

# Economic Models for Allocating Resources in Computer Systems

*Donald F. Ferguson* \*      *Christos Nikolaou* †      *Jakka Sairamesh* ‡  
*Yechiam Yemini* §

## 1 Introduction

With the advances in computer and networking technology, thousands of heterogeneous computers can be interconnected to provide a large collection of computing and communication resources. These systems are used by a growing and increasingly heterogeneous set of users. A macroscopic view of distributed computer systems reveals the complexity of the organization and management of the resources and services they provide. This complexity arises from size (e.g. no. of systems, no. of users) and heterogeneity in applications (e.g. on-line transaction processing, multimedia, intelligent information search) and resources (CPU, memory, bandwidth, locks, naming services).

The complexity of resource allocation is further increased by several factors. First, in many distributed systems, the resources are in fact owned by multiple organizations. Second, the satisfaction of users and the performance of applications is determined by the simultaneous allocation of multiple resources. A multimedia server application requires I/O bandwidth to retrieve content, CPU time to execute server logic and communication protocols, and networking bandwidth to deliver the content to clients. The performance of applications may also be altered by trading one resource for another. For example, the multimedia server application may perform better by releasing memory and acquiring higher CPU priority. This trade may result in smaller buffers for I/O and networking but improve the performance

---

\*Donald F. Ferguson is with IBM T. J. Watson Research Center, Hawthorne, New York, 10532. E-mail: dfferg@watson.ibm.com

†Christos Nikolaou is with Institute for Computer Science-FORTH, Science and Technology Park, P.O. Box 1385, GR 711 10, Heraklion, Crete, Greece, E-Mail: nikolau@ics.forth.gr

‡J. Sairamesh is with the Center For Telecomm. Research, Columbia University, New York, 10027. E-mail: jakka@cs.columbia.edu, jakka@ctr.columbia.edu

§Yechiam Yemini is with the dept. of Computer Science, Columbia University, New York, 10027. E-mail: yemini@cs.columbia.edu

of the communication protocol execution. Finally, in a large distributed system, the set of systems, users and applications is continuously changing.

This massive complexity makes traditional approaches to resource allocation impractical in modern distributed systems. Traditional approaches attempt to optimize some system-wide measure of performance (e.g. average response time, throughput). The optimization is performed either by a centralized algorithm with complete information, or by a decentralized, consensus based algorithm. The current and future complexity of resource allocation problems described above makes it impossible to define an acceptable system-wide performance metric. What single, system-wide performance metric adequately reflects the performance objectives of a multimedia server and an online transaction processing system? Centralized or consensus based algorithms are impractical in a dynamic system owned by multiple organizations.

Resource allocation complexity due to decentralization and heterogeneity is also present in human economies. In general, modern economies allocate resources in systems whose complexity overwhelms any algorithm or technique developed for computer systems. In this paper, we discuss the similarities between complex distributed systems and human economies. We demonstrate how competitive economic models provide algorithms and tools for allocating resources in distributed computer systems. Several different computer system resource allocation problems are presented and economic based algorithms for allocating resources are evaluated.

## 1.1 *Motivation and goals*

In designing resource allocation and control mechanisms in large distributed systems and networks several goals need to be considered. Some of the important goals are outlined below. In this section, we briefly present many of these goals. In later sections, we present case studies in which economic models have been successfully applied to resource control problems in distributed systems, and discuss how the goals are achieved.

- Partition large complex allocation problems into smaller, disjoint allocation problems.
- Decentralize resource access, allocation and control mechanisms.
- Design reliable, fault-tolerant and robust allocation mechanisms.
- Design scalable architectures for resource access in a complex system.
- Provide guarantees to users and applications on performance criteria. Some of the performance criteria in distributed systems include the following:

- Average response time
  - Throughput
  - Application failure probability
  - Information loss; packet loss in communication networks
  - Maximum response time (message delay)
- Define system performance criteria that reflect in aggregate the diverse individual criteria of users and applications.
  - Design a unified framework in which users have transparent access to the services of a distributed system, and services are provided in an efficient manner. This framework should hide the complexity of multiple resource suppliers and resource allocation policies.

## 1.2 *Why use economic models?*

Economic models provide several interesting contributions to resource sharing algorithms. The first is a set of tools for limiting the complexity by decentralizing the control of resources. The second is a set of mathematical models that can yield several new insights into resource sharing problems.

In an economy, decentralization is provided by the fact that economic models consist of agents which selfishly attempt to achieve their goals. There are two types of agents, *suppliers* and *consumers*. A consumer attempts to optimize its individual performance criteria by obtaining the resources it requires, and is not concerned with system-wide performance. A supplier allocates its individual resources to consumers. A supplier's sole goal is to optimize its individual satisfaction (profit) derived from its choice of resource allocations to consumers.

Most economic models introduce money and pricing as the technique for coordinating the selfish behavior of agents. Each consumer is endowed with money that it uses to purchase required resources. Each producer owns a set of resources, and charges consumers for the use of its resources. The price a producer charges for a resource is determined by its supply and the demand of the agents for the resource. The price system ensures that a realizable allocation of resources is achieved. The performance criteria of the system as a whole is determined by some combination of the performance criteria of the individual agents. Several papers apply versions of this model to decentralized resource allocation in computer systems [16, 12, 7, 8, 9, 10, 28, 17, 23, 25].

In an economic model of a computer system, the consumers are applications such as transactions, computational jobs, file transfer applications, multimedia teleconferences, and news distribution. Computer systems are the primary suppliers and control local resources such as CPU time, memory, cache, disks, I/O bandwidth and communication resources. Suppliers also include servers which provide specific *logical* services such as network file servers, yellow pages servers, name servers, news servers and recently *Web* servers [24]. Suppliers control access to their resources via prices, and consumers buy resources from the suppliers to satisfy their computing needs (or service requirements). Prices are adjusted by the suppliers based on the demand placed by the consumers.

The main goals of this paper are to describe the economic models and their use in controlling access to resources in large and complex distributed computer systems. In section 2, we discuss the economic models and problems to solve in a computer system modeled as an economy. We provide an overview of economic concepts that are used in the case studies and discussions. In section 3, we present some of the applications of economic models (specially micro-economic and game models) to several resources allocation problems in computer networks and distributed systems. We describe a few of the case studies in detail to highlight some of the economic concepts and their applicability in controlling large distributed systems. In section 4, we conclude by presenting some of the technical challenges in applying economic models to distributed computer systems and discuss directions for future work.

## 2 Economic Models

### 2.1 Resources and Allocations

An economy contains a set of resources  $r_1, r_2, \dots, r_M$ . An *allocation* of resources to an agent is a vector  $x = \{x_1, x_2, \dots, x_M\}$  in which  $x_i$  represents the quantity of resource  $r_i$  allocated to the agent. In a simple computer system, the resources might be CPU and memory. An allocation could be 10% of the CPU and 20% of real memory.

An agent *prefers* some allocations over others. Agent preferences are typically represented by *utility* functions. These functions are mappings from a resource vector space to a real number. Let  $U(.)$  be a utility function which maps the resource space to a real number and let  $x$  and  $y$  be two resource allocation vectors. An agent prefers  $x \succ y$ , when  $U(x) > U(y)$ . Debreu [2] has shown that if the preference ordering is complete, reflexive, transitive, and continuous, then a continuous utility function can represent this preference. In a simple computer system, each allocation of CPU and memory might define a response time for an application. An agent could prefer allocations that provide shorter response times, and the

utility function would be  $-1.0$  times the response time.

A formal way to look at preferences is as follows: A user has a preference on an allocation, such that  $x \succeq y$ , which implies that vector  $x$  is preferred to vector  $y$ , where  $x, y \in X$  ( $X$  is the resource consumption set, and  $X \in \mathfrak{R}^M$ ). A user has no preferences or is indifferent to the allocations  $x$  and  $y$  if  $x \sim y$ . The preferences are transitive, if  $x \succeq y$  and  $y \succeq z$ , implies  $x \succeq z$  (where  $z \in X$ ).

The set of allocations that are *feasible* under the budget constraint is given by the following for each agent (where  $w$  is the wealth of an agent and  $\vec{p}$  is the price vector corresponding to the resources. That is, the price per unit of  $r_i$  is  $p_i$ ).

$$B(\vec{p}) = \{x : \vec{p} \bullet x \leq w\} \quad (1)$$

$B(\vec{p})$  is called the *feasible set* or *budget set*.

The *demand set* for an agent is the set of allocations that lie in the feasible set and provide maximal benefit or preference.

$$\Phi(\vec{p}) = \{x : x \in B(\vec{p}), U(x) \geq U(x'), \forall x' \in B(\vec{p})\} \quad (2)$$

Stated simply, the agent demands the most preferred, affordable allocation of resources.

## 2.2 Allocation and Pricing Models

In economic models, there are two main ways to allocate resources among the competing agents. One of them is the *exchange based economy* and the other is the *price based economy* [1, 2, 3, 15]. In the exchange based economy, each agent is initially endowed with some amounts of the resources. They exchange resources till the marginal rate of substitution of the resources is the same for all the agents. The agents trade resources in the direction of increasing utility (or maximal preference). That is, two agents will agree on an exchange of resources (e.g. CPU for memory) which results in an improved utility for both agents. The *pareto optimal* allocation is achieved when no further, mutually beneficial resource exchanges can occur.

In a price based system, the resources are priced based on the demand, supply, and the wealth in the economic system. The allocations are done based on the following mechanism. Each agent is endowed with some wealth. Each agent computes the demand from the utility function and the budget constraint. The aggregate demand from all the agents is sent to the suppliers who then compute the new resource prices. If the demand for a resource is greater than its supply, the supplier raises the price of the resource. If there is surplus supply, the price is decreased. The agents again compute their demands given the current prices and

present the demand to the suppliers. This process continues iteratively till the equilibrium price is achieved where demand equals the supply.

Bidding and auctioning resources is another form of resource allocation based on prices. There are several auctioning mechanisms such as the *Sealed Bid Auction*, *Dutch Auction*, and *English Auction*. The basic philosophy behind auctions and bidding is that the highest bidder always gets the resources, and the current price for a resource is determined by the bid prices.

### **2.3** *What are the economic based problems?*

In this subsection, some of the interesting problems encountered when designing an economic based computer system are discussed and stated.

- How do agents demand resources? This a fundamental question regarding the agents preferences on the resources they consume. Are there smooth utility functions that can capture the agents preferences of the resources? Are there utility functions that can capture the diversity of the agents preferences?
- How are the prices adjusted to clear the economy or to clear the markets? In an economic model, efficient allocation of resources occurs when the demand equals the supply (for a good) at a certain equilibrium price vector.
- What rational pricing mechanisms do the suppliers adopt? This question raises issues on pricing mechanisms that will attract agents (consumers).
- How do suppliers provide price guarantees to agents? This is a fundamental question in advertising and providing price guarantees to agents. Delays in information about prices, and surges in demand can cause prices to vary. Therefore agents can make bad decisions.
- What are the protocols by which the consumers and suppliers communicate to reserve resources ?
- What are the general allocation principles? Can economic models give insight into the allocation mechanisms that can cause the computer system to reach equilibrium? Can these principles be used practically to evolve the computer system in a way that price equilibrium can be achieved?

- What strategies do the agents take to choose the right suppliers? What are the mechanisms or incentives by which migration of several agents between suppliers can be limited?

The solutions to some of these problems are presented in the case studies in the next section.

### 3 Applications of Economic Models

Most studies of resource allocation mechanisms have used a performance model of the resource, where the very concept of the resource is defined in terms of measurable qualities of the service such as utilization, throughput<sup>1</sup> and delays. Optimization of resource allocation is defined in terms of these measurable qualities. One novelty introduced by the economic approach is the use of multi-objective (utilities) optimization techniques to characterize and compute optimum allocations. Economic modeling of computer and communication resource sharing uses a uniform paradigm described by two level modeling:

Resource sharing → performance model → economic optimization model.

In the first step one transforms a resource sharing model to a performance model (example: queueing service model). This model establishes quantifiable parametrization of resource allocation. These parameters are then used to establish an economic optimization model. In particular, the question of whether the resource is a piece of hardware, a network link, a software resource such as a database or a NFS server, or a virtual network entity such as a TCP connection does not matter. The first modeling transformation eliminates the details of the resource and captures its relevant behaviors and optimization parameters.

This paper discusses two types of economic models, and their applications to modeling computer systems. One of them is a non-pricing based approach which relies on game-theoretic [4] models to design decentralized control mechanisms for allocating resources [5, 6, 18, 19, 20, 22, 21, 11, 29]. The other is a *pricing* based approach where consumers and suppliers interact via *market* mechanisms which is based on resource prices and money. Case studies involving pricing models and market based mechanisms are discussed in detail with several technical and numerical results to support the effectiveness of economic models in resource optimization and allocation in computer systems.

---

<sup>1</sup>Transmission rate: number of units transmitted per unit of time. In this paper, throughput and thruptut are the same.

### 3.1 *Non-Pricing Based Approaches*

#### 3.1.1 *Selfishness in Resource Allocation*

Early work in applying economic models to resource allocation problems in large computer networks was considered by Yemini [5, 6], Jaffe [11], Lazar [18, 19], and Mazumdar [21]. Their papers proposed several decentralized control mechanisms based on game theoretic models to optimally control the transmission rates of packets in a packet based network. Each user of the packet network performed selfish optimization (independent of the other users) of a performance index (called the power) which is a function of the throughput and average delay.

In [6], two specific examples of decentralized control of resources in complex computer communication systems are discussed, where centralized control is neither applicable nor practical. The first example is on general principles for optimal broadcast channel access schemes, where it was shown using selfish optimization techniques that the pareto optimal transmission rates for  $N$  competing stations could be obtained by the following principle: *the silence and thruput of each competing station must be equal*. The second example is on decentralized flow control in fixed virtual circuit (VC) networks using *power maximization* as the performance index. Two VCs maximized a utility function which is a function of throughput and delay. The utility function is given below with  $x$  as the throughput of VC 1 and  $y$  as the throughput of VC 2.

$$U_1(x, y) = \frac{x}{(\mu - x - y)}, \quad U_2(x, y) = \frac{y}{(\mu - x - y)} \quad (3)$$

Using multiobjective optimization techniques, the set of pareto optimal thruput pairs computed for these VCs is the line  $x + y = \mu/2$ . The main result is that pareto optimal resource allocation methods (which involves selfish users) can result in optimal decentralized resource allocation algorithms.

#### 3.1.2 *Cooperation in Resource Allocation*

Similar to the decentralized control of resources in computer networks, allocation of files (or data objects) in a complex distributed computer system was considered by Kurose *et. al.* [13]. Their paper provided decentralized algorithms to allocate resources (such as files or file fragments) in a cooperative and non-competitive manner among agents (computer systems). The optimization criteria included communication cost and average processing delay. Unlike the selfish model where each user (VC) had its own utility function, this model has a global

utility function which is known to all the users ( $N$  nodes) in the distributed system. The utility function for accessing a single file resource is given below (where  $K$  is a constant that defines the relative cost of communication versus computation):

$$U(x) = \sum_{i=1}^{i=N} (C_i + \frac{K}{\mu - \lambda x_i}) x_i \quad (4)$$

In the above equation,  $C_i$  is the communication cost of accessing the file object at node  $i$ ,  $\lambda$  is the system wide arrival rate,  $x_i$  is the probability of an access being directed at the fraction of the file resources stored at  $P_i$  (node  $i$ ). Each node is given an initial allocation ( $x_i^0$ ). They then compute the optimal allocation by solving the global optimization problem in a distributed fashion. This done by trading appropriate amounts of resources iteratively (marginal rate of substitution of resources) among themselves till they reach a point where the marginal rate of substitution of resources is equal. The allocation at this point is pareto optimal.

A fundamental problem in distributed control mechanisms which attempt to allocate resources in an efficient manner defined by pareto efficiency [1, 2, 3, 15] or Nash equilibrium [4] is that the distributed algorithms must converge to the equilibrium point. In the works by [13, 12, 18, 19, 20, 5, 6, 22, 21, 29] several experiments based on simulations, and theoretical work were performed to verify the convergence properties of the algorithms.

## 3.2 Pricing Based Approaches

A different economic approach based on pricing rather than non-pricing schemes is considered by Ferguson *et. al.* [7, 8, 9, 10], Kurose *et. al.* [12], Mark *et. al.* [16], Stonebraker *et. al.* [23], Carl *et. al.* [28], Ron *et. al.* [25], and Sairamesh *et. al.* [17]. Pricing schemes provide a decentralized way for consumers (users) to buy resources from the computer systems (such as networks and distributed systems). The interaction between the consumers and suppliers is through market mechanisms such as prices. In this paper, we present this approach to resource allocation in computer systems using the case studies of load balancing and data management in distributed systems, flow control and Quality of Service (QoS) provisioning in Integrated Services Networks, and Multiple Access Protocols in Broadcast Packet Networks.

### 3.2.1 Load Balancing Economy

The load balancing economy was considered in [7, 10]. In this economy, there are  $N$  processors ( $P_1, P_2, P_3, P_4, \dots, P_N$ ) connected via a point-to-point network. A link  $e_{ij}$  is a connection between  $P_i$  and  $P_j$ . Each link has a delay variable which is  $d_{ij}$ . The service rate of each

processor  $P_i$  is  $r_i$  (service rate is with respect to  $P_1$  where  $r_1 = 1$ ). The resources are processor time (CPU time) and the communication bandwidth. Jobs enter the distributed system and request resources based on the prices. Job  $j$  has a service time of  $\mu_j$ , and it purchases  $\mu_j/r_k$  time units on  $P_k$ . Jobs have various preferences on the service they wish to receive. The preferences are the following:

- Price Preference (PP): Jobs prefer service to be done as cheaply as possible. The cost  $C_k$  is composed of the cost of accessing service at processor  $P_k$  and cost of to and fro communication bandwidth from the job origin to  $P_k$  (here  $k \in [1, N]$ ).

$$\min_{\forall k} [C_k] \tag{5}$$

- Service time preference (ST): Jobs prefer the element of the budget set which gives the least response time (or service time). Job  $j$  located on  $P_i$  computes the service time at processor  $P_k$  from the following:

$$\min_{\forall k} [ST_k = \frac{\mu_j}{r_k} + ReqBC_j * d_{ik} + RspBC_j * d_{ki}] \tag{6}$$

The number of bytes needed to send a request message for a job to migrate is  $ReqBC$ , and the number of bytes required to describe the result of the job is  $RspBC$ .

- Service and price preference (SVTP): Jobs place a relative preference of service time over cost at a processor. This is given by the equation below ( $A$  is a weight giving relative importance to ST).

$$\min_{\forall k} [C_k + A * ST_k] \tag{7}$$

The pricing mechanism in the economy is based on auctions held by the processors and bidding by the jobs (consumers). There are four types of auctions considered by this economy. The first is an *English auction* where the price of the resource is gradually increased with the bidding. The highest bidder obtains the resource. The second is the *Dutch auction*, where prices are gradually lowered by the seller till one of the buyers claims the resource. The third is the *Hybrid auction*, where the asking price of a resource is increased if a bid is submitted, and decreased if no bid is submitted. The Hybrid auction attempts to find the highest price using this process. The fourth is the *Sealed Bid auction*, where sealed bids are submitted by the agents and the highest bid is given access to the resource. In this auction model, agents are not aware of the amounts bid by the other consumers.

## Performance Comparison

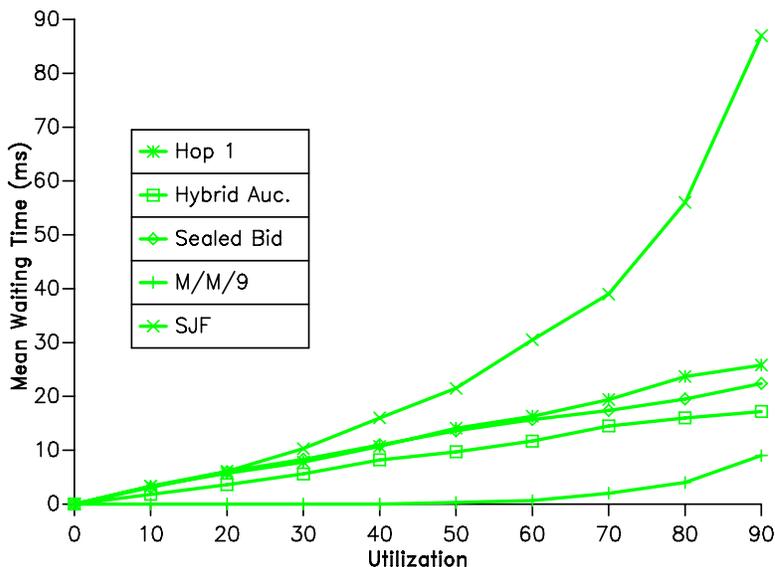


Figure 1: Economic Versus Non-economic Algorithms: This graph plots the mean job waiting times of the HOP 1 algorithm (non-economic alg.), the Sealed Bid and Hybrid Auction economies.

Jobs (when they arrive) perform three operations to purchase resources which are: compute the budget set, find the most preferred elements (demand set) of the budget set, and generate a bid for the demand set. The processors auction resources (link and CPU time) to the jobs, advertise resource prices in local bulletin boards and neighboring processors, and update prices based on auctioning results and arriving price updates from neighboring processors.

The load balancing problem is to design algorithms that minimize the mean job waiting time by migrating the jobs to balance the workloads of the processors nodes. Each job independently computes the best place (node) to be served based on its preferences and wealth, and the resources prices. The main goal of each processor (node) in the economy is to maximize revenue.

The load balancing economy was studied using simulation tools on a sample network of 9 processors connected in a 3x3 mesh. Figure 1 plots the mean job waiting times as a function of the system utilization. The graph shows that Hybrid and Sealed-Bid auctioning based economies performed better than the HOP 1 [26, 27] algorithm which is a non-economic algorithm, and the Hybrid-auction economy performed the best at all utilizations. At low utilizations the Sealed-Bid economy is as good as HOP 1, and better at high utilizations. The SJF (shortest job first on an  $M/M/1$  system [14]) algorithm performed the worst. The

$M/M/9$  case is depicted for comparison purposes only. The  $M/M/9$  is a queueing system [14] where 9 processors serve one queue, and there is no communication delay, and all the information is globally available and exact.

An interesting phenomena in the load balancing economy is that jobs migrate in search of suppliers based on the job preference model. Numerical studies on this phenomena indicated that jobs with response time preference migrated less compared to the jobs that have price preference. Jobs which had a combination of service time and price preferences migrated based on the weights given to service time or price.

Efficient allocation of CPU time and link bandwidth among several competing jobs (or tasks) was accomplished in a decentralized manner by limiting the complexity of resource control. Effective load balancing is done using the bidding and auctioning mechanisms, and price controls. This case study shows that for the load balancing problem, competitive economic concepts can achieve better levels of performance when compared to non-economic algorithms.

### 3.2.2 *Flow Control Economy*

The problem of allocating network bandwidth among several virtual circuits (VCs) in a packet based computer network was considered by Ferguson *et. al.* [7, 9]. They used preference relations to model the heterogeneous requirements of the VCs. The relations modeled the trade-offs in average packet delay and over all throughput of each VC. They presented decentralized flow control algorithms to allocate link capacity to VCs with heterogeneous service requirements.

The flow control economy consists of several VCs competing for link bandwidth resources in a network  $G(V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of links. Each VC  $a$  competes for bandwidth on a path  $P_a$  which consists of several links ( $M$ ) of the network. The allocation vector for a VC is  $x = \{x_1, x_2, x_3, \dots, x_M\}$ . A price system associated with the path is a vector  $\vec{p} = \{p_1, p_2, p_3, \dots, p_M\}$ . Each VC has a binary preference relation over allocations. These preferences capture each VC's thrupt and delay goals.

In the economy, each VC that arrives wants to send data from its source to its destination at a rate  $\gamma_a$ . With the current link prices along a path, VC  $a$  computes to see if it can afford the minimum thrupt goal of  $\gamma_a$ . If it can, then it uses the remaining wealth to minimize (selfish optimization) the worst case average delay of its allocation. The average delay function on each link is the average delay introduced by an  $M/M/1$  type queueing model [14]. Based on this link delay function, the utility function of VC  $a$  is given by,

$$U(x) = \sum_{i \in P_a} \frac{1}{C_i - S_i + x_i - \gamma_a} \quad (8)$$

$S_i$  is the supply on link  $i$  and  $C_i$  is the capacity of the link  $i$ . The total demand for a resource  $i$  at prices  $\vec{p}$  is the sum of the individual demands (from the VCs), or

$$d_i(\vec{p}) = \sum_{a \in A} \phi_i^a(\vec{p}). \quad (9)$$

where  $\phi_i^a(\vec{p})$  is VC  $a$ 's demand for link  $i$  at prices  $\vec{p}$ , and  $A$  is the set of VCs in the network. The excess demand function is given by  $Z(\vec{p})$ , which is the demand minus the supply at the current price vector  $\vec{p}$ . The excess demand's  $i$ th component is simply given by  $Z_i(\vec{p}) = d_i(\vec{p}) - S_i$ . The flow control economy attempts to compute the competitive network equilibrium by balancing the supply and demand.

The flow control economy is in equilibrium at prices  $\vec{p}$  if for all  $i \in E$  either: 1)  $Z_i(\vec{p}) = 0$ , or 2)  $Z_i(\vec{p}) \leq 0$  and  $p_i = \epsilon$ . This means that a minimum charge for a VC to access a link is  $\epsilon > 0$  (in spite of excess supply).

In the economy, prices are updated using a *tatonement* process [1, 15]. The process changes the prices of resources if there is an imbalance in the supply and demand. The rule for updating  $p_i$  is the following:

$$p_i = \text{Max}[p_i + p_i \frac{Z_i(\vec{p})}{S_i}, \epsilon], \quad \epsilon > 0 \quad (10)$$

*Pricing Algorithm:* The tatonement process is as follows:

- 1 Choose an initial price  $\vec{p}$ .
- 2 For all  $a \in A$ , compute  $\vec{\phi}_a(\vec{p})$ .
- 3 If  $\forall i \in E, (Z_i(\vec{p}) = 0)$  or  $(Z_i(\vec{p}) \leq 0 \text{ and } p_i = \epsilon)$ . Equilibrium has been reached and the iteration stops.
- 4 Otherwise, for all  $i \in E$ : update prices  $p_i$  (formula above).
- 5 Go to step 2.

Each link (supplier) performs the price update algorithm independently of other links (in a decentralized manner). Similarly each buyer (VC) computes the demand based on its preferences on the resources (independent of other VCs).

### Demand Convergence Link Inclusion Prob.

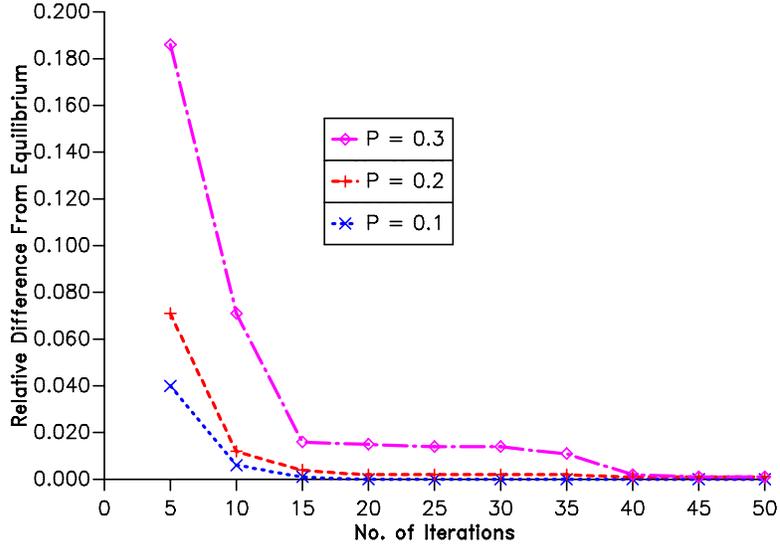


Figure 2: Convergence to Equilibrium: Flow Control Economy. This figure plots the economy’s convergence to equilibrium. The three curves represent random economies in which the probability of link  $i$  being in VC’s path is 0.1, 0.2, and 0.3. These yield mean VC lengths of 2.5, 5.0, and 7.5

The main results in the flow control economy are given by the following theorems. The first theorem states that at the equilibrium price vector, the flow control economy allocated resources in a pareto-optimal way, and then the allocation was unique given the wealth of each VC. The second theorem states that there is at least one equilibrium price vector. There are several examples given by Ferguson [7, 9] where many equilibrium price vectors could exist in an economy.

**Theorem 1** *Let  $\vec{p}$  be an equilibrium price in the flow control economy. Then, the set of allocations*

$$\vec{\phi}_1(\vec{p}), \vec{\phi}_2(\vec{p}), \vec{\phi}_3(\vec{p}), \dots, \vec{\phi}_N(\vec{p})$$

*is pareto-optimal and unique.*

**Theorem 2** *There is an equilibrium price vector  $\vec{p}$  for the flow control economy.*

The proofs of these two theorems are given in [7, 9].

The flow control economy was also simulated to verify if the economy converged to an equilibrium. In each simulation, the network had 50 virtual circuits and 25 links. Figure 2

### Demand Convergence Add/Delete VCs

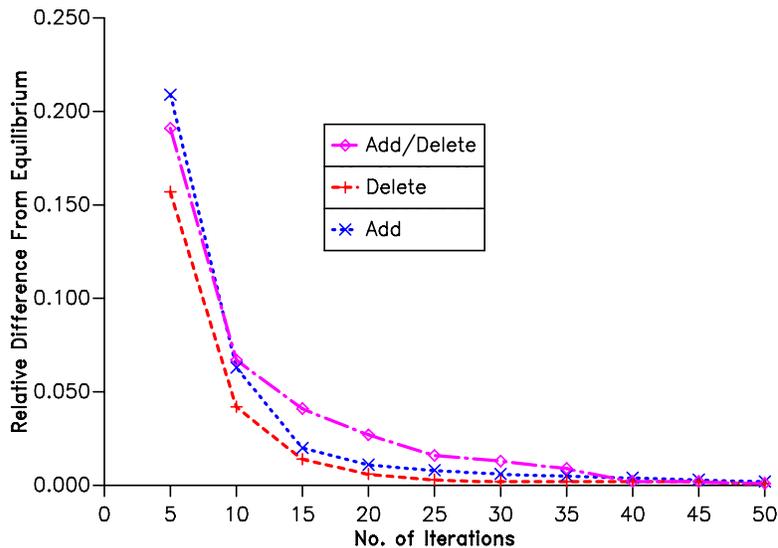


Figure 3: Convergence to Equilibrium: Flow Control Economy. This figure plots the average convergence of the economy to the new equilibrium after a new VC is activated or a VC terminates. The curves represent, adding 50 VCs 1 at a time, deleting 50 VCs one at a time, and a random ordering of 50 adds and deletes.

plots the convergence behavior of the economy for various VC throughput goals. The three economies differ by the probability of a link being in agent  $a$ 's path (denoted by  $Pr(i \in P_a)$ ). These values define random economies with different mean VC lengths as shown in the figure. The first conclusion drawn from the graph is that the economy converges to equilibrium very rapidly. The second conclusion is that the mean VC length has little effect on the rate of convergence. In another experiment, each VC's throughput demand  $\gamma_a$  is a random variable uniformly distributed on the interval  $[0, T]$ . The mean VC length was set to 5 links. The economy was shown to converge rapidly to an equilibrium.

Simulation based experiments were also performed to show the convergence of the economy to a new equilibrium when a VC is added or when a VC leaves the economy. Figure 3 shows the convergence behavior of the flow control economy. VCs are randomly generated in the economy. The plots in the figure represent the average number of iterations required to reach the new equilibrium. The three cases are: 1) Adding 50 VCs, 2) Deleting 50 VCs, and 3) a random ordering of 50 additions and deletions.

### 3.2.3 Data Management Economy

Unlike the flow control and load balancing economies where users maximized an utility function to compute the required allocation, this economy considers data migration, replication, and pricing strategies for a data management economy. The problem of data migration and replication was formulated in an economic setting by Ferguson [7, 8]. Transactions that entered the system for service were charged by the processors for read and write access to data objects. Processors also leased resources to other processors to make profit using the revenue they earned.

The distributed system consisted of  $N$  processing nodes connected via links. Each processor  $P_i$  ( $i \in [1, N]$ ) has rate  $r_i$  at which it can process operations on local data. A link  $e_{ij}$  connects processor  $P_i$  to  $P_j$ . There are  $M$  data objects denoted by  $D_1, D_2, \dots, D_M$ .  $S(D_i)$  defines the size of  $D_i$  in bytes. The economy treats these as abstract data objects. In a real system, they could correspond to *relations, tuples, files, records* or any other data structure.

The data management problem is to minimize the mean transaction response time with the following as control variables.

- Number of copies of data object.
- Assignment of copies to processing nodes.
- Pricing strategies of suppliers.

In the data management economy there are four types of agents. The consumers are transactions, and the suppliers are data object managers, local data agents and processors. The economy functioned in the following way. Each transaction  $T$  that arrived had an allocation of money  $M_T$ . Transactions pay to access data at a processor  $P_i$ . Data access is provided by the processor by leasing copies of data objects from data object managers. The local data agents act as an intermediary between a processor  $P_i$  and the object managers (remote).

A simple data management economy and system architecture is shown in Figure 4. There are three processors ( $P_1, P_2, P_3$ ) in this economy and three objects ( $A, B, C$ ). There may be multiple read copies of an object, but only one write copy exists. In this example,  $P_1$  has a read and a write copy of A, and  $P_1$  and  $P_2$  have a read copy.

Several numerical results via simulation were obtained which showed that the economy improved the response time per transaction compared to other approaches. Also this economy dynamically replicated data objects and their read/write capability based on the demand. The simulation was done on a 5 processor distributed system with 10 data objects,

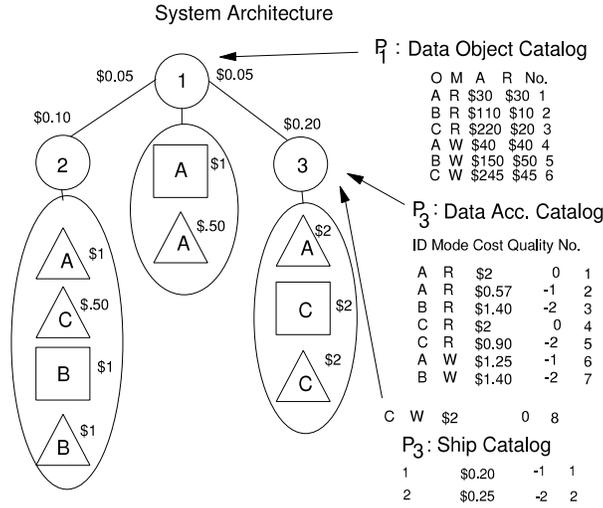


Figure 4: Data Management Economy: System Architecture

where each data object contained 1000 records. The processor service time was 1 data unit. The transmission delay was 2 time units for all links. Each access request required 1 message and shipping an entire data object required 1000 messages.

The processor pricing policy for local resources was static with the following prices: (1) read access is \$0.50, (2) write access is \$0.50, and (3) Link use: \$0.01 per message transmitted. Transactions entered the system with money \$10, and randomly submitted between 1 and 10 access operations. Each data object manager used an English auction to sell a single write copy. The manager's algorithm for setting the read renewal prices was to charge \$2 times the number of leases held.

Using simulations, the data management economy (DM) was compared to a single copy (SC), and a read everywhere (RE) approach<sup>2</sup>. Figure 5 shows the performance of the DM, SC and RE mechanisms where the mean transaction response times versus the write probability  $p_w$  is plotted. From the figure at  $p_w = x$ , a transaction that enters the system access the data

<sup>2</sup>The SC scheme has a single read/write copy of each object  $D_j$ . All read and write operations are sent to  $P_i$  that contains the copy. The copies are uniformly distributed. The RE scheme contains a read copy of every object on every  $P_i$ . However, there is a single write copy of each data object uniformly assigned to the processors.

## Response Time vs Write Probability

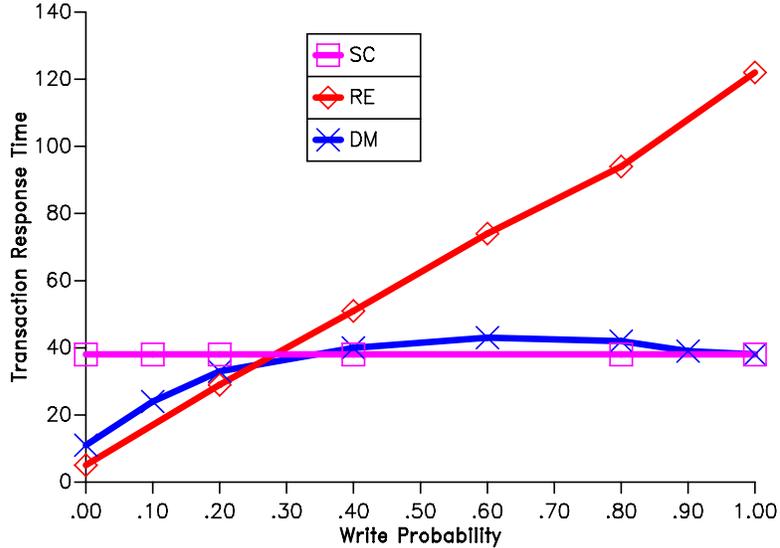


Figure 5: Adapting to Read Write Ratio: This figure shows the mean response times of the data management economy (DM), the single copy system (SC), and the read everywhere system (RE).

object  $D_j$  with  $p_w = x$ . The read everywhere (RE) policy exhibits the best response time for  $p_w \leq 0.30$ . The mean response time increases linearly with the write probability. The data management (DM) economy is no more than 13% worse than the minimum response time.

Two economic factors cause the data management economy to adapt the number of read copies of each object  $D_j$  to the read/write ratio. These are:

1. The total revenue that all processors earn by selling  $Read(D_j)$  decreases as  $p_w$  increases.
2. The read lease price for  $D_j$  increases linearly with the number of copies  $c(j)$ .

The data management economy uses decentralizes decision making to compute the number of read copies of each object. The business strategies of the processors are decoupled, and  $P_i$  only uses only local information to estimate its revenue. The economy adapts itself to any read/write ratio without any external intervention. The economy is not completely self tuning, however, there is a subtle interaction between the following factors:

- Lease price function.
- Transaction arrival rates.
- The wealth of the transactions.

### 3.2.4 Multiple Access Protocols

An exchange economy and a pricing model were considered by Kurose *et. al.* [12] to study the problem of optimal transmission probabilities in multiple access protocols. They found the optimal transmission probabilities for a channel sharing system using a decentralized approach based on pricing.

The utility function considered was the thruptut, where each user computed the optimal transmission probabilities based on the wealth they possessed, and their goal was to maximize the thruptut. The maximization problem is the following:

$$\begin{aligned}
 \textbf{find:} & & X_{i \rightarrow j}^i, X_{i \rightarrow j}^j \\
 \textbf{such that:} & & \max U_i = Psucc_i \\
 \textbf{constraints:} & & \sum_{j=1}^N p_{i \rightarrow j} X_{i \rightarrow j}^i + p_{j \rightarrow i} X_{j \rightarrow i}^i \leq w_i
 \end{aligned} \tag{11}$$

where  $X_{i \rightarrow j}^i$  is the transmission probability between station  $i$  and  $j$  (demanded by  $i$ ), and  $X_{i \rightarrow j}^j$  is the *silence* between station  $i$  and  $j$  (demanded by  $j$ ). The constraint for station  $i$  is:  $X_{i \rightarrow j}^i + X_{i \rightarrow j}^j = 1$ .  $Psucc_i$  is the probability of transmission success for station  $i$ ,  $p_{i \rightarrow j}$  is the price of transmission between station  $i$  and  $j$ , and  $w_i$  is the wealth of station  $i$  (where  $i \in [1, N]$ ).

The actual allocation of resources is done via a pricing scheme which changes with the demand. The price adjustment scheme is a *tatonement* process (described earlier). Prices are adjusted to clear the market. The allocation in equilibrium is proven to be pareto-optimal. These techniques were applied to study the optimal transmission probabilities in a *slotted aloha* protocol. Simulations and analysis were used to show that the economy converged to an equilibrium, and allocations were pareto-optimal.

### 3.2.5 Quality of Service in Computer Networks

With the emergence of multimedia networks, several types of applications such as video, image, voice, and data are integrated into one network. The problem of QoS (Quality of Service) provisioning in such an integrated network is challenging because of the heterogeneous requirements of the multimedia applications. This problem is considered in Sairamesh *et. al.* [17], where economic models are applied to study allocation of link capacity and network buffers to various types of applications.

The economy consists of  $K$  classes of sessions (or  $K$  traffic classes) competing for link resources at a switch (node) in the network. The link resources are buffer space and link

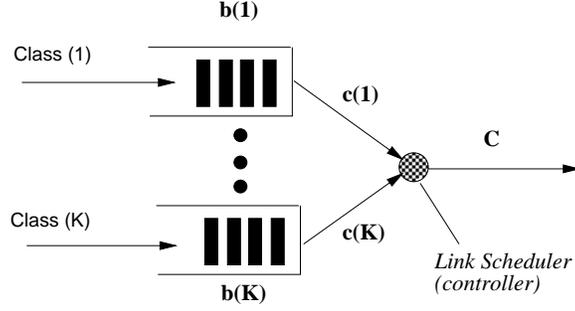


Figure 6: K traffic classes competing for buffer and link bandwidth resources at a single link at a single node.

capacity (each node is a supplier). Sessions that arrive are mapped to a traffic class based on similar traffic characteristics and QoS requirements. Each session arrives with some wealth and adds its wealth to the traffic class it joins.

Traffic classes have  $M/M/1/B$  [14] type queueing models ( Figure 6 shows the link model) The utility function of each traffic class is the packet loss probability in the  $M/M/1/B$  system, which is given by:

$$U = f(c, b, \lambda) = \begin{cases} \frac{(1-\frac{\lambda}{c})(\frac{\lambda}{c})^b}{1-(\frac{\lambda}{c})^{1+b}} & \lambda < c \\ \frac{1}{b+1} & \lambda = c \\ \frac{(-1+\frac{\lambda}{c})(\frac{\lambda}{c})^b}{-1+(\frac{\lambda}{c})^{1+b}} & \lambda > c \end{cases}$$

Each traffic class  $k$  (or an agent representing that traffic class) performs the following minimization (preference maximization) to place resource demands at a link.

$$\begin{aligned} \mathbf{find:} & \quad \{c_k^*, b_k^*\} \\ \mathbf{such\ that:} & \quad \min U_k = \min f(c_k, b_k, \lambda_k) \\ \mathbf{constraints:} & \quad p_b b_k + p_c c_k \leq w_k \\ & \quad c_k \in [0, C], \quad b_k \in [0, B], \quad \forall k \in [1, K] \end{aligned} \tag{12}$$

The resource constraints are  $\sum_{k=1}^{k=K} c_k = C$ , and  $\sum_{k=1}^{k=K} b_k = B$ , where  $C$  is the link capacity, and  $B$  is the switch buffer. Here,  $c_k, b_k$  are the allocation variables (link and buffer respectively) for class  $k$ , and  $w_k$  is the wealth of class  $k$ . Each traffic class  $k$  has an arrival rate  $\lambda_k$ .

Numerical results were obtained using the first order conditions of multi-objective optimization techniques [2, 3, 15] for a single node with maximum link capacity  $C = 60$  units

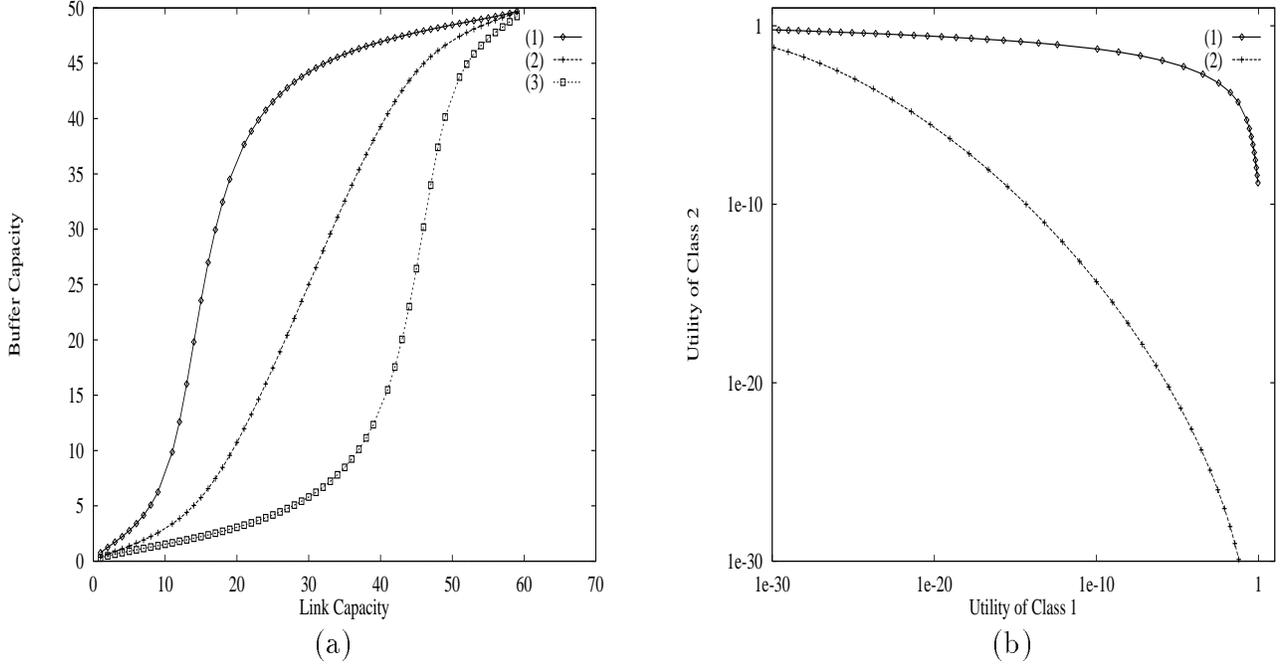


Figure 7: Pareto Surface show in an Edgeworth Box (a). The output link capacity is  $C = 60$  packets (or units) per second. The buffer space is  $B = 50$  packets (or units). Utility Surface of  $TC_2$  versus  $TC_1$  in logscale (b). This space corresponds to points on the pareto surface

per second, and  $B = 50$  storage units (packets or jobs). We assume that buffer allocation variables are continuous. In [17], it was shown that the utility function is decreasing convex with respect to  $c$  and  $b$ .

In Figure 7 (a), the pareto surfaces for three different traffic arrival rates<sup>3</sup> are shown for two competing traffic classes in an *Edgeworth box* diagram [2, 3, 15]. Pareto surface is a collection of pareto optimal points. A point  $(x, y)$  in the Edgeworth box corresponds to an allocation  $\{x, y\}$  to traffic class 1 ( $TC_1$ ) and an allocation  $\{C - x, B - y\}$  to traffic class 2 ( $TC_2$ ). Plot (2) shows the pareto surface for two classes of traffic that have same arrival rates ( $\lambda_1 = \lambda_2 = 10$ ), whereas in plot (1) the pareto surface is shown for different arrival rates ( $\lambda_1 = 10, \lambda_2 = 40$ ). It is clear from the plot (1) that  $TC_2$  which has a higher arrival rate needs more resources compared to  $TC_1$ . Therefore a shift in the pareto surface. Plot(3) is the just the reverse of plot(2), where  $\lambda_1 = 40, \lambda_2 = 10$ .

For the two traffic classes, the utility of  $TC_2$  versus the utility of  $TC_1$  is shown in Figure 7 (b). The points in the utility space are computed from the points on the pareto surface, which is shown in Figure 7 (a). In plot (1), the utility surface with two classes of traffic with the same arrival rates is shown. The surface is symmetric as the utility functions are the same. However, in plot (2), the arrival rates are different.  $TC_2$  ( $\lambda_2 = 40$ ) needs a lot more

<sup>3</sup>The jobs have exponential service time with unit mean.

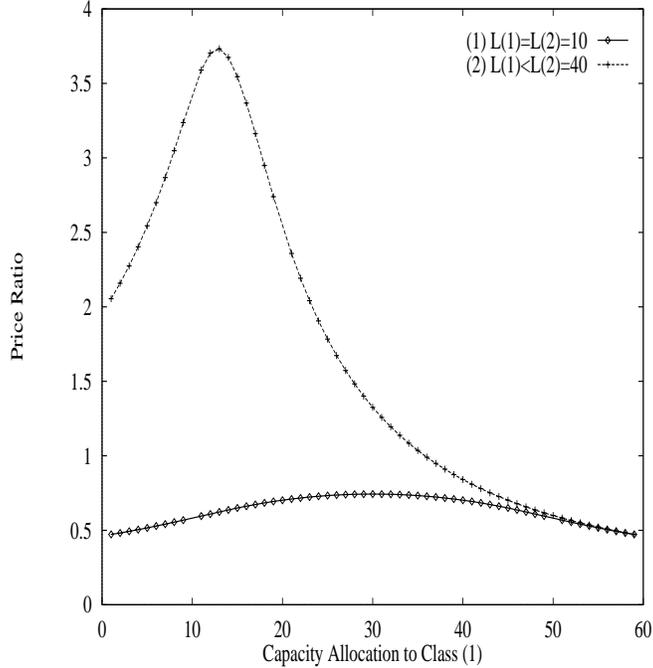


Figure 8: Equilibrium Price ratio ( $p_c/p_b$ ) versus  $c_1$ , curve (1):  $\lambda_1 = \lambda_2 = 10$ , curve (2):  $\lambda_1 = 10, \lambda_2 = 40$ , resources:  $C = 60, B = 50$

resources to have the same marginal utility as  $TC_1$ . These plots show the trade-offs in utility between the competing traffic classes. In Figure 8, the equilibrium price ratios (price of link capacity over the price of buffer) are shown for the same loading conditions. In plot (1), the traffic classes have the same arrival rates, therefore the price ratio curve is symmetrical with respect to the allocation to class (1). In plot (2), the arrival rate of  $TC_2$  is  $\lambda_2 = 40$ . The impact on loading is obvious,  $TC_2$  places a larger demand on the resources (buffer and link). From the plot, it is evident that the price of link capacity has increased substantially with respect to the price of buffer.

The QoS economy provided valuable insight into the pareto optimal allocations of link capacity and buffer to traffic classes at a switch. A set of these allocations provided QoS guarantees to the traffic classes. In [17], the equilibrium prices were determined for the resources under the QoS constraints. The impact on equilibrium prices and optimal allocations due to traffic load and variability was also investigated. Using the economic models practical packet scheduling and session admission policies which provide QoS guarantees to traffic classes were proposed.

## 4 Discussion and Future Work

Several novel techniques of resource allocation using various economic concepts have been presented in this paper. The case studies show the application of economic models to computer systems (distributed systems and computer networks). In particular the mechanisms by which the consumers (jobs, tasks, transactions, traffic classes, etc.) interact with the suppliers of resources (processors, communication links, transmission channels, data object, network buffers) to obtain access to the resources is discussed in detail with several numerical results (via simulations) and analysis to support the effectiveness of the computer system economies. The two level economic modeling of computer and communication resource sharing is discussed in several of the case studies. In the first level, the resource sharing problem is transformed into an appropriate performance model (example: queueing model) with the service parameters (at each supplier) to optimize and control. In the second level, the performance model of the system is transformed into an economy where economic optimization techniques and mechanisms were used to obtain the efficient utilization of resources in a decentralized manner.

In this paper, we have demonstrated the application of economic theories to resource management problems in distributed systems and computer networks. The concepts of mathematical economics were used to develop effective market based control mechanisms, and to show that the allocation of resources are pareto optimal. However, there are drawbacks to this form of modeling where several agents have to use market mechanisms to decide where to obtain service (which supplier?). If the demand for a resource varies substantially over short periods of time, then the actual prices of the resources will also vary causing several side effects such as indefinite *migration* of jobs between suppliers, and price guarantees over shorter periods of time. This might potentially result in degradation of system performance where the resources are being underutilized due to the bad decisions (caused by poor market mechanisms) made by the users in choosing the suppliers. Unlike human economies, the resources in a computer system are not that easily substitutable. The future work is to design robust market mechanisms and rationalized pricing schemes which can handle surges in demand and variability, and can give price guarantees to consumers over longer periods of time. Another drawback is that resources in a computer system are indivisible resulting in non-smooth utility functions, which may yield sub-optimal allocations, and potential computational overhead.

Recently, Stonebraker *et. al.* [23] have looked at the problem of efficient query processing and data migration in large database systems which involve thousands of data sites (nodes). They chose an economic paradigm similar to the one described in [7, 8, 9] to design a

distributed database system: **Mariposa**, which can handle several thousand data sites. They designed efficient bidding mechanisms for clients to access the data sites. This is the *first* known working system which is based on an economic paradigm. Economic models can also be used for service provisioning in the *Internet* (which is always growing). Internet currently connects millions of users and hundreds of sites. Several services exist on many of these sites, notably *The World Wide Web* [24], which provides access to various information sources distributed across the Internet. Many more services (example: multimedia applications) are envisioned to be supported in the Internet. To access these large number of services, agents have to share limited network bandwidth and server capacities (processing speeds). Such gigantic networks require decentralized mechanisms to control access to services. Economic concepts such as pricing and competition can provide some solutions to reduce the complexity of service provisioning and decentralize the access mechanisms to the resources.

## References

- [1] K. Arrow *et. al.*, *Handbook of Mathematical Economics*, vol I and vol II, North Holland Publishing, Amsterdam, 1981.
- [2] G. Debreu, *The Theory of Value*, NewYork, Wiley, 1959.
- [3] A. Mas-Colell, *The Theory of General Economic Equilibrium: A Differentiable Approach*, Cambridge University Press, 1985.
- [4] M. Shubik, *Game Theory in Social Sciences*, The MIT Press , Cambridge, Mass., 1983.
- [5] Y. Yemini and L. Kleinrock, "On a General Rule for Access Control, or Silence is Golden...", *Proc. of the International Conf. on Flow Control in Computer Networks*, pg 335-347, North Holland Press Amsterdam, Versailles, February 1979.
- [6] Y. Yemini, "Selfish Optimization in Computer Networks," *Proceedings of the 20th IEEE Conf. on Decision and Control*, pages 281-285, San Diego, Decemeber, 1981.
- [7] D.F. Ferguson, "The Application of Microeconomics to the design of resource allocation and control algorithms in Distributed Systems," *PhD thesis*, Columbia University, New York, 1989.
- [8] D.F. Ferguson, C. Nikolaou and Y. Yemini, "An Economy for Managing Replicated Data in Autonomous Decentralized Systems," *Proc. Int. Symp. on Autonomous Decentralized Sys. (ISADS'93)*, Kawasaki, Japan, 1993.

- [9] D.F. Ferguson, C. Nikolaou and Y. Yemini, "An Economy for Flow Control in Computer Networks," *Proc. of the INFOCOM'90*.
- [10] D.F. Ferguson, Y. Yemini and C. Nikolaou, "Microeconomic Algorithms for Load Balancing in Distributed Computer Systems," *Proc. of International Conference on Distributed Systems (ICDCS'88)*.
- [11] J. Jaffe, "Flow Control power is non-decentralizable," *IEEE Trans. on Commun.* vol, COM-29, no. 7, September 1981.
- [12] J.F. Kurose, M. Schwartz and Y. Yemini, "A microeconomic approach to optimization of channel access policies in multiaccess networks," *Proc. of 5th Int. Conf. Distrib. Comput. Syst.*, Denver, May 1985.
- [13] J.F. Kurose and R. Simha, "A Microeconomic approach to optimal resource allocation in distributed computer systems," *IEEE Trans. on Comp.*, May 1989.
- [14] L. Kleinrock, *Queueing Systems: Volume 1: Theory*, Wiley Interscience, New York, 1975.
- [15] H. R. Varian, *Microeconomic Analysis*, 3rd ed., Norton Press, 1992.
- [16] M. Miller and E. Drexler, "Markets and Computation: Agoric Open Systems," *The Ecology of Computation*, North Holland, Amsterdam, 1988.
- [17] J. Sairamesh, D. Ferguson, and Y. Yemini, "An Approach to Pricing, Optimal Allocation and Quality of Service Provisioning in High Speed Packet Networks," *Proceedings of the INFOCOM'95*.
- [18] A. Bovopoulous and A. A. Lazar, "Decentralized Algorithms for Optimal Flow Control," *Proc. of the 25th Annual Allerton Conference on Communication, Control and Computing*, September 1987.
- [19] M.T. Hsiao and A.A. Lazar, *Optimal Decentralized Flow Control in Computer Communication Networks*, Phd thesis, CTR-Tech report, Columbia University, NY, 1986.
- [20] Y. Korilis, A. A. Lazar, and A. Orda, "On the Existence of Equilibria in Noncooperative Optimal Flow Control," *Proc. of the ITC Workshop*, Bangalore, India, 1993.
- [21] C. Douglaris and R. Mazumdar, "On Pareto Optimal Flow Control in a Multiclass Environment," *CTR Technical Report, Columbia University*, 1987.

- [22] S. Shenker, "Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines," *Proc. of the SIGCOMM'94*, pg 47-57, August 1994.
- [23] M. Stonebraker *et. al.*, "An Economic Paradigm For Query Processing and Data Migration in Mariposa," *Proc. of Parallel and Distributed Information Systems*, pages 58-67, 1994.
- [24] T. Berners-Lee *et. al.*, "The World Wide Web," *Communications of the ACM*, August 1994, Vol 37. No. 8, pages 76-82.
- [25] R. Cocchi *et. al.*, "A Study of priority pricing in multiclass networks," *Proc. of the SIGCOMM'91*.
- [26] M. Livny, *The Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems*, PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1984.
- [27] M. Livny and M. Melman, "Load Balancing in Homogeneous Distributed Systems," *Proc. of the ACM Computer Networking Perf. Symposium*, pages 47-55, April 1982.
- [28] C. A. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and S. Stornetta, "Spawn: A distributed computational ecology," *IEEE Trans. on Software Engg.*, 18(2), February, 1992.
- [29] B. Sanders, "An Asynchronous Distributed Flow Control Algorithm for Rate Allocation in Computer Networks," *IEEE Trans. on Computers*, Vol 37, No. 9, September 1988.