in misuse of the shared resources. In the presence of pricing, the user satisfaction depends not only on QoS parameters, such as EED, jitter or synchronization skew, but also on pricing structures for QoS. Pricing models can be simple and completely static (as with telephone pricing) where a service does not adapt to variations in demand and a user cannot specify multiple levels of QoS. Such models have no incentive for proper usage and hence are inappropriate for our purposes. Complex models of pricing include bidding mechanisms where a server constantly and continuously negotiates for the best price. Studies indicate that fixed-rate pricing policies perform substantially worse than volume-based pricing policies for Internet traffic [8]. Economic models for network provisioning have been studied in [20]. A market-system approach to improve QoS on the Internet involves negotiating and establishing price and quality contracts based on economic principles [8]. Pricing schemes that charge more for peak time usage discourage non-critical requests when the system and network are overloaded. Scaling video applications that permit degradation in picture quality, frame loss coupled with monetary incentives [9], indicate increased user benefit and lower request blocking.

The objective of the billing process that draws a relationship between US and RC is to improve QoS and performance by integrating economic issues with technical issues. Following the results of the empirical study, we would like to design our billing model based on the class of service; i.e., the *QoS level*. For each QoS class described in the workload model, we try to determine a relation between price and resource usage that in turn determines a billing mechanism for that class. Let $Q_{level} = \{Q_{min}, .., Q_{max}\}$ be the ordered set of available QoS levels, $Q_{min} < .. < Q_{max}$. This set partitions the users into QoS equivalence classes according to quality, price and resource requirements. Hence, each QoS class has 3 attributes (P_i, Q_i^j, C_i) where: (1) P_i represents the price range, and quantifies the price that user i must pay for access. (2) Q_i^j represents the QoS desired by the user; $Q_i^j \in Q_{level}$. (3) C_i represents the system resource consumption for user i's requests, i.e.,the resource share attributed to the request by user i. The billing mechanism determines the price (P_i) of service based on Q_i^j (quality desired) and C_i (resource consumed). Note that these factors incorporate the two components of the QoS metric, Q_i^j representing the user satisfaction(US) and C_i representing the resource consumption(RC).

The billing model must accommodate two optimizing criteria from two perspectives: maximizing profit to the service provider while supplying the negotiated QoS to the user. This can be translated as a need to provide a balance between price P_i and QoS Q_i of information, represented as a *benefit function*, B_i. The user i specifies resource requirements to maximize benefit (B_i), i.e. the best QoS for the least price. The user goal is stated as:

$$Maximize\ B_i = Q_i^j - P_i \tag{12}$$

The service provider specifies resource requirements and availability to maximize profit. The service provider goal is stated as:

$$Maximize\ P_i - C_i \geq 0\ while\ Q_i^j. \tag{13}$$

where Q_i^j could be a constraint vector that specifies the resources assigned and price charged.

The pricing model adopted is critical to solving this optimization problem. In general, a pricing model, that determines P_i, must include: (1) the cost of resource consumption, (2) a static startup cost (L), i.e. that of admission control, negotiation, resource reservation and connection setup and (3) a profit factor (K) that quantifies the surcharge introduced by the service provider. Hence, $Price$ is expressed as follows:

$$P_i \propto K * C_i + L_i \tag{14}$$

where C_i determines the resource consumption needed to satisfy the user satisfaction criteria.

In the case of video, the resource consumption(C_i) for the service is a function and is likely to depend on a number of factors such as: (a) Quality of Service Level(Q_i^j), (b) Resource Availability (Res_{avail}), (c) Duration of the Request (δt), and (d) demand for information (Dem) based on popularity of data and time of request.

4.4 An Economic Framework for Metric-based QoS Management

We can use the analysis discussed in the previous section to design a QoS architecture based on economic principles. Figure 5 shows this architecture which consists of users and service providers that establish and provide QoS-based transactions. The components at the service provider(SP) are: (1) *Negotiation Module:* that interfaces with the user to execute negotiation protocols to decide pricing and resource usage for a session. (2) *Trading Module:* that encodes functions to calculate trading profiles that choose trade-offs between resources at the SP to achieve the desired QoS. It is useful when generating resource requirements for a new level of QoS that may be negotiated with a user. (3) *Reward Generation Module:* that determines for given system conditions and resource usage, the reward to be given to the user for cooperating with the system. System conditions and user behavior are monitored by the *Service Monitoring Unit.*

4.4.1 Negotiation Module

User level negotiation protocols allow the SP a possible negotiation mechanism for price/quality tradeoffs. Resource-Price negotiation protocols may be (a) user initiated by users

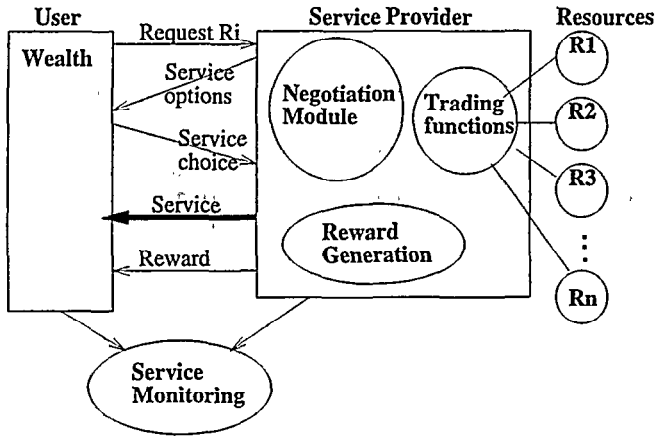

Figure 5: A Metric Based QoS Architecture

```
* User sends request profile R_i to SP.
* SP determines
   Is R_i possible?
   Is S_i in equilibrium.  If not,
    Get resource tradeoffs.
    Call reward algorithms.
* Service options to user
  <S_1,P_1,Rew_1>,...,<S_n,P_n, Rew_n>
* User returns choice of service to SP.
   <S_i, P_i, Rew_i>
* SP provides service.
* SP assesses rewards or penalties.
* Alter wealth of user.
```

Figure 6: A User-level Negotiation Protocol

interested in compromising quality for cost or (b) SP initiated, where the system calculates service alternatives, possibly to admit more requests.

In Figure 6, we describe a simple SP initiated negotiation protocol. The protocol is SP initiated since the service options that represent possible trading choices are sent to the user by the SP. In a user initiated protocol, the user would present the SP with a list of possible service request options. Modifications of this simple protocol for more complex conditions are currently being studied. Note that there is a concurrency issue in the simple protocol. The availability of the resource-price structure promised in the service-choice list must be ensured till the user returns with a choice. This implies that sufficient resources must be made available for any of the options to be executed. There are two factors of non-determinism that affect the timing: (i) the communication delay between the SP and user to communicate choices; (ii) the time taken by the user to respond to the choice. This can affect the availability of resources for other requests that arrive at the SP in the meantime. A simple solution would reserve the maximum amount of each resource required under any service choice for a specified period of time The user response is expected to arrive back to the SP within this specified interval.

```
* Determine wealth(W) of each user.
* Specify rewards for resource levels
   R(x_i) = y_i (x < X_i)
* Specify benefit threshold B_bar.
* User wants 'x' resources:
   get P(x), U(x), B(x).
   If (
    price P(x) <= W
    benefit B(x) <= B_bar
      :
    ) admission positive.
* Check if reward possible
   If x<= x_i then reward R_i
     W := W + R_i
   If x> x_i  then no reward.
* User uses the resource,
    W := W - C(x)
```

Figure 7: A Reward Algorithm

4.4.2 Trading and Reward Modules

Trading involves user-level negotiation protocols that promise reward levels for users who agree to resource tradeoffs and maintain resource behavior during a session. Rewards and penalties are mechanisms used to encourage and ensure good behavior from users. Penalties are assessed if a user violates the terms of a negotiated contract. The definition of maintaining good behavior depends on the degree of flexibility the user has to alter the service provided once it has started. There are two possible approaches, *static service negotiation* and *dynamic service alternation*. With static service negotiation, the user and SP negotiate the terms of the contract at the start of the session. Rewards and penalties are assessed at the beginning of the session. Once the session has been initiated, the user has no scope to alter service except through re-negotiation with SP. With dynamic service alteration, the user and SP negotiate the terms (i.e. trading rules) of the session. The user is expected to adhere to the terms of the negotiated contract, but may change conditions if desired and receive a penalty. In this case, rewards and penalties can be assessed only at the end of the session. A level-based reward algorithm (Figure 7) calculates the reward that a user obtains for maintaining a given level of service and the penalties assessed for violation of service constraints.

5 Related Work

Majority of QoS research has concentrated on analyzing network resource consumption and translations between network service demands and network service parameters, For example, [15] illustrates similarly how to design an architecture to provide end-to-end QoS for applications distributed across a network. In the Omega architecture, QoS parameters are translated between application and network by a QoS broker [15]. [10] presents an ODP view of QoS and proposes an *environment contract* to specify QoS characteristics of computational objects. The authors also provide a translation of these specifications to ATM related QoS parameters. These network parameters incorporate time related characteristics that record time delay (cell interarrival time, cell delay variation, cell transfer delay),and capacity related characteristics that record communication throughput (for

e.g. cell transfer capability, mean cell rate), guarantee levels, channel availability(priority level, cell loss probability) and accuracy (for e.g. cell sequence integrity, cell loss ratio, cell error ratio, cell misinsertion rate, severely-errored cell block ratio, cell insertion rate, bit error probability). The TENET project [1, 7] classifies services into deterministic service, statistical service, predicted service and feedback based schemes to deal with tradeoffs in QoS, network utilization and overload. In [17], a method for modifying the HTML protocol to reduce communication latencies is proposed. In collaboration with SANDIA national labs, the work characterizes video-conferencing applications (trying to understand network support for such applications) and determines QoS parameters for MM networking. [2] describes protocols for connection establishment for supporting real-time multiparty applications like video conferencing with guaranteed QoS. [12] addresses issues of distribution and duration of metrics like call-completion rate in the telecommunication world.

Many approaches for QoS support refer to an existing infrastructure or environment. For example, systems that focus on QoS management and its mapping on to the transport system include the Heidelberg High Speed Transport System [27], and the Lancaster QoS architecture [4]. The approach presented in [5] deals with QoS support in a heterogeneous environment with diverse communication requirements, varying levels of QoS in terms of latency, bandwidth, jitter etc. The need for maintaining a uniform view of the entire system to account for heterogeneity in the network model, request model etc. is emphasized. Mechanisms discussed include the scaling of media streams in terms of picturesize, content,picture rate, cost etc. as an effective technique to deliver acceptable perceptive QoS and avoid overloading a system with unnecessary information.

However, there is less research in the area of determining user-satisfaction. [11] discusses the multilevel specification of QoS factors like synchronization, interactivity, availability that are applicable to a range of applications. The quality function presented in [24], is a comparable effort in providing parameters of user-satisfaction. The approach focuses on preserving fidelity of presentation and hence formalizes presentation features that correspond to parameters in the US function such as jitter and end-to-end delay. An explicitly defined presentation error model serves to assign consistent values to quality parameters that can be used to define the weights represented in the metric we propose in this paper. While the approach proposed by [22, 23] is device and implementation independent, the weighted cost-satisfaction ratio adds (a) a notion of devices and resource consumption and (b)an attributed cost to a service request.

6 Conclusions and Future Directions

In this paper, we have developed, based on empirical results from 3 systems (the VOSAIC system, the hierarchical VOD system and the remote VCR system) an integrated video QoS metric - the *weighted cost-satisfaction ratio*. We discussed in detail the economic relation between the user satisfaction and resource consumption as well as the advantages of providing unified metrics for multimedia services within an economic framework. Due to the close tie between the empirical results in Section 3 and the modeling of the QoS metric in Section 4, the design of our metric-based QoS architecture implies enhanced performance and cost-effectiveness for the user and SP.

We are exploring the use of the proposed metric in the design of admission control policies and resource management algorithms for multimedia systems. Specifically, we are interested in developing adaptive policies that use the metric to measure performance upgrades and degradations, determine the sources of bottlenecks, apply trade-off techniques and possibly adapt parameters of service. We also continue to investigate the formalization of the economic framework, the optimization relations and its implications in the design of effective metrics for multimedia systems. Our current work focuses on extending the proposed metric to more generalized Internet and Web server environments by exploring any additional parameters such as signal fidelity, locality, cache size and security that may be needed in these environments.

Acknowledgements

This work was supported by Hewlett-Packard and the National Science Foundation CAREER Grant under CCR-96-2386. The authors would like to thank Yasuhiko Miyazaki and Lintian Qiao for their contribution to the implementation of the hierarchical VOD system and measurements of its performance characteristics. The authors would also like to thank the VOSAIC team for supplying logfiles that provided data for the empirical study and Richard Friedrich for useful discussions during the course this work.

References

[1] A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma, and H. Zhang. The tenet real-time protocol suite: Design, implementation and experiences. *IEEE Transactions on Networking*, 4(1):1–10, February 1996.

[2] R. Bettati, D. Ferrari, A. Gupta, W. Heffner, W. Howe, M. Moran, Q. Nguyen, and R. Yavatkar. Connection establishment for multi-party real-time communication. In Thomas D.C. Little and Riccardo Gusella, editors, *4th International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 95 Durham, New Hampshire, USA*, pages 240–250, April 1995.

[3] Z. Chen, S. Tan, R. H. Campbell, and Yongchen Li. Real time video and audio in the world wide web. In *World Wide Web Conference, Boston, MA*, 1995.

[4] Francisco Garcia, David Hutchison snd Andreas Mauthe, and Nicholas Yeadon. Qos support for distributed multimedia communications. In *1st International Conference on Distributed Platforms, Dresden, Germany*, March 1996.

[5] Tino Hutschenreuther and Sascha Kuemmel. Ina-qost - integrated network architecture for qos-based transmission in heterogeneous environments. In *4th International IFIP Workshop on Quality of Service, IwQos96 Paris, France*, March 1996.

[6] K. Kawachiya and et al. Evaluation of qos-control servers on real-time mach. In Thomas D.C. Little and Riccardo Gusella, editors, *4th International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 95 Durham, New Hampshire, USA*, pages 117–128, April 1995.

[7] E. Knightly and P. Rossaro. Effects of smoothing on end-to-end performance guarantees for vbr video. In *International Symposium on Multimedia Communications and Video Coding, New York, NY*, October 1995.

[8] P. Konana, A. Gupta, and A.B. Whinston. Digital contract approach for consistent and predictable multimedia information delivery in electronic commerce. In *MMNC '97, SPIE Vol. 3020*, 1997.

[9] A. Krishnamurthy, T.D.C. Little, and D. Castanon. A pricing mechanism for scalable video delivery. *ACM Multimedia Systems*, 4:328–337, 1996.

[10] P. Leydekkers and V. Gay. Odp view on qos for open distributed mm environments. In *4th International IFIP Workshop on Quality of Service, IwQos96 Paris, France*, pages 45–55, March 1996.

[11] F.H.S. Lima and E.R.M. Madeira. Odp based qos specification for the multiware platform. In *4th International IFIP Workshop on Quality of Service, IwQos96 Paris, France*, pages 45–55, March 1996.

[12] M. Malhotra and M. Veeraraghavan. Position statement: Challenges in defining and evaluating qos metrics for communication services. In *4th International IFIP Workshop on Quality of Service, IwQos96 Paris, France*, pages 69–72, March 1996.

[13] R.E. McGrath. What we do and don't know about the load on the ncsa www server. In *http://www.ncsa.uiuc.edu/InformationServers*, September 1994.

[14] Yasuhiko Miyazaki. Hierarchical client-server multimedia systems. Master's thesis, University of Illinois at Urbana-Champaign, December 1996.

[15] K. Nahrstedt and J. M. Smith. The qos broker. *IEEE Multimedia*, 2:53–67, 1995.

[16] K. Nahrstedt and R. Steinmetz. Resource management in networked multimedia systems. *IEEE Computer*, 28(5):52–65, May 1995.

[17] Venkata N. Padmanabhan and Jeffrey C. Mogul. Improving http latency. Technical report, Digital Equipment Corporation Western Research Laboratory, 1994.

[18] Lintian Qiao and Klara Nahrstedt. Lip synchornization within an adaptive vod system. In *SPIE International Conference on Multimedia Computing and Networking (MMNC '97), San Jose, CA*, February 1997.

[19] Jerome Rolia and Richard Friedrich. Quality of service management for federated applications. In *4th International IFIP Workshop on Quality of Service, IwQos96 Paris, France*, March 1996.

[20] J. Sairamesh, D.F. Ferguson, and Y. Yemini. An approach to pricing, optimal allocation and quality of service provisioning in high-speed packet networks. In *INFOCOMM '95*, 1995.

[21] L. C. Schreier and M. B. Davis. System-level resource management for network-based mm applications. In Thomas D.C. Little and Riccardo Gusella, editors, *NOSSDAV 95, Durham, New Hampshire, USA*, pages 121–125, April 1995.

[22] Richard Staehli. *Quality of Service Specification for Resource Management in Multimedia Systems*. PhD thesis, Oregon Graduate Institute of Science and Technology, Portland, OR, January 1996.

[23] Richard Staehli, Jonathan Walpole, and David Maier. Device and data independence for multimedia presentations. In *Computing Surveys Symposium on Multimedia, 27:4*, pages 640–643, December 1995.

[24] Richard Staehli, Jonathan Walpole, and David Maier. Quality of service specifications for multimedia presentations. *ACM Multimedia Systems*, 3(5/6):251–263, nov 1995.

[25] R. Steinmetz and C. Engler. Human Perception of Media Synchronization. Technical Report 43.9310, IBM European Networking Center Heidelberg, Heidelberg, Germany, 1993.

[26] N. Venkatasubramanian and S. Ramanathan. Load management in distributed video servers. In *International Conference on Distributed Computing Systems (ICDCS 97)*, May 1997.

[27] C. Vogt and R. Herrtwich, R.G. snd Nagarajan. HeiRAT: The heidelberg rsource administration technique, design philosophy and goals. In *Conference on Communication in Distributed Systems, Munchen, Germany*, 1992.