Building an Adaptive Multimedia System using the Utility Model

Lei Chen[1], Shahadat Khan[2], Kin F. Li[3], and Eric G. Manning[3]

[1] Siara Research Canada, 305-8988 Fraserton Court, Burnaby, B.C., Canada V5J 5H8
glchen@siara.com
[2] Infranet Solutions, Suite 409-100 Park Royal, West Vancouver, B.C. Canada V7T 1A2
skhan@infranetsolutions.com
[3] University of Victoria, Victoria B.C., Canada V8W 3P6
{kinli, emanning}@sirius.uvic.ca

Abstract. We present our experience of building a prototype system based on the Utility Model for adaptive multimedia. The Utility Model is proposed to capture the issues and dynamics in multi-session multimedia systems where the quality of service (QoS) of individual sessions is adapted to dynamic changes of available resources and of user preferences. We present the design and implementation of our prototype multimedia system, and report experimental results. We demonstrate that the Utility Model supports two types of adaptation: reactive adaptation for systems where only a subset of the applications follows the adaptation model, and proactive adaptation for systems where all the applications follow the adaptation model. Our results demonstrate that the Utility Model may be effectively used for dynamic quality adaptation in real-time multimedia systems.

1 Introduction

Multimedia applications require system support with quality of service (QoS) guarantees. For instance, in order to support a video stream at a refresh rate of 10 frame/sec, the stream handler has to be scheduled onto the CPU once every 0.1 second. In a traditional video transmission system, it is possible that the session is getting sufficient network bandwidth, but cannot utilize this bandwidth effectively because the system cannot allocate sufficient CPU cycles to decode and render the video stream. Thus, to enforce QoS guarantees effectively, resources in a multimedia system must be managed in an integrated way. Designing an adaptive multimedia system with integrated resource management requires a comprehensive understanding of the problems, issues and dynamics within such a system.

In a dynamic system environment, multimedia applications are potentially adaptive. For example, if there is enough network bandwidth available, one can have real-time video transmitted at 30 frame/sec; otherwise, one may reduce the bandwidth requirement to half by reducing the rate to 15 frame/sec. How do we design a multimedia system to exploit the potential for adaptation of multimedia applications? In this arti-

cle, we present the design, implementation and experimental results for a prototype system based on the Utility Model for adaptive multimedia.

The rest of this article is organized as follows. Section 2 discusses some related work. Section 3 presents a brief overview of the Utility Model. Section 4 presents our prototype implementation. Section 5 presents our experiments and performance results. Finally, section 6 concludes the article.

2 Related Work

In recent years, there has been significant interest in the understanding of adaptive multimedia systems[1,3,7,8,9,11,12]. Schreier and Davis proposed the *Benefit Model* for adaptation of quality attributes of a single-user multimedia application[12]. The user expresses her media quality preferences by associating benefit functions with performance attributes such as audio quality, video frame rate and audio/video synchronization. The objective is to maximize the benefit by adjusting the amount of processing, communication and storage resources provided to the application.

Moser presented an Optimally Graceful QoS Degradation (OGQD) Model, where the realized QoS of a multimedia session is gracefully degraded to conform to changes in resource availability[9]. Suppose a multimedia service consists of several elementary services such as audio input, audio output, video input and video output. For each elementary service, the system has to find a suitable implementation and the operating QoS of that implementation, depending on available resources. The goal of the OGQD system is to provide multimedia service with minimum degradation from desired levels of QoS.

Since the Benefit Model and OGQD are both targeted to the adaptation problem of a *single* multimedia session, they do not capture issues peculiar to systems with multiple concurrent sessions. For example, they do not address the following:

1. how to specify and achieve the adaptation objective of a multi-session system,
2. how to incorporate the notion of relative importance of different sessions, and
3. how to deal with session admission.

In [4], we presented the Utility Model — a mathematical model to capture the issues of resource management in adaptive multimedia systems with multiple concurrent sessions, where the quality of individual sessions is dynamically adapted to the available resources and to the run-time user preferences. In this paper, we present our experience with a prototype multimedia system designed using the Utility Model.

In the context of operating system design to support continuous media, Kawachiya *et al*. of the Keio Multimedia Project have proposed a processor execution model, called Q-Thread, where a periodic continuous media task can specify the tolerable ranges for the period of scheduling and computation time in each period, and the system controls the operating quality, specified by period and computation time, of tasks dynamically[3]. One of the main challenges for such a system is as follows: At a given system state, how does the system find an appropriate operating quality (such as period and computation time) for each task? Our proposal of the Utility Model is an effort to address this issue.

McCanne proposed an adaptive multimedia transport system using the Receiver-driven Layered Multicast (RLM)[8]. In RLM, a multimedia source transmits a stream into multiple subscription groups, each with different components of the stream, and receivers adapt the quality of reception by joining and leaving subscription groups. We use the concepts of RLM for implementing a video-on-demand application in our prototype system.

3 The Utility Model

Figure 1 illustrates an adaptive multimedia system (AMS) where *n* sessions share *m* resources. When sufficient resources are available, the system should provide every session the highest desired quality for everyone. However, when available resources cannot support the highest desired quality, the system must compromise the operating quality of certain sessions. For example, when a higher-priority video-conference session needs better quality, compromising the quality of a lower-priority virtual environment session is in order. For a video session, this may mean reduced refresh rate or reduced resolution, and for a virtual environment, this may mean lower levels of detail of three dimensional objects.

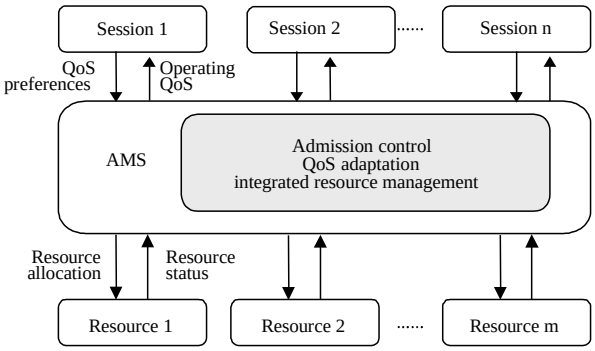

Figure 1: Adaptive multimedia system requirements.

Adaptation in a multimedia system may be triggered by many events, such as change of system load, change of network load or congestion/failure in the network, and dynamic change of users' requirements. An AMS has the following requirements:

- The users must be able to specify their quality requirements/preferences in a flexible way.
- The system must know the current status of system resources.
- The system must have the option to accept or to reject new session requests.
- The system must provide quality adaptation and integrated resource management in order to make best use of the available resources.

In order to capture the issues of resource management in adaptive multimedia systems with multiple concurrent sessions, where the quality of individual sessions is dynamically adapted to the available resources and to the run-time user preferences,

we proposed a mathematical model, called the Utility Model[4]. It provides a unified and computationally feasible way to solve the admission problem for new multimedia sessions, and the dynamic quality adaptation and integrated resource allocation problems for existing sessions.

3.1 Concepts

The Utility Model is based on the concepts of quality profile, quality-resource mapping, session and system utility, and system resource constraints.

Quality Profile

The quality profile of a session is a sequence of acceptable operating qualities in increasing order of preference (from minimum acceptable quality to highest desired quality). For example, suppose a multimedia session consists of three media: audio, video and still image. Table 1 shows a possible set of user choices where the session may have one of three operating qualities: gold, silver or bronze. Suppose user *i*'s quality profile is expressed as $\mathbf{P}_i=(\boldsymbol{q}_3, \boldsymbol{q}_2, \boldsymbol{q}_1)$ where $\boldsymbol{q}_3$ is the highest desired quality, and $\boldsymbol{q}_1$ is the worst quality the user is ready to accept and pay for. Figure 2(a) shows the user profile where each operating quality is represented as a vector of three media qualities: audio quality, video quality and image quality. The quality profile provides a flexible way to specify the quality requirements and preferences of the user of a particular session.

Quality-resource Mapping

We assume the existence of a unique mapping from an operating quality to the resources required to provide that quality. Suppose $\boldsymbol{q}_i$ represent the operating quality of session *i*. Then the resources required by this session are given by $\boldsymbol{r}(\boldsymbol{q}_i)$ where $\boldsymbol{r}(\cdot)$ represents the quality-resource mapping function. For example, suppose a system has three resources — CPU cycles, system RAM, and network bandwidth. Quality $\boldsymbol{q}_3$ may be mapped to 50% of server CPU cycles, 1 Mbyte of server RAM, and 5 Mbps network bandwidth; whereas quality $\boldsymbol{q}_1$ may be mapped to 10% of server CPU cycles, 10 Kbyte of server RAM, and 1 Mbps network bandwidth. (Quality-resource mappings are addressed in research projects at Keio University. In [10], Nishio *et al*. present a quality-resource table based on measurements of the RT-Mach system.)

Table 1: Flexible choice for the user of adaptive multimedia system.

	Bronze (q_1)	Silver (q_2)	Gold (q_3)
Audio	Mono	Stereo	Surround
Video	Low resolution, no color	High resolution, no color	High resolution, Color
Image	Low resolution	Medium resolution	High resolution

Session and System Utility

Any adaptive system must have an objective, which decides the direction of adaptation in any given situation. We assume that the objective of an AMS is to maximize *system*

utility, which is a weighted sum of the individual *session utilities*. The concept of utility is taken from economics, where it means the capacity of a commodity or a service to satisfy some human want, and the goal of an economic system is to allocate resources to maximize utility.

We assume the existence of an utility function which maps a session's operating quality $\boldsymbol{q}_i$ to session utility $u_i(\boldsymbol{q}_i)$. For simplicity, we assume that system utility objective is to maximize the system utility function U given by:

$$U = \Sigma\, u_i(\boldsymbol{q}_i). \qquad (\mathbf{1})$$

For example, in a multimedia service provider application, the session utility may be the revenue paid by individual sessions (such as $100 for gold, $60 for silver and $40 for bronze), and the system utility is the total revenue earned by the system.

System Resource Constraints

At any moment, the system state has to obey the following *system resource constraints*: For each resource, the sum of the quantities of the resource allocated to all the sessions cannot exceed the total available quantity of the resource. Suppose the available system resources are expressed as a vector $\boldsymbol{R}$. The resource constraints are expressed as

$$\Sigma\, \boldsymbol{r}(\boldsymbol{q}_i) \le \boldsymbol{R}. \qquad (\mathbf{2})$$

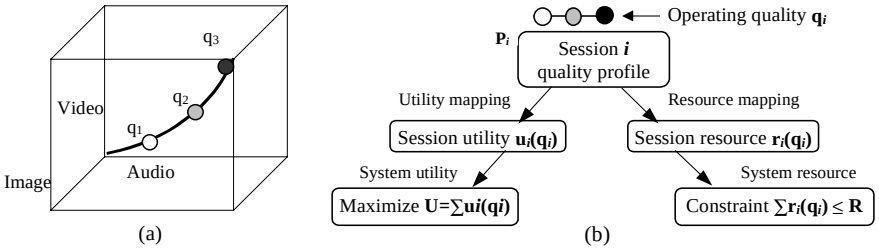

Figure 2: (a) Quality profile and (b) the main concepts of the Utility Model.

3.2 Using the Model

The main concepts of the Utility Model are illustrated in Figure 2(b):

- Each user specifies a quality profile, set of acceptable session operating qualities.
- A session's operating qualities are mapped uniquely to required resources.
- A session's operating qualities are mapped uniquely to session utilities.
- The system utility is the sum of all session utilities.
- The system is subject to the system resource constraints.

Now the main problem in an adaptive multimedia system is as follows: *Find* the operating quality $\boldsymbol{q}_i$ of each session *i* which *maximizes* the system utility U under the system *resource constraints*. In [4], we show that this problem can be mapped to a multiple-choice multi-dimension knapsack problem (MMKP). We also provide two

solutions for the MMKP: an exact algorithm BBLP using branch-and bound searching, and a heuristic HEU for fast and approximate solution. The latter solution is suitable for time-critical applications such as real-time multimedia systems.

The Utility Model may be used in session admission, dynamic quality adaptation and integrated resource management in real-time multimedia systems.

- *Session Admission*: Suppose the system has *n* sessions, and a user requests another session. Provided that there is enough system resources available for the new session, the achievable system utility *U'* with (*n*+1) sessions is compared to current system utility *U* with *n* sessions. If *U'>U*, then the new session should be admitted; otherwise the new session should be rejected.
- *Quality Adaptation*: Suppose a session is dropped or the amount of resource available is changed due to external/uncontrollable reasons. The system computes the solution of the Utility Model under the new constraint, and some of the existing sessions may have to change their operating qualities.
- *Integrated Resource Management*: Since the Utility Model considers allocation of all the system resources in an integrated way, it would avoid situations such as a session getting sufficient network bandwidth, but cannot utilize this bandwidth effectively because the system cannot allocate sufficient CPU cycles to decode and render the video stream.

4 Prototype Implementation

To demonstrate and test our work, we designed and implemented a prototype system based on the Utility Model. The system consists of an off-the-shelf Linux operating system, a *Utility Model Engine* (UME), and some applications. The UME[1] implements the core functionality defined by the Utility Model (see Figure 3): it manages system resources through *resource monitoring*, *admission control*, and *quality adaptation*. To facilitate the interaction among applications and the UME, an application level proxy, QoS Agent, is also defined.

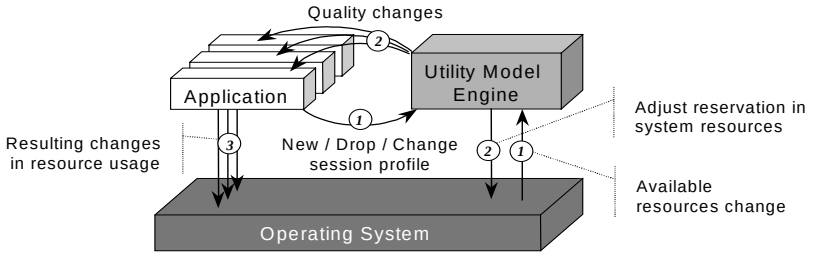

Figure 3: Components and interactions in the prototype system.

In our prototype system, applications are *admitted*, system resources are *monitored*, and applications' operating qualities are dynamically *adapted*. The admission

[1] For performance reasons, HUE is implemented in this prototype system, but BBLP is not.

control requires an application to submit a quality profile to the UME. The profile, maintained by a QoS Agent, encapsulates *quality-utility*, *quality-methods* and *method-resources* mappings. The quality-utility mapping assigns a dollar value to each of the three quality levels: *gold*, *silver*, and *bronze*; the quality-methods mapping associates quality levels with implementations that deliver them; and finally, method-resources mapping defines the resource requirements for each implementation. An application is admitted only if the system will yield a higher utility value as a result.

For implementation convenience, only three system resources are monitored: *CPU* (in percentages of cycles per second), *inbound bandwidth* and *outbound bandwidth* (in packets per second) of the network interface. Since we are constrained by the lack of reservation capabilities of the operating system, we allow 5% of the *raw* system resources to be used by system functions, while the remaining 95% is *manageable* by the UME. At times of admission control, the UME *reserves* the amount of resources required for applications' minimal admitted quality; during quality adaptation, the UME *allocates* the amount of resources for applications' operating quality.

The goal of the UME is to maximize the total system utility, while obeying user preferences and system resource constraints (i.e., not over-subscribing total available resource). Specifically, it needs to adapt the amount of manageable resource, based on the monitored resource usage, in order to avoid contention for resource use[2] The UME implements the following algorithm for adapting manageable resources and avoiding usage contention:

if ($monitored_t < allocated_t$) *then* $manageable_t = 95\% * raw_t$ *(i.e. initial condition)*
else if ($monitored_t < reserved_t$) *then* $manageable_t = raw_t - (monitored_t - allocated_{t-1})$
else if ($monitored_t == raw_t$) *then* $manageable_t = raw_t - (monitored_t - allocated_{t-1}) - CR$[3]
if ($manageable_t < reserved_t$)[4] *then* $manageable_t = reserved_t$
do quality adaptation and calculate new $allocated_t$

With the above mechanisms in place, our system is able to provide reactive adaptation in a best-effort mixed environment, and to enforce proactive adaptation in an entirely reservation-based system. The results of the following section will demonstrate this point.

5 Experiments and Results

We experimented with two kinds of systems: a best-effort mixed system and an entirely reservation-based system. In *a best-effort mixed system*, (e.g., the Internet) only a subset of the applications follows the adaptation model, whereas *in a reservation-based system*, all the applications follow the adaptation model. Applications not fol-

---

[2] Contention is possible since we are using a best-effort system where some applications are out of the Utility Model domain.

[3] CR (*Cushioning Region*), defined as 5% of the raw resource in our system, decides the aggressiveness of the UME in scaling back the manageable resource to avoid contention.

[4] Note that because the UME must guarantee the minimal qualities of admitted applications, the manageable resources are not allowed to reduce beyond the reserved resources.

lowing the Utility Model may cause resource contention. When this happens, we want the UME to scale back the quality of the applications following the model (see Figure 4a). In an entirely reservation-based system, the UME will adapt operating qualities of the applications in order to optimize the system utility (see Figure 4b).

Our experiments use two types of applications: a file transfer application and a video-on-demand application. For the case of a best-effort mixed system, the common ftp program is used together with a video-on-demand application (VoD[5]). A similar setup is used for an entirely reservation-based system, but with the ftp program replaced by a rate-controllable file transfer program (QFTP[6]). In the latter case, both applications conform to the Utility Model, and the receivers adjust their operating qualities based on adaptation signals received from the UME.

The topology of our experiments involved two server hosts (A and B), and one client host (C). The UME is implemented only at the client host, where VoD and FTP clients run. For simplicity, the inbound bandwidth is considered to be the only constrained resource in our experiments.

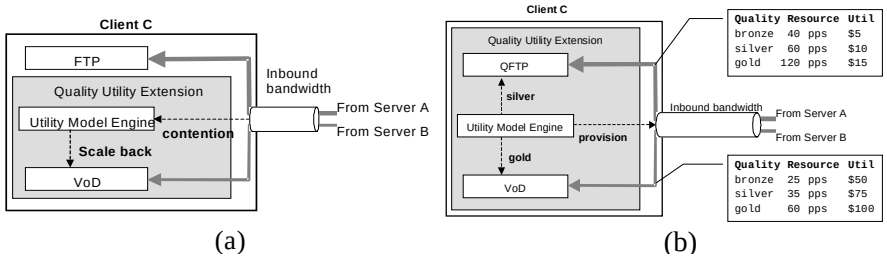

Figure 4: (a) Best-effort mixed system and (b) Reservation-based system.

5.1 Best-effort Mixed System

Figure 5 shows the adaptation events in this experiment. We note that the UME adapts to changes of system state by scaling the amount of manageable resource and dynamically adapting the resource allocation to the VoD application. We observe the following:

- Before event A, the UME monitored some background network activities. It adjusted the amount of manageable resource, so that resource would not be oversubscribed when admitting new sessions.
- At event A, the VoD application was admitted (not started!). The UME reserves the bandwidth required for its minimum quality (bronze) so that it can deliver the minimum quality as long as the session lives. However, since there is no contention, the UME allocates bandwidth for its desired quality (gold). The monitored resource, at event A, shows the resource consumption of VoD; the amount of manageable resource remained steady in the absence of resource contention.

---

[5] In VoD, a client receives a hierarchical layered video stream over some distinct IP multicast groups. Video quality varies as the client incrementally joins or leaves the groups.

[6] Unlike FTP, which uses the slow-start algorithm, a QFTP client controls the server's transmission rate, thus varying the transfer throughput quality.

- At event B, the FTP application is started. Our resource monitor shows that this application seizes large amounts of network bandwidth. To avoid contention, the UME scaled back the amount of manageable resource, and degraded the operating quality of VoD to bronze level by reducing its bandwidth allocation.
- At event C, the FTP terminated, and the UME scaled up the amount of manageable resource to 95%, and upgraded the operating quality of VoD.

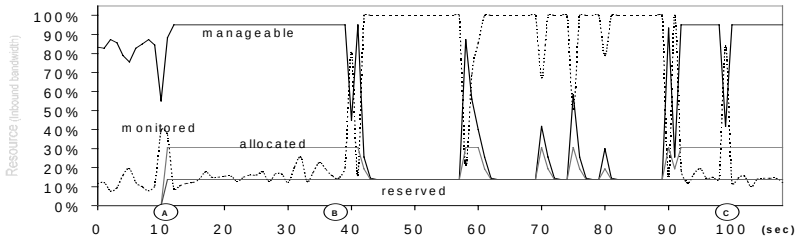

Figure 5: Scaling manageable resource and quality adaptation[7].

5.2 Reservation-Based System

Figure 6 shows the resource allocation and quality adaptation of the system during our experiment. Since both applications followed the adaptation model, the UME allocated the system resource in a systematic way; there is no need to scale back the manageable resource. We observe the following events:

- At event A, the UME admitted VoD, and allocated resource for gold quality.
- At event B, the UME admitted QFTP. Since the available resource is not enough to support gold quality for both applications, the UME has allocated resource for silver quality for QFTP. At this time, VoD gets gold quality because it has offered more for the gold quality.
- Now at event C, QFTP raised its offer for gold quality to $150. The UME adapts to this change by upgrading the operating quality of QFTP to gold, and by degrading the operating quality of VoD to silver.
- Finally at event D, application QFTP terminates, and frees the resource allocated to it. The UME adapts to this by upgrading the operating quality of VoD to gold.

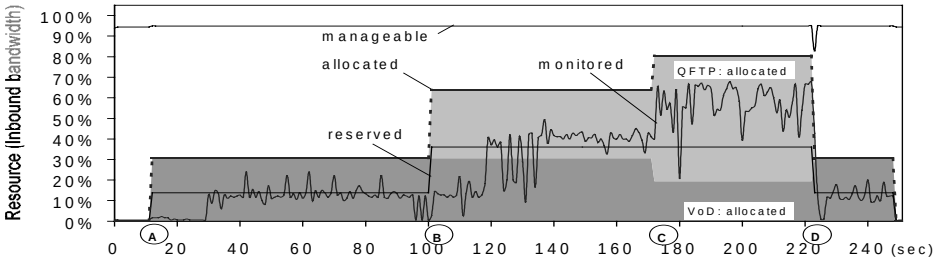

Figure 6: Optimal quality adaptation in a reservation-based system.

[7] The area above the manageable line represents unmanageable amount of resource.

6 Conclusions

We have presented the design and implementation of a prototype system using the concepts of the Utility Model for adaptive multimedia. We reported experimental results with two types of adaptation: reactive adaptation in a best-effort mixed system and proactive adaptation in an entirely reservation-based system. The design and implementation of the prototype system, and the performance results validated that the Utility Model can be effectively used for session admission and dynamic quality adaptation in practical real-time multimedia systems.

7 Reference

1. A. Campbell, G. Coulson, and D. Hutchison, “Supporting Adaptive Flows in Quality of Service Architecture”, ACM Multimedia Systems Journal, 1996.
2. Lei Chen, “Utility Model Applied to Layered-coded Sources”, M.Sc. Thesis, Department of Computer Science, University of Victoria, October 1998.
3. Kiyokuni Kawachiya and Hideyuki Tokuda, “QOS-Ticket: A New Resource-Management Mechanism for Dynamic QOS Control of Multimedia”, In Proceedings of Multimedia Japan '96, April 1996.
4. Shahadat Khan, "Quality Adaptation in a Multisession Multimedia System: Model, Algorithms and Architecture", Ph.D. Dissertation, Department of ECE, University of Victoria, May 1998.
5. Shahadat Khan, Kin F. Li and Eric G. Manning, "Padma: An End-System Architecture for Distributed Adaptive Multimedia", IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, Victoria, August 1997.
6. Shahadat Khan, Kin F. Li and Eric G. Manning, "Optimal and Fast Approximate Solutions for the Multiconstraint Multiple-choice Knapsack Problem", Technical Report ECE-97-3, Department of ECE, University of Victoria, September 1997.
7. Shahadat Khan, Kin F. Li and Eric G. Manning, "The Utility Model for Adaptive Multimedia Systems", Presented at The International Conference on Multimedia Modeling, Singapore, November 1997.
8. Steve McCanne, Van Jacobson and Martin Vetterli, M., “Receiver-driven Layered Multicast”, ACM SIGCOMM, Stanford, CA, August 1996.
9. Martin Moser, “Declarative Scheduling for Optimally Graceful QoS Degradation”, In Proceedings of IEEE Multimedia Systems, Hiroshima, Japan, 1996.
10. Nobushiko Nishio and Hideyuki Tokuda, QOS Translation and Session Coordination Techniques for Multimedia Systems”, In Proceedings of Sixth International Workshop on Network and Operating System Support for Digital Audio and Video, Jushi, Japan, 1996.
11. Klara Nahrstedt, “An Architecture for End-to-End Quality of Service Provision and its Experimental Validation”, Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, August 1995.
12. Louis C. Schreier and Michael B. Davis, “System-level Resource Management for Network-based Multimedia Applications”, In Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, NOSSDAV 95, Durham, NH, April 1995.