A Market-based Resource Management and QoS Support Framework for Distributed Multimedia Systems *

Wonjun Lee
School of Engineering
Information & Communications University
P.O.Box 77, Yusong, Taejon,305-600, Korea

Jaideep Srivastava
Dept. of Computer Science & Engineering
University of Minnesota - Twin Cities
Minneapolis, MN 55455, USA

Abstract

In this paper we present a welfare economic (market-based) resource management model that is QoS-based, which models the actual price-formation process of an economy. This approach manages resource and QoS allocation optimally so that the total utility of the system is maximized through a tatonnement process, in which operating markets for each resource is done separately. We use the constructs of application benefit functions and resource demand functions to represent the system configuration and to solve the resource allocation problems.

Keywords: Economic Framework, Quality of Service, Multimedia, Resource Management, Distributed Systems

1 Introduction

The last decade has seen an explosive growth in multimedia applications and considerable research in related technologies, with special emphasis on Quality of Service (QoS) requirements, such as timeliness, precision, and accuracy [12]. Meeting QoS guarantees in a distributed real-time multimedia systems is an end-to-end issue because the users are interested in the end results; i.e., from application to application. Allocating proper resources to the applications with respect to QoS provisioning to maximize the total system utilization or benefit, is a fundamental problem in all multimedia systems today. The term *utility* or *benefit* may take on the meaning of usefulness, satisfaction, or of pleasure, depending on the context. We may treat utility as something which can be measured in the sense that one can say how much someone would give in order to obtain the utility of a good or service [16]. Hence, there is a continuing need to develop better resource management techniques for such systems. For this purpose, we need a resource allocation model for QoS management in which each application is associated with one of multiple levels of performance, and each level is characterized by a set of QoS parameters that are service specific.

The computational economy provides one type of mechanism for allocating limited resources in such an environment in a distributed, dynamic way. Economic principles such as rationality and efficiency have been used implicitly in artificial intelligence [21, 5, 28] for many years. The use of economic principles in distributed resource allocation based on QoS requirements is a comparatively recent development. The applications of economic concepts include the use of ideal resources in a network and approximating solutions of complex problems by transforming the problem into a *general equilibrium* framework. Using competition and a price system to allocate resources has many benefits, including limited complexity, decentralized decision making, and dynamic adjustment.

Economists have established a well-tried solution to the scarce resource allocation problem, namely the use of *markets*. Markets can achieve optimal allocations under many circumstances with little or no central guidance. That is, decentralization is provided in an economy by the fact that economic models consist of participants (agents) which only think of themselves, and the equilibrium allocation will very often be optimal, or very close to optimal. In an economy, there are two types of agents: (1) a consumer which attempts to optimize its individual performance criteria by obtaining the resources it requires, and is not concerned with system-wide performance; and (2) a supplier which allocates its individual resources to consumers in such a way that its individual profit

*This work is supported by U.S. Army Research Lab number DA/DAKF11-98-9-0359 to the University of Minnesota.

CIKM 2000, McLean, VA USA
© ACM 2000 1-58113-320-0/00/11 . . .$5.00

(benefit) derived from its choice of resource allocations to consumers is optimized. For this, the *pricing* system is introduced as the technique for coordinating the selfish behavior of agents. The price a producer charges for a resource is determined by its supply and the demand of the agents for the resource. It ensures that a realizable allocation of resources is achieved. From the agents' perspective, the state of the world is completely described by the going prices. In other words, the prices determine the maximizing behaviors through an *Invisible Hand* [26]. This arrangement is extremely modular; therefore, agents need not expressly consider the preferences or capabilities of others. Henceforth, markets have significant advantages over central allocation schemes, some of which are mentioned in the following section.

In this paper, we propose a *market-based mechanism*, which allows for efficient resource allocation along multiple QoS dimensions, in the presence of distributed resources. We also present a generic resource allocation model which is mathematically proven.

This paper is organized as follows. In Section 2, we discuss recent research with respect to QoS management and resource allocation. Section 3 provides the basic definitions, attributes, and assumptions used for our QoS-based resource management model, as well as the *Resource Demand Function (RDF)*, and the *Benefit Function (BF)*. In Section 3, we will also discuss QoS allocation problems in terms of resource and QoS dimensions. Section 4 presents an algebraic modeling for both single and multiple resource environments, and its relevant theorems and algorithms. In Section 5, we explain the market-based approach for solving constrained global optimization problems. In Section 6, we present our concluding remarks and discuss problems that remain unsolved.

2 Related Work

A considerable amount of research has been carried out within the field of QoS support for distributed multimedia systems and resource management, in terms of QoS allocation to satisfy specific application-level requirements. Such work can be classified into various categories. The first category involves systematic approaches, which allocate appropriate resources to achieve a specific level of QoS for an application. Next, analytical (algebraic) models for QoS management satisfy application requirements. Lastly, we will investigate the well-understood class of market-based mechanisms for resource allocations for a set of computational agents by computing the competitive equilibrium of artificial economies.

2.1 Systematic Approach

Abdelzaher et al. [1] describe a distributed pool of processors with which timeliness for real-time applications using admission control and load-sharing, is guaranteed. In the *Rialto* O/S [18], a modular O/S approach is presented, the goal of which is to maximize the user's perceived utility of the system, instead of maximizing the performance of any particular application. Other flexible QoS systems are being developed; such as the *SMART* scheduler [22], which autonomously changes the resource allocations given to the applications; and the *Processor Capacity Reserves* in *RT Mach* [10], which change the allocations as a result of an explicit request by the applications. Jensen's work [6] in Benefit-based scheduling is relevant to this view. Jensen proposed soft real-time scheduling, based on application benefit, where the goal is to schedule the applications so that the system can maximize the overall system benefit. The above-mentioned schemes are lacking in their theoretical basis to maximize the system benefit (utility). Most systems are only based on simply increasing/decreasing the benefit (utility) functions through the execution level information and the application state information, and no consideration for multiple resources is provided.

2.2 Analytical Approach

Recently, control theories have been examined for QoS adaptation. DeMeer [4] proposed a control model for adaptive QoS specification in an end-to-end scenario, and Satyanarayanan [25] suggested the application of control theory as a future research direction to analyze adaptation behavior in wireless environments. Nahrstedt [17] proposed a control-based middleware framework to enhance QoS adaptations by dynamic control and reconfigurations to the internal functionalities of a distributed multimedia application. Rajkumar et al. [23] propose a *QoS-based Resource Allocation Model (Q-RAM)*, which assumes a system with multiple concurrent applications, each of which can operate at different levels of quality based on the system resources available to it. Their goal is to allocate resources to the various applications such that the overall system utility is maximized, under the constraint that each application can meet its minimum needs. However, their greedy algorithm to obtain a good resource allocation for each application in a system with all linear dimensional utility functions does *not* always lead to an optimal resource allocation (i.e., it is suboptimal). Also, they only deal with single resource environments. Our prior work [14] was most closely related to their work, but was ex-

tended in several ways. First, ours guarantees to find the optimal solution mathematically. Second, they only deal with single resource environments, while our model can characterize the resource allocation for both single and multiple resources.

2.3 Economic Approach

Economic mechanisms have been mainly considered in distributed AI research [28], and recently in the distributed computing community for allocating computational resources of various kinds [3, 8, 11]. These applications are centered around a global model for the resource, from which each agent or module calculates the marginal value of the resource for itself. In our previous work on *admission control and QoS negotiations for soft-real time applications* [13], we used *Market based schemes* [3] for computing the optimal resource allocations, in terms of benefit and resource demand curves, to redistribute the resources among the degraded applications. Here, the measure of performance that we try to maximize is the total benefit of the system, and we trade off the marginal benefit of each application for the increment in resource allocation. The dual of the constrained optimization problem derived by using the method of *Lagrange multipliers* can be used to compute the solution. The free-market-based auction methods solve the dual problem, and the Lagrange multiplier is the *price* of the resource fixed by the auction [24].

3 QoS-based Resource Management Model

3.1 Model Definition

In multimedia systems such as Video-on-Demand (VOD) and Continuous Media servers, applications arrive with a request for a certain amount of resources. In our formulation of the problem, this information is provided to the system via the *resource demand function* [24]. The function describes the resource demand for different QoS settings of the application. For example, a streaming video application will represent the resource demand as a function of the frame rate, the frame size and the frame quality [24]. We generalize this concept to multiple resource levels.

In addition to the resource demand function, all the applications specify a *benefit function* [24] that describes the relative benefit between the different QoS parameter settings for the application. This function is useful to trade off between the QoS parameters when there are constraints on the resources [24]. For example, the goal of admission control and the QoS negotiation process is to maximize the total benefit over all the applications [13]. The degraded versions of the applications use fewer resources. The benefit functions and the resource demand functions essentially encode the information about how the different levels of the application quality parameters compare with each other in terms of the benefit to the user and the usage of the resources. This is not really a restrictive assumption, as the applications that do not have different levels of quality parameters can still be modeled as benefit functions that are zero everywhere, except at the parameter setting requested by the application [13].

In our model, we assume that the system consists of n applications $\{a_1, a_2, ..., a_n\}$ and m resources $\{R_1, R_2, ..., R_m\}$. CPU cycles, the disk bandwidth, the buffer available, and the network bandwidth are the resources. Each application is also specified by QoS parameters $\{q_1, q_2, ..., q_l\}$. We introduce the following definitions and notation:

- *QoS vector,* $\vec{q}$: a vector consisting of QoS parameters. This is used as an index for a resource demand function $R(\vec{q})$.
- *Resource demand,* x: an amount of resource requested (or allocated) to each application is calculated by a resource demand function $R(\vec{q})$. To represent the resource for application j, we use the subscript j, i.e. x_j. For the multiple resource allocation problem, x_{ij} denotes the amount of resource i requested (or allocated) to application j.
- *Total available resources,* R_{total}: the total available resources are represented by R_{total}. In multiple resource allocation, we use a subscript i to denote *resource* i, i.e., R_{total_i}.
- *Total system benefit,* B_{total}: the total system benefit which is achieved by the optimal resource allocation is denoted as B_{total}.
- *Cost function* $C(.)$: the negative function of $B(.)$, i.e., $C(.) = -B(.)$. The objective value (function) in our mathematical model is $B(.)$. It can be interpreted as the profit or reward incurred by the resulting resource allocation. For algorithmic convenience, we will convert the problem of *maximizing* $B(.)$ to a problem of *minimizing* $C(.)$. They essentially have the same result because *maximizing* $B(.)$ is equal to *minimizing* $-B(.)$.

Further details about the resource demand functions and the benefit functions will be described in

the following subsections. Table 1 summarizes the attributes used in our model.

Attributes	*Descriptions*
a_i	application i: $a_1, a_2, ..., a_n$
n	number of applications
R_i	resource i: $R_1, R_2, ..., R_m$
m	number of resources
R_{total}	total available resources (positive real constant): $\sum_{i=1}^{m} R_{total_i}$
R_{total_i}	total available i resources (positive real constant)
q_j	QoS parameter for application j
q_{kj}	k^{th} QoS parameter for application j, where $k \in [1..l]$
q_{kij}	k^{th} QoS parameter for application j on resource i, where $k \in [1..l]$
$\vec{q}$	QoS vector with QoS parameters: $\vec{q} = (q_1, q_2, ..., q_l)$
$\vec{q}_j$	QoS vector for application j with QoS parameters: $\vec{q}_j = (q_{1j}, q_{2j}, ..., q_{lj})$
l	number of QoS dimensions
x_{ij}	amount of resource i allocated to application $j(= a_j)$
x_j	amount of resource allocated to application $j(= a_j)$: $\sum_{i=1}^{m} x_{ij}$
$R_j(\vec{q}_j)$	Resource Demand Function (RDF), which maps into an x_j in a *single* resource: $R(\vec{q}_j) = x_j, \vec{q}_j = (q_{1j}, ..., q_{lj})$, where $j \in [1..n]$
$R_j(\vec{q}_{ij})$	Resource Demand Function which maps into an x_{ij} in *multiple* resources: $Q(\vec{q}_{ij}) = x_{ij}, \vec{q}_{ij} = (q_{1ij}, ..., q_{lij})$, where $i \in [1..m]$, $j \in [1..n]$
$B_j(x_j)$	system benefit for application j
$B_j(\sum_{i=1}^{m} a_{ij}x_{ij})$	system benefit for application j on resource i, where a_{ij} are positive real constants
B_{total}	total system benefit
$C_j(.)$	Cost function; i.e., negative function of $\alpha_j \times B_j(.)$
α_j	priority for application j; i.e., relative importance between n applications

Table 1: Attributes used in the Model

In our model, we make the following assumptions:

- Applications are independent of one another.
- Benefit Functions (Object Functions), $B_j(.)$, are *nondecreasing (increasing)* in each type of resource, and are thus *concave*.
- $B_j(x_j)$ is *continuously differentiable* over an interval including [0,R_{total}] at x_j.

3.2 Resource Demand Function

Given a QoS vector $\vec{q}_j$, for an application j with the dimension of QoS parameters being l, the Resource Demand Function is given as:

$$R_j(\vec{q}_j),\ where\ \vec{q}_j = (q_{1j}, ..., q_{lj}),\ j \in [1..n] \quad (1)$$

where x_j denotes the amount of resource allocated to application j (i.e. a_j). If we define R_{total} as the total available resources given, and $R_j(\vec{q}_j)$ is the resource demand for the application j, the resource constraint is:

$$\sum_{j=1}^{n} R_j(\vec{q}_j) = \sum_{j=1}^{n} R_j((q_{1j}, ..., q_{lj})) = \sum_{j=1}^{n} x_j = R_{total} \quad (2)$$

where, $\vec{q}_j$ consists of QoS parameters: e.g., frame rate, frame size, Q factor, that are specific to the application. This equation is used as a constraint in the optimization problem on resource allocation for QoS management.

3.3 Benefit Function

In multimedia transmission applications, the total system benefit (B_{total}) can be defined as a weighted sum of the benefit of each application. The term *benefit* (or *utility*) may take on the meaning of users' satisfaction or of pleasure, depending on the context. For example, the function of *CPU utilization* in a queuing system can be a benefit function to be maximized. In another case, *frame loss probability* or *total waiting time* can be benefit functions which should be

The total benefit for a single resource environment is as follows [24]:

$$B_{total} = \sum_{j=1}^{n} \alpha_j \times B_j(x_j) \quad (3)$$

The weights α_i describe the priority (or relative importance) between the n applications. $B_j(x_j)$ is the benefit function for application j, where x_j is determined by a Resource Demand Function $R_j(.)$ and a QoS factor $\vec{q}_j$. In practical real-time multimedia systems, applications may need to consider various QoS dimensions (factors) on various multiple (or single) resources.

We extend the above single resource-based benefit function to the multiple level. In the case of the multiple resource problem, the benefit function is [14]:

$$B_{total} = \sum_{j=1}^{n} \alpha_j \times B_j(\sum_{i=1}^{m} a_{ij}x_{ij}) \quad (4)$$

where x_{ij} denotes the allocated amount of resource i for application j. α_j is a positive constant to denote application j's weight between n applications. a_{ij} is a positive constant. In a multiple resource problem, the total system benefit (B_{total}) is the summation of each application's benefit, which is a function of a parameter given by the summation of all the allocated resources.

3.4 QoS Allocation Problem

Given that applications operate at their maximum levels of quality or adapt at tolerable levels of quality, the system should consider the amount of total available resources and the allotment of resources for each application based on its benefit function. Therefore,

the question of *"How should we allocate resources to the competing applications under resource constraints so that the total system benefit (utility) can be maximized?"* arises, and we would generalize this problem using our analytical resource management model which was presented in Section 3, and we would solve the optimal solutions.

From the resource constraint (Eq. 2) and the total system benefits (Eq. 3 and Eq. 4), we now define some optimization problems to maximize the system benefit, subject to the resource constraints. The solution gives the optimal values of the QoS parameters for each application. The dual of the constrained optimization problem derived by using the method of Lagrange multipliers can be used to compute the solution.

4 Algebraic Approach

As described before, the resource allocation problem in this paper is an optimization problem. Given a fixed amount of the resource, we are asked to determine its allocation to n applications so that the benefit function under the constraint could be maximized. The amount of resource allocated to each application is treated as a continuous variable, depending upon the case. This is a special case of a *nonlinear programming problem*, and therefore our approach to modeling and solving the resource allocation optimization problems draws upon many concepts from the maturely developed topics and algorithms that have been widely used in the mathematics and physics fields. We will put together the resource allocation framework that we have proposed, mathematical techniques and algorithms, and economic theories into an architecture for approaches to solving single and multiple resource allocation problems. To solve our resource allocation problems which are nonlinear programming problems, we have adopted the well-known *Kuhn-Tucker Theorem* and *Lagrangian multiplier method* [2, 19, 27]. Our previous work [14] focused on local resource management done by a *Local Resource Manager (LRM)* in each site. That is, our purpose was to optimize (maximize) the total system benefit under the given resource constraints on a single site. However, for the global optimization of a distributed multiple resource management, we should consider *global resource management* done by a *Global Resource Manager (GRM)*, which can do the *inter-site balancing of resources for computations.*

4.1 Single Resource Allocation

We can represent the single resource allocation problem as follows:

Problem 1 ***(SR):*** *We refer to the following problem as the single resource allocation problem.*

$$find \quad x_j, \quad j \in [1,n] \tag{5}$$

$$s.t.\ minimize \quad \sum_{j=1}^{n} C_j(x_j) \tag{6}$$

$$subject\ \ to \quad \sum_{j=1}^{n} x_j = R_{total}, \tag{7}$$

$$x_j \geq 0, \quad i = 1,2,...,n. \tag{8}$$

where C_j is concave and continuously differentiable over an interval, including [0,R_{total}], and R_{total} is a positive constant.

The algorithm for solving the above single resource problem is present in [15]. Relevant lemmas and proofs are also not present here for brevity.

4.2 Multiple Resource Allocation

We could generalize the single resource allocation problems to the multiple resource allocation problem, which allows more than one type of resource, and hence modeling capability is substantially enhanced by this generalization. Given the resources i of amounts $R_{total_i}, i = 1,2,...,m$, and applications $j = 1,2,...,n$, maximize the total benefit, $\sum_{j=1}^{n} C_j(\sum_{i=1}^{m} a_{ij}x_{ij})$, where x_{ij} denotes the amount of resource i allocated to application j, and C_j denotes a concave benefit function of application j.

Problem 2 ***(MR):*** *We refer to the following problem as the single resource allocation problem with continuous variables.*

$$find \quad x_{ij}, \quad i \in [1,m],\ j \in [1,n] \tag{9}$$

$$s.t.\ minimize \quad \sum_{j=1}^{n} C_j(\sum_{i=1}^{m} a_{ij}x_{ij}) \tag{10}$$

$$subject\ to \quad \sum_{j=1}^{n} x_{ij} = R_{total_i}, \quad i = 1,...,m, \tag{11}$$

$$x_{ij} \geq 0, \quad i = 1,2,...,m, \quad j = 1,2,...,n. \tag{12}$$

Where the C_j are concave and continuously differentiable over an interval including [0,R_{total_i}], and R_{total_i} are positive constants. The a_{ij} are nonnegative.

We adopted several non-linear programming theorems and algorithms from [20, 9, 7, 27] to solve the above problems. For further details, please refer to [15].

5 Welfare Economics Approach

To achieve a distributed resource allocation problem by formulating a computational economy and finding its competitive equilibrium, we adopt welfare economic theories and an auction/bidding algorithm. In this section, we will consider an example of a continuous media server system and will apply the welfare economic theories to it. Through formulating a pricing-based economy (market), we can achieve the Pareto optimal allocations under many circumstances with little or no central guidance.

5.1 Economic Optimization

Auctions are a market institution with an explicit set of rules determining resource allocations and prices on the basis of bids from participating market agents. Hence, the free-market-based auction methods solve the problem, and the Lagrange multiplier is the price of the resource fixed by the auction. We have adapted a sealed-bid increasing auction [3], where each agent (application) presents the fair bid for the available resource, based on the current price of the resource and the application benefit function. The auction makes trade-offs between the values generated by the agents. The basic protocol is that agents send bids, and the auction determines an allocation [24]. Table 2 and

Algorithm 1 *Economic_Optimization (avail_resource)*

▷ θ_i : *scale factor for price increase on resource i, where i ∈ [1,m]*
Initialize $price_i$;
Initialize $done_i = 0$, where i ∈ [1, m];
while (! $\bigcap_i done_i$)
  for each resource i whose $done_i == 0$, i ∈ [1, m]
    for each agent (or application) a_j, where j ∈ [1, n]
      ▷ *bidding*
      if ($\alpha_j * B_j(.) > p_i * \Delta x_i$)
        ▷ *(priority * benefit) is greater than cost (=price * Δ of resource)*
      then
        ▷ *buy* Δx_i
        $bid_i \leftarrow R_i + \Delta x_i$;
      else
        ▷ *sell* Δx_i
        $bid_i \leftarrow R_i$ - Δx_i;
    end for
  end for
  for each resource i, where i ∈ [1, m]
    if ($bid_i > avail\_resource_i$)
      then $price_i \leftarrow price_i + \theta_i$;
      else $done_i \leftarrow 1$;
  end for
end while

Table 2: Algorithm for *Economic_Optimization*

We run a bidding process for each application a_j on each resource i. If the value of benefit of i times the priority of i is greater than the cost (i.e., the current price of resource i times Δ *of resource*), the application would buy Δ amount of resource i. Otherwise, the application sells at the same amount. This process is repeated for all resources. For each resource i, if the total amount of each application's bid is less than current available resource, the flag is set on *done*. If it is greater, the auctioneer increases the current price of resource by some amount and restarts the bidding.

5.2 Problem Statements

In a market based economy, decisions are based on prices, and communications are done via exchanges of bids and prices between agents. In many cases, this decentralizing mechanism minimizes the communication overheads and converges in reasonable time. We use the *price* of resources fixed by an auction as the Lagrange multiplier in the constraint resource optimization problem:

$$d\,Benefit - L * d\,Constraint = 0 \qquad (13)$$

where d, $Benefit$, L, and $Constraint$ denote derivation, the utilization (object) function to be optimized, the Lagrange multiplier, and constraint functions, respectively. If the unit of the benefit function is money ($), we could think of the Lagrange multiplier as the *price* of a resource from Eq. 13.

Problem 3 ***(Economic SR):*** *We refer to the following problem as the single resource allocation problem.*

$$find \quad x_j, \quad j \in [1,n] \qquad (14)$$

$$s.t.\ maximize \quad \sum_j^n \alpha_j * B_j(x_j) \qquad (15)$$

$$subject\ to\ : \quad \sum_j^n p * x_j \leq E_{total} \qquad (16)$$

where E_{total} is the total budget (i.e., available resource times the price of the resource).

Problem 4 ***(Economic MR):*** *We refer to the following problem as the multiple resource allocation problem.*

$$find \quad x_{ij}, \quad i \in [1,m],\ j \in [1,n] \qquad (17)$$

$$s.t.\ maximize \quad \sum_j^n \alpha_j * B_j(\sum_i^m a_{ij} x_{ij}) \qquad (18)$$

$$subject\ to\ :\quad \sum_{j}^{n} p_i * x_{ij} \leq E_{total,i}, \quad for\ \forall i \in [0,m] \tag{19}$$

$E_{total,i}$ is the budget for resource i, which is the available resource i times the price of i.

5.3 Example

Consider the continuous media server system shown in Figure 1.

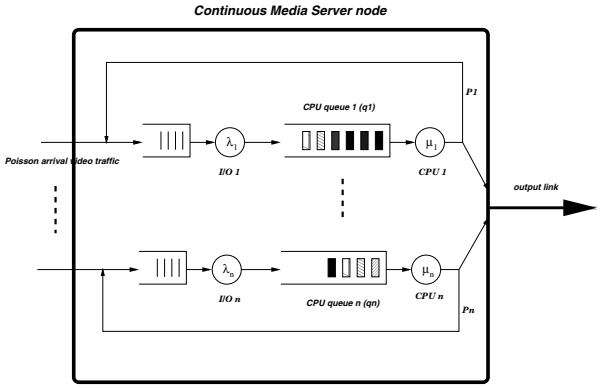

Figure 1: The Finite State Space Queuing Model of a Continuous Media Server

The system consists of multiple CPUs and their CPU queues (buffers). We assume the service needs (or video traffics) for application i as a stochastic process with a specific arrival rate λ_i. That is, a_i arrives in a finite size of CPU queue (q_i) with a Poisson distribution and is executed by the CPU i at a certain service rate (e.g. μ_i). Each I/O module also has its own I/O buffer. We assume that the average CPU execution time per burst is $1/\mu_j$, which are independent exponentially distributed random variables, and that successive I/O burst times are also independent exponentially distributed with a mean of $1/\lambda_j$. At the end of a CPU burst a video frame requests an I/O operation with the probability $0 \leq P_j \leq 1$; otherwise, it completes execution. At the end of a video frame completion, another statistically identical video frame enters the server system. Each video traffic in this scenario can be modeled as a *finite state space* birth-death queuing model (so called *M/M/1/n* queue model). Here, the birth rate is λ_j, and the death rate is $\mu_j P_j$. Let the number of programs in the CPU queue including any being served at the CPU denote the state of the system i, where $0 \leq i \leq n$. We see that the steady-state probabilities are given by:

$$p_i = \rho^i p_0. \tag{20}$$

$$p_0 = \frac{1}{\sum_{i=0}^{n} \rho^i}. \tag{21}$$

where ρ denotes $\lambda/(\mu P)$, which is the utilization of the individual server.

We could get the *CPU utilization*, which is given by:

$$U_1 = \begin{cases} \frac{\rho - \rho^{n+1}}{1-\rho^{n+1}} & \rho \neq 1 \\ \frac{n}{n+1} & \rho = 1 \end{cases}$$

Let U_2 denote the frame loss probability. Then we also get the following equations for U_2 :

$$U_2 = \begin{cases} \frac{1-\rho}{1-\rho^{n+1}} & \rho \neq 1 \\ \frac{1}{n+1} & \rho = 1 \end{cases}$$

In this example, we assume that there are two applications (just to draw the Edgeworth box. This can be extended to any number of applications, as explained). The two utility functions (the CPU utilization and frame loss probability) can be considered as benefit functions. Two variables, *queue length (q)* and *CPU service rate (μ)*, can be regarded as resources, since q denotes the required buffer size or buffer capacity, and μ denotes the link capacity. Hence, we could apply a benefit function and resource variables into the welfare economic theories of Pareto optimal points, contract curves and the Edgeworth box. We may get the result as shown in Figure 2.

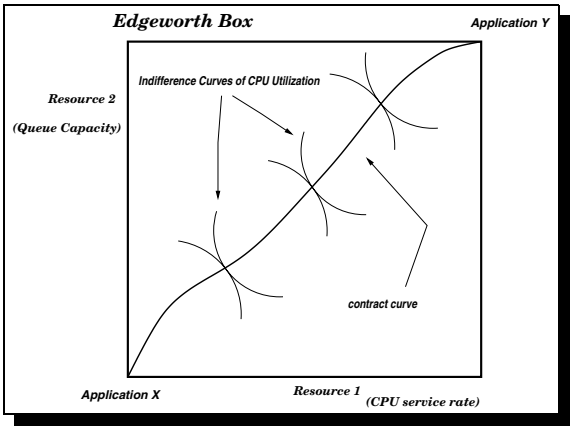

Figure 2: Pareto Optimal Points on Contract Curve in the Edgeworth Box

We can also apply our economic framework in distributed network environments such as a wide area cell network built up of high-speed data links interconnected by high-speed cell switches, such as ATM switches.

6 Conclusions & Future Work

In this paper, we have proposed a market-based resource management and QoS provisioning model that exploits the various properties of distributed multimedia applications. Our model can represent the resource and QoS requirements of applications in multimedia system environments using end-to-end QoS based metrics (i.e., application benefit functions and resource demand functions), which are defined over single or multiple resources. Our future work includes incorporating the current model and algorithms into practical systems.

References

[1] T. F. Abdelzaher, E. M. Atkins, and K. G. Shin. QoS Negotiation in Real-Time Systems and Its Application to Automated Flight Control. In *Proceedings of IEEE Real-Time Technology and Applications Symposium*, Montreal, Canada, 1997.

[2] M. Avriel, editor. *Nonlinear Programming: Analysis and Methods*. Englewood Cliffs, Prentice-Hall, New Jersey, 1976.

[3] S. H. Clearwater. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Publishing Co., Singapore, 1996.

[4] J. DeMeer. On the Specification of End-to-End QoS Control. In *Proceedings of 5th International Workshop on Quality of Service*, May 1997.

[5] J. Doyle. A Reasoning Economy for Planning and Replanning. In *Proceedings of the ARPA Planning Initiative Workshop*, 1994.

[6] H. T. E. Jensen, C. Locke. A Time-Driven Scheduling Model for Real-Time Operating Systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, 1985.

[7] J. Einbu. A Finite Method for the Solution of a Multi-Resource Allocation Problem with concave Return Functions. *Mathematics of Operations Research*, 9:232–243, 1984.

[8] M. S. et. al. An Economic Paradigm for Query Processing and Data Migration in Mariposa. In *Proceedings of Parallel and Distributed Information Systems*, 1994.

[9] S. G. H. Luss. Allocation of Effort Resources Among Competitive Activities. *Operations Research*, 23:360–366, 1975.

[10] K. Kawachiya, M. Ogata, N. Nishio, and H. Tokuda. Evaluation of QoS-Control Servers on Real-Time Mach. In *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio Video*, April 1995.

[11] J. Kurose and R. Simha. A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.

[12] T. Lawrence. The Quality of Service Model and High Assurance. In *Proc. 2nd IEEE High-Assurance System Engineering Workshop*, Aug. 1997.

[13] W. Lee and B. Sabata. Admission Control and QoS Negotiations for Soft-Real Time Applications. In *Proceedings of the 6th IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, June 1999.

[14] W. Lee and J. Srivastava. An Analytical QoS-based Resource Management Model for Multimedia Applications. In *Proceedings of the 3rd IASTED/ISMM Conference on Internet and Multimedia Systems and Applications (IMSA99)*, Nassau, Grand Bahamas, Oct 1999.

[15] W. Lee and J. Srivastava. An Algebraic QoS-based Resource Management Model for Competitive Multimedia Applications. *To appear in Journal of Multimedia Tools and Applications, Kluwer Academic Publishers*, 2000.

[16] J. M. Levy, editor. *Essential Microeconomics for Public Policy Analysis*. POraeger, 1995.

[17] B. Li and K. Nahrstedt. A Control Theoretical Model for Quality of Service Adaptations. In *Proceedings of 6th International Workshop on Quality of Service*, Napa, CA, May 1998.

[18] A. F. M. Jones, J. Barbera III. An Overview of the Rialto Real-Time Architecture. In *Proceedings of the seventh ACM SIGOPS European Workshop*, Sept 1996.

[19] O. Mangasarian, editor. *Nonlinear Programming*. McGraw-Hill, New York, 1969.

[20] K. Mjelde, editor. *Methods of the Allocation of Limited Resources*. Chichester: John Wiley & Sons, 1983.

[21] K. D. M.S. Miller, editor. *Markets and Computation: Agoric Open Systems*. B.A. Huberman, editor, The Ecology of Computation, North-Holland, Amsterdam, 1988.

[22] J. Nieh and M. Lam. The Design, Implementation and Evaluation of SMART: A Scheduler for Multimedia Applications. In *Proceedings of the 16th ACM Symposium of Operating Systems Principles*, Oct 1997.

[23] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. A Resource Allocation Model for QoS Management. In *Proceedings of the IEEE Real-Time Systems Symposium*, 1997.

[24] B. Sabata, S. Chatterjee, and J. Sydir. Dynamic adaptation of video for transmission under resource constraints. In *Proc. of 17th IEEE International Conference of Image Processing*, Chicago, IL, Oct. 1998.

[25] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 1996.

[26] A. Smith, editor. *An Inquiry into the Nature and Causes of the Wealth of Nations*. Volume 39 in Robert Maynard, Great Books of the Western World. Encyclopaedia Britannica, Inc., London, 1954.

[27] N. K. Toshihide Ibaraki, editor. *Resource Allocation Problems: Algorithmic Approaches*. The MIT Press, 1988.

[28] M. W. W.E. Walsh. A Market Protocol for Decentralized Task Allocation. In *Proceedings of the third International Conference on Multiagent Systems*, July 1998.