iBusiness, 2013, 5, ***-***
Published Online September 2013 (http://www.scirp.org/journal/ib)

Quality of Service on Queueing Networks for the Internet*

Hans W. Gottinger

STRATEC, Munich, Germany.
Email: hg528@bingo-ev.de

Received *************** 2013

ABSTRACT

Most studies of resource allocation mechanisms in Internet traffic have used a performance model of the resource provided, where the very concept of the resource is defined in terms of measurable qualities of the service such as utilization, throughput, response time (delay), security level among others. Optimization of resource allocation is defined in terms of these measurable qualities. One novelty introduced by an economic mechanism design approach is to craft a demand-driven system which takes into account the diverse QoS requirements of users, and therefore use multiobjective (utility) optimization techniques to characterize and compute optimum allocations. Economic modelling of computer and communication resource sharing uses a uniform paradigm described by two level modelling: QoS requirements as inputs into a performance model that is subject to economic optimization.

Keywords: Internet Economics; Network Economy; Queueing Systems; Mechanism Design; Performance Management

1. Introduction

In the context of congested networks for Internet traffic the phenomenon of packet loss is due to two reasons: the first, packets arrive at a switch and find that the buffer is full (no space left), and therefore are dropped. The second is that packets arrive at a switch and are buffered, but they do not get transmitted (or scheduled) in time, then they are dropped. A formal way of saying this: for real-time applications, packets, if delayed considerably in the network, do not have value once they reach the destination, a job that misses the deadline may have no value at all. The sort and variety of delay can severely impact the operability and efficiency of a network and therefore is of eminent interest for economic analysis (Radner [1], van Zandt [2], Mount and Reiter [3]).

A way to look at the network economy is to invoke mechanism design principles supporting market mechanisms (Deng *et al.* [4]; Neumann *et al.* [5]).

The present paper builds on previous work (Gottinger [6]) and expands on Quality of Service principles and management procedures toward creating demand for service in the Internet economy.

1.1. Quality of Service (QoS)

With the Internet we observe a single quality of service (QoS): "best effort packet service." Packets are transported first come, first-served with no guarantee of success. Some packets may experience severe delays, while others may be dropped and never arrive. Different kinds of data place different demands on network services (Shenker, [7]). Email and file transfer requires 100 percent accuracy, but can easily tolerate delay. Real-time voice broadcasts require much higher bandwidth than file transfers, and can tolerate minor delays but they cannot tolerate significant distortion. Real-time video broadcasts have very low tolerance for delay and distortion. Through the widespread usage of VOIP the Internet has become more and more data sensitive. Because of these different requirements, network allocation algorithms should be designed to treat different types of traffic differently but the user must truthfully indicate which type of traffic he/she is preferring, and this would only happen through incentive compatible pricing schemes which is at the very core of economic mechanism design.

[*] "*In the* '*real world*' *the invisible hand of free markets seems to yield surprisingly good results for complex optimization problems. This occurs despite the many underlying difficulties: decentralized control, uncertainties, information gaps, computational power, etc. One is tempted to apply similar market based ideas in computational scenarios with similar complications, in the hope of achieving similarly good results.*"
—Noam Nisan, "Algorithms for Selfish Agents: Mechanism Design for Distributed Computation", in STACS 99 Trier, Springer: Berlin 1999.

An early pathbreaking analytical framework for invoking economic design principles into the present day Internet has been proposed by Ferguson *et al.* [8]. As measurable ingredients for QoS they identified performance criteria such as average response time, maximum response time, throughput, application failure probability and packet loss. QoS can be affected by various factors, both quantitative (*i.e.*, latency, CPU performance, storage capacity etc.) and qualitative that proliferate through reputation systems hinging on trust and belief and as the Internet matures in promptness, reliability, availability and foremost security for a certain quality service level targeted, thus embracing a "Generalized Quality of Service" level. If you relate performance to utility as a multiplicative factor, performance varies over the range of utility in [0,1]. This would result in service level arrangements (SLAs) comprising service reliability and user satisfaction (Macias *et al.* [9]).

From the work of Ferguson *et al.* [8] network pricing could be looked at as a mechanism design problem. The user can indicate the "type" of transmission and the workstation in turn reports this type to the network. To ensure truthful revelation of preferences, the reporting and billing mechanism must be incentive compatible.

1.2. A Simple Mechanism Design

There are k agents in an Internet economy that collectively generate demand competing for resources from a supplier. The supplier herself announces prices entering a bulletin board accessible to all agents (as a sort of transparent market institution). In a simple form of a trading process we could exhibit a "tatonnement process" on a graph where the agents set up a demand to the supplier who advertisea prices on a Bulletin Board which are converted to new prices in interaction with the agents.

The tatonnement process in economics is a simple form of an algorithmic mechanism design (AMD), Nisan and Ronen [10], that in modern computer science emerged as an offspring to algorithmic game theory (Nisan *et al.* [11]).

The approach to mechanism design would enable users of applications to present their QoS demands via utility functions defining the system performance requirements.

The resource allocation process involves economic actors to perform economic optimization given scheduling policies, load balancing and service provisioning.

Along this line such approaches have been the basis of business models for grid and cloud service computing (Mohammed *et al.* [12]).

Distributed algorithmic mechanism design for internet resource allocation in distributed systems is akin to an equilibrium converging market based economy where selfish agents maximize utility and firms seek to maximize profits and the state keeps an economic order providing basic public goods and public safety (Feigenbaum *et al.* [13]).

In fact, it's a sort of Hayek type mechanism limited to a special case of a diversified Internet economy (Myerson [14]).

1.3. Pricing Congestion

The social cost of congestion is a result of the existence of network externalities. Charging for incremental capacity requires usage information. We need a measure of the user's demand during the expected peak period of usage over some period, to determine the share of the incremental capacity requirement. In principle, it might seem that a reasonable approach would be to charge a premium price for usage during the pre-determined peak periods (a positive price if the base usage price is zero), as is rountinely done for electricity pricing (Wilson [15]). However, in terms of internet usage, peak demand periods are much less predictable than for other utility services. Since the use of computers would allow to schedule some activities during off-peak hours, in addition to different time zones around the globe, we face the problem of shifting peaks. By identifying social costs for network externalities, the suggestion by MacKie-Mason and Varian [16] is directed toward a scheme for internalizing this cost as to impose a congestion price that is determined by a real-time Vickrey auction. The scheme requires that packets should be prioritized based on the value that the user puts on getting the packet through quickly. To do this, each user assigns his/her packets a bid measuring his/her willingness-to-pay (indicating effective demand) for immediate servicing. At congested routers, packets are prioritized based on bids. In line with the design of a Vickrey auction, in order to make the scheme incentive compatible, users are not charged the price they bid, but rather are charged the bid of the lowest priority packet that is admitted to the network. It is well-known that this mechanism provides the right incentives for truthful revelation. Such a scheme has a number of desirable characteristics. In particular, not only do those users with the highest cost of delay get served first, but the prices also send the right signals for capacity expansion in a competitive market for network services. If all of the congestion revenues are reinvested in new capacity, then capacity will be expanded to the point where its marginal value is equal to its marginal cost.

2. Quality of Service Parameters

2.1. Internet Communication Technologies

The Internet and Asynchronous Transfer Mode (ATM) have strongly positioned themselves for defining the future information infrastructure. The Internet is successfully operating one of the popular information systems, the World Wide Web (WWW), which suggests that the information highway is forming on the Internet. However, such a highway is limited in the provision of advanced multimedia services such as those with guaranteed quality of service (QoS). Guaranteed services are easier to support the ATM technology. Its capability far exceeds that of the current Internet, and it is expected to be used as the backbone technology for the future information infrastructure. ATM proposes a new communications paradigm. ATM allows integration of different types of services such as digital voice, video and data in a single network consisting of high speed links and switches. It supports a Broadband Integrated Services Digital Network (B-ISDN), so that ATM and B-ISDN are sometimes used interchangeably, where ATM is referred to as the technology and B-ISDN as the underlying technical standard. ATM allows efficient utilization of network resources, and simplifies the network switching facilities compared to other proposed techniques in that it will only require one type of switching fabric (packet switch). This simplifies the network management process. The basic operation of ATM, and generally of packet-switched networks, is based on statistical multiplexing. In order to provide QoS, the packets need to be served by certain scheduling (service) disciplines. Resource allocation algorithms depend heavily on the scheduling mechanism deployed. The scheduling is to be done at the entrance of the network as well as the switching points. The term “cell” designates the fixed-size packet in ATM networks. ATM allows variable bit rate sources to be statistically multiplexed. Statistical multiplexing produces more efficient usage of the channel at the cost of possible congestion at the buffers of an ATM switch. When the congestion persists, buffer overflow occurs, and cells are discarded (or packets are dropped). Therefore, resources (*i.e.* bandwidth and buffer space) need to be carefully allocated to meet the cell loss and the delay requirements of the user (Gottinger [17]).

The delay and the cell loss probability that the user wishes the network to guarantee are referred to as the QoS parameters. Overall, QoS is usually defined in terms of cell loss probability, delay bounds and other delay and drop-off parameters. How can one provide such QoS with guarantees? The general approach is to have an admission or performance algorithm that takes into account the traffic characteristics of the source and assigns suitable amounts of resources to the new connection during channel establishment. The admission algorithm is responsible for calculating the bandwidth and buffer space necessary to meet the QoS requirements specified by the user. The algorithm depends on how the traffic is characterized and the service disciplines supported by the switches.

Although the Internet is capable of transporting all types of digital information, it is difficult to modify the existing Internet to support some features that are vital for real time communications. One important feature to be supported is the provision of performance guarantees. The Internet uses the Internet Protocol (IP), in which each packet is forwarded independently of the others. The Internet is a connectionless network where any source can send packets any time at speeds that are neither monitored nor negotiated. Congestion is bound to happen in this type of network. If congestion is to be avoided and real-time services are to be supported, then a negotiation (through pricing or rationing) between the user and the network is necessary. ATM (Asynchronous Transfer Mode) is a connection-oriented network that supports this feature. A virtual channel is established, and resources are reserved to provide QoS prior to data transfer. This is referred to as channel establishment.

2.2. Traffic in B-ISDN

In a B-ISDN environment high-bandwidth applications such as video, voice and data are likely to take advantage of compression. Different applications have different performance requirements, and the mechanisms to control congestion should be different for each class of traffic. Classification of these traffic types is essential in providing efficient services. There are two fundamental classes of traffic in B-ISDN: real-time and non real-time (best effort) traffic. The majority of applications on the Internet currently are non-real-time ones based on TCP/IP. TCP/IP is being preserved with an ATM technology. The Internet can support data traffic well but not real-time traffic due to the limitations in the functionality of the protocols. B-ISDN needs to support both non-real-time and real-time traffic with QoS guarantees. Most data traffic requires low cell loss, but is insensitive to delays and other QoS parameters. Standard applications such as Telnet require a real-time response and should therefore be considered real-time applications. Video is delay-sensitive and, unlike Telnet, requires high bandwidth. High throughput and low delay are required of the ATM switches for the network to support video services to the clients. This puts a constraint on the ATM switch design in that switching should be done in hardware and the buffer sizes should be kept reasonably small to prevent long delays. On the other hand, best effort traffic tends to be bursty, and its traffic characteristics are hard to predict. This puts another, opposite constraint on an ATM switch, which requires large buffers at the switching point, further complicating its design.

2.3. Congestion Control

Statistical multiplexing can offer the best use of resources, however, this is done at the price of possible congestion. Congestion in an ATM network can be handled basically in two ways: reactive control and preventive control. Reactive control mechanisms are commonly used in the Internet, where control is triggered to alleviate congestion after congestion has been detected. Typical examples of reactive control are 1) explicit congestion notification (ECN), 2) node to node flow control and 3) selective cell discarding. In the more advanced preventive control approach, congestion is avoided by allocating the proper amount of resources and controlling the rate of data transfers by properly scheduling cell departures. Some examples of preventive control mechanisms are 1) admission and priority control, 2) usage parameter control and 3) traffic shaping. Appropriate mathematical tools rooted in AMD are available for congestion control (Srikant [18]).

Reactive and preventive control can be used concurrently, but most reactive controls are unsuitable for high-bandwidth real-time applications in an ATM network, since reactive control simply is not fast enough to handle congestion in time. Therefore, preventive control is more appropriate for high speed networks.

2.4. Service Discipline

Traffic control occurs at various places in the network. First, the traffic entering the network is controlled at the input, second, the traffic is controlled at the switching nodes. In either case, traffic is controlled by scheduling the cell departures. There are various ways how to schedule departure times, and these mechanisms are part of service disciplines. The service discipline must transfer traffic at a given bandwidth by scheduling the cells and make sure that it does not exceed the buffer space reserved (or the delay bound assigned) for each channel. These functions are usually built into the hardware of the ATM switch and into the switch controller. When implementing a service discipline in an ATM network, it is important to choose it simple enough that it can be easily integrated into an ATM switch. However, the discipline must support the provision of quality of service guarantees. This also means that the service discipline is responsible for protecting “well-behaved” traffic from the “ill-behaved” traffic and must be able to provide certain levels of QoS guarantees. The service discipline also needs to be flexible enough to satisfy the diverse requirements of a variety of traffic types, and to be efficient, that is, to permit a high utilization of the network. Various service disciplines have been proposed, and many of them have been investigated thoroughly and compared. An important class is that of disciplines used in rate-allocating servers.

2.5. Bandwidth-Buffer Tradeoff

A simple example on the representation of QoS parameters is the bandwidth-buffer tradeoff. Bandwidth can be traded for buffer space and vice versa to provide the same QoS. If a bandwidth is scarce, then a resource pair that uses less bandwidth and more buffer space should be used. Resource pricing is targeted to exploit this tradeoff to achieve efficient utilization of the available resources. The pricing concept for a scarce resource is well-known in economics, but in the context of exploiting the bandwidth-buffer tradeoff, Low and Varaiya [19] use non-linear optimization theory to determine centralized optimal shadow prices in large networks, it shows widespread applicability of mechanism design theory (Vohra [20]). With respect to large scale application, however, the complex optimization process limits the frequency of pricing updates, which causes inaccurate information about available resources. In order to make pricing in the context of a buffer-bandwidth tradeoff more adjustable and flexible it should be based on decentralized pricing procedures according to competitive bidding in large markets where prices will be optimal prices if the markets are efficient. This would also allow flexible pricing which results in accurate representation of available resources in that prices are updated as the instance connect request arrives. The subsequent procedure is based on distributed pricing as a more feasible alternative to optimal pricing.

3. Network Economy

The economic mechanism design model consists of the following players: Agents and Network Suppliers. Consumers or user classes: Consumers (or user classes) request for QoS. Each user class has several sessions (or user sessions). Users within a class have common preferences. User classes have QoS preferences such as preferences over packet-loss probability, max/average delay and throughput. Users within an class share resources.

Agents and Network Suppliers: Each user class is represented by an agent. Each agent negotiates and buys services (resource units) from one or more suppliers. Agents demand for resources in order to meet the QoS needs of the user classes. Network providers have technology to partition and allocate resources (bandwidth and buffer) to the competing agents. In this competitive setting, network providers (suppliers) compete for profit maximization.

Multiple Agent-Network Supplier Interaction: Agents present demands to the network suppliers. The demands are based on their wealth and QoS preferences of their class. The demand by each agent is computed via utility functions which represent QoS needs of the user classes. Agents negotiate with suppliers to determine the prices. The negotiation process is iterative, where prices are adjusted to clear the market; supply equals the demand. Price negotiation could be done periodically or denpending on changes in demand.

Each agent in the network is allocated a certain amount of buffer space and link capacity. The buffer is used by the agent for queueing packets sent by the users of the class. A simple FIFO queueing model is used for each class. The users within a class share buffer and link resources. This makes sense to create and maintain network stablity in large networks (Weinard [21]).

Agent and supplier optimality: Agents compete for resources by presenting demand to the supplier. The agents, given the current market price, compute the affordable allocations of resources (assume agents have limited wealth or budget). The demand from each agent is presented to the supplier. The supplier adjusts the market prices to ensure demand equals supply.

The main issues from the economic model are:

- Characterization of class QoS preferences and traffic parameters via utility functions, and computation of demand sets given the agent wealth and the utility function.
- Existence and computation of Pareto optimal allocations for QoS provisioning, given the agent utility functions.
- Computation of equilibrium price by the supplier based on agent demands, and conditions under which price equilibrium exists. Price negotiation mechanisms between the agents and suppliers.

This is in adjustment to computing classical economic equilibrium models such as in Scarf [22].

3.1. Problem Formulation

Network model: The network is composed of nodes (packet switches) and links. Each node has several output links. Each output link is associated with an output buffer. The link controller, at the output link, schedules packets from the buffers and transmits them to other nodes in the network. The switch has a buffer controller that can partition the buffer among the traffic classes at each output link. We assume that a processor on the switch aids in control of resources.

We have confined ourselves to problems for a single link (output link) at a node, but they can be applied to the network as well. Let B denote the output buffer of a link and C be the corresponding link capacity. Let $\{c_k, b_k\}$be the link capacity and buffer allocation to class k on a link, where $k \varepsilon [1, K]$.

Let

$$p = \{p_c, p_b\}$$

be the price per unit link capacity and unit buffer at a link, and w_k be the wealth (budget) of traffic class k. The utility function for TC_k is

$$U_k = f(c_k, b_k, Tr_k).$$

The traffic of a class is represented by a vector of traffic parameters (Tr_k) and a vector of QoS requirements (such as packet loss probabilities, average packet delay and so on.).

Agent (TC: traffic class) buys resources from the network at the given prices using its wealth. The wealth constraint of agent TC_k is:

$$p_b * b_k + p_c * c_k \le w_k.$$

A budget set is the set of allocations that are feasible under the wealth constraint (budget constraint). The budget set is defined as follows:

$$B(p) = (x : x\varepsilon X, px \le w_k\} \tag{1}$$

Computation of demands sets: The demand set for each agent is given by the following:

$$\Phi(p) = (x : x\varepsilon B(p), U(x'), \forall x'\varepsilon B(p)\} \tag{2}$$

As set up by Ferguson *et al.* [8] the goal of TC_k is to compute the allocations that provide maximal preference under w_k and p. Each TC_k performs the following to obtain the demand set (defined above): solve $\{c_k, b_k\}$ such that

$$\max U_k = f(c_k, b_k, Tr_k)$$

and budget constraints

$$p_b b_k + p_c c_k \le w, c_k \varepsilon [0, C], b_k \varepsilon [0, B]$$

3.2. Utility Parameters

In the previous section, we showed a general utility function which is a function of the switch resources; buffer (b) and bandwidth (c). The utility function could be a function of the following:

- Packet loss expected utility $U_t = g(c, b, Tr)$
- Average packet delay $U_d = h(c, b, Tr)$
- Packet tail utility $U_t = v(c, b, Tr)$
- Max packet delay $U_b = f(b, b_T)$
- Throughput $U_c = g(c, c_t)$
- Security level $U[\Pr(|m - m'|)]$ where m' is a compromised message accessible by other agents , and $|m - m'| = 0$ is true message preserving.

The variables b and c in the utility functions refer to buffer space allocation and link bandwidth allocation. In the utility functions U_b and U_c; the parameters b_T and c_T are constants. For example, the utility function

$$U_b = f(b, b_T)$$

for max packet delay is simply a constant as b increases, but drops to 0 when $b = b_T$ and remains zero for any further increase in b.

We look at utility functions which capture packet loss probability of QoS requirements by traffic classes, and we consider loss, max-delay and throuput requirements.

After this we proceed to utility functions that capture average delay requirements, followed by utility functions that capture packet tail utility requirements. We also could give examples of utility functions for agents with multiple objectives; agents have preferences over several QoS parameters.

3.3. Packet Loss

The phenomenon of packet loss is due to two reasons: the first, packets arrive at a switch and find that the buffer is full (no space left), therefore, are dropped. The second is that packets arrive at a switch and are buffered, but they do not get transmitted (or scheduled) in time, then they are dropped. A formal way of saying this: for real-time applications, packets, if delayed considerably in the network, do not have value once they reach the destination.

A proper way to deal with it is through queueing systems.

We consider *K* agents, representing traffic classes of M/M/1/B type, competing for resources from the network provider. The utility function is packet loss utility (U_1) for the user classes. We choose the M/M/1/B model or traffic and queueing for the following reasons:

- The model is tractable, where steady state packet loss utility is in closed-form, and differentiable. This helps in demonstrating the economic models and the concepts.
- There is interest in M/M/1/B or M/D/1/B models for multiplexed traffic (such as video), where simple histogram based traffic models capture the performance of queueing in networks (Kleinrock [23]).

In this case we look at the simple queueing system, for example with Poisson inputs, and exponential interarrival and general service time distribution B at a single server (Kleinrock and Gail [24] p. 7,21).

For more complex traffic and queueing models (example of video traffic) we can use tail utility functions to represent QoS of the user class instead of loss utility.

In the competitive economic model, each agent prefers less packet loss, as the more packet loss, the worse the quality of the video at the receiving end. Let each agent TC_k have wealth w_k, which it uses to purchase resources from network provider.

Let each TC transmit packets at a rate λ (Poisson arrivals), and let the processing time of the packets be exponentially distributed with unit mean. Let *c*, *b* be allocations to a TC. The utility function *U* for each TC is then a function of the allocations and the Poisson arrival rate.

3.4. Loss Probability Requirement: Utility Function

In view of queueing discipline, we consider *K* agents, representing traffic classes of M/M/1/B type, competing for resources from the network provider. The utility function is packet loss probability (U_t) for the user classes. We choose the M/M/1/B model of traffic and queueing for the following reasons. The model is tractable, where steady state packet loss probability is in closed form, and differentiable. This helps in demonstrating the economic models and concepts. Models such as M/M/1/B or M/D/1/B for multiplexed traffic (such as video) are appropriate where simple histogram based traffic models capture the performance of queueing in networks.

For more complex traffic and queueing models (say, example of video traffic) we can use tail probability functions to represent QoS of the user class instead of loss probability. In the competitive economic model, each agent prefers less packet loss, the more packet loss the worse the quality of the video at the receiving end. Let each agent TC_k have wealth w_k which it uses to purchase resources from network providers.

Let each TC transmit packets at a rate λ (Poisson arrivals), and let the processing time of the packets be exponentially distributed with unit mean. Let *c*, *b* be allocations to a TC. The utility function *U* for each TC is given as follows:

$$U = f(c,b,\lambda) = \left\{ \begin{array}{l} (1-\lambda/c)(\lambda/c)^b \big/ (1-(\lambda/c))^{1+b}, \text{if } \lambda < c \\ 1\big/(b+1), \text{if } \lambda = c \\ (-1+\lambda/c)\big/(\lambda/c)(\lambda/c)^b \big/ -1+(\lambda/c)^{1+b}, \text{if } \lambda > c \end{array} \right\}$$

The above function is continuous and differentiable for all $c \in [0,C]$, and for all $b \in [0,B]$. We assume $b \in \Re$ for continuity purposes of the utility function.

3.5. Loss Probability Constraints

The loss probability constraint is defined as follows: it is the set of (bandwidth, buffer) allocations

$$\{x : x \in X, U(x) \le L^c\}$$

where $U(x)$ is the utility function (loss probability function where lower loss is better) and L^c is the loss constraint. The preferences for loss probability are convex with respect to buffer and link capacity.

Computation of the QoS Surface by Supplier: assume that the supplier knows the utility functions of the agents, which represent the QoS needs of the traffic classes, then the supplier can compute the Pareto surface, and find out the set of Pareto allocations that satisfy the QoS constraints of the two agents.

This set could be a null set, depending on the constraints and available resources.

The QoS surface can be computed by computing the points A and B with a bandwidth-buffer space pair on the burstiness curve used for resource allocation. The burstiness curve represents the buffer size necessary to avoid cell losses at each service rate level.

Point A is computed by keeping the utility of (say) class 1 constant at its loss constraint and computing the Pareto-optimal allocation by maximizing the preference of (say) class 2. Point B can be computed in the same way. The QoS surface is the set of allocations that lies in [A, B]. The same technique can be used to compute the QoS surface when multiple classes of traffic compete for resources. There are situations where the loss constraints of both the traffic classes cannot be met. In such cases, either the demand of the traffic classes must go down or the QoS constraints must be relaxed. This issue is treated as an admission control problem, where new sessions are not admitted if the loss constraints of either class is violated.

3.6. Max and Average Delay Requirements

A max delay constraint simply imposes a constraint on the buffer allocation, depending on the packet sizes. If the service time at each switch for each packet is fixed, the max delay is simply the buffer size or a linear function of buffer size. Once the QoS surface for loss probability constraints are computed, then the set of allocations that meet the buffer constraint will be computed. This new set will provide loss and max delay guarantees. A traffic class will select the appropriate set of allocations that meet the QoS requirements under the wealth constraint.

A class of interesting applications would require average delay constraints on an end-to-end basis. Some of these applications include file transfers, image transfers, and lately Web based retrieval of multimedia objects. Consider a traffic model such as M/M/1/B for each traffic class, and consider that several traffic classes (represented by agents) compete for link bandwidth and buffer resources at a link with QoS demands being average delay demands.

Let us now transform the average delay function into a normalized average delay function for the following reasons: average delay in a finite buffer is always less than the buffer size. If a user class has packet loss probability and average delay requirements, then buffer becomes an important resource, as the two QoS parameters are conflicting with respect to buffer. In addition, the switch buffer needs to be partitioned among the traffic classes. Another way to look at this: a user class can minimize the normalized average delay to a value that will be less than the average delay constraint. This normalized average delay function for an M/M/1/B performance model, for an agent, is shown below:

$$(*)U_d = f(c,b,\lambda)$$
$$= \begin{Bmatrix} \left[\lambda/c(1-\lambda/c)-b(\lambda/c)^{1+b}/1-(\lambda/c)^b\right]\Big/\lambda b \quad \lambda < c \\ (b+1)/2b\lambda \quad \lambda \to c \end{Bmatrix}$$

This function is simply the average delay divided by the size of the finite buffer. This function has convexity properties. Therefore, an agent that prefers to minimize the normalized average delay, would prefer more buffers and bandwidth from the packet switch supplier.

THEOREM. 1: The utility function (*) (normalized average delay) for an M/M/1/B system is decreasing convex in c for $c \in [0,C]$, and decreasing convex in b for all $b \in [0,B]$.

Proof. Using standard techniques of differentiation one can show very easily that U' is positive.

$$U' = (c/\lambda)^b \lambda \log(c/\lambda)\Big/c\left[-1+(c/\lambda)^b\right]^2$$

and

$$\lim_{c\to\lambda} U' = -1/2b^2\lambda.$$

The second derivative is also positive:

$$U'' = \left(1+(c/\lambda)^b\right)(c/\lambda)^b \lambda \log(c/\lambda)^2\Big/c\left[1+(c/\lambda)^b\right]^3$$

and

$$\lim_{c\to\lambda} U' = 1/b^3\lambda.$$

Consider that agents use such utility functions to obtain the required bandwidth and buffers for average delay requirements. Then competition exists among agents to buy resources Due to convexity properties, the following theorem is stated:

THEOREM 2: Consider *K* agents competing for resources at a switch with finite buffer and finite bandwidth (link capacity) *C*. If the *K* agents have a utility function as shown in (*), then both Pareto optimal allocation s and equilibrium prices exist.

Proof. An intuitive proof can be based on the fact that the traffic classes have, by assumption, smooth convex prefereences in $c_k, (\forall c_k \in [0,C])$ and $b_k\,(\forall b_k \in [0,B])$, and that the utility functions are decreasing convex in the allocation variables. The prices can be normalized such that $p_c + p_b = 1$. By normalizing the prices, the budget set *B*(*p*) does not change, therefore the demand function of the traffic classes (utility under the demand set Φ(*p*) is homogeneeous of degree zero in the prices. It is also well known that if the user (traffic class) has strictly convex preferences, then their demand functions will be well defined and continuous. Therefore, the aggregate demand function will be continuous, and under the resource constraints, the excess demand functions (which is simply the sum of the demands by the K traffic classes at each link minus the resource constraints at each link) will also be continuous.

The equilibrium point is defined as the point where the excess demand function is zero. Then using fixed point theorems (Brouwer's fixed point theorem), the existence of the equilibrium price for a given demand can be shown. Different sets of wealth inputs of the traffic classes will have different Pareto allocations and price equilibria.

If the user preferences are convex and smooth, then under the resource constraints, a Pareto surface exists. This can also be shown using fixed-point theorems in an exchange economy type model, where each user (traffic class) is given an initial amount of resources. Each user then trades resources in the direction of increasing preference (or increasing benefit) until a point where no more exchanges can occur and the allocation is Pareto optimal. The proof is the same when using the unit price simplex property $p_c + p_b = 1$.

An agent can use a utility function which is a combination of the packet loss probability and normalized average delay function.

3.7. Tail Probability Requirements: Utility Functions

Here we assume that agents representing traffic classes have tail probability requirements. This is similar to loss probability. Some applications prefer to drop packets if they spend too much time in the network buffers. More formally, if a packet exceeds its deadline in a certain buffer, then it is dropped. Another way to formalize this is: if the number of packets in a buffer exceed a certain threshold, then the new incoming packets are dropped. The main goal of the network supplier is to minimize the probability that the number of packets in a buffer cross a threshold. In queueing terminology, if the packet tail probability exceeds a certain threshold, then packets are dropped. The problem for the agent is to minimize packet tail probability. The agents compete for resources in order to reduce the tail probability. First we discuss tail probability for the M/M/1 model, and then we consider agents which represent traffic classes with on-off models. Which are of particular relevance to ATM networks. We assume all the traffic classes have the same requirement of minimizing tail probability which implies competing for resources from the supplier.

3.7.1. Tail Probability with M/M/1 Model

Consider agents representing traffic classes with tail probability requirements, and consider an infinite buffer M/M/1 model, where the main goal is to minimize the tail probability of the queueing model beyond a certain threshold. Formally,

$$\text{Tail.Prob.} = P(X > b) = (\lambda/c)^{b+1} \tag{3}$$

The system assumes that $\lambda < c$. From the above equation, the tail probability is decreasing convex with respect to c as long as $\lambda < c$, and is decreasing convex with respect to b as long as $\lambda < b$.

Consider agents using such a utility function for obtaining buffer and bandwidth resources, then using the convexity property and the regions of convexity being ($\lambda < c$). Using the equilibrium condition, as derived in Gottinger [25], we obtain for Pareto optimal allocation and price equilibrium:

$$\begin{aligned} p_c/p_b &= (b_1+1)/c_1 \log(\lambda_1/c_1) \\ &= (b_2+1)/c_2 \log(\lambda_2/c_2) \\ &= \cdots \\ &= (b_n+1)/c_n \log(\lambda_n/c_n) \end{aligned} \tag{4}$$

We assume K agents competing for buffer and bandwidth resources, with tail probability requirements as shown in (3). For the case of two agents in competition, the equilibrium condition is as follows:

$$\begin{aligned} &\log\rho_1/\log\rho_2 = \left[(b_1+1)/c_1\right]\left[c_2/(b_2+1)\right] \\ &\text{with } \rho = \lambda/c \end{aligned} \tag{5}$$

For equilibrium in network economies we can interpret (5) as the ratio of the logs of the utilizations of the classes is proportional to the ratio of the time spent in clearing the buffer contents.

3.7.2. Tail Probability with On-Off Models

In standard performance models the utility functions are derived using simple traffic models such as Poisson, with the mean arrival rate as the main parameter. Here we use on-off (bursty) traffic models in relation to the competitive economic model. The traffic parameters are mean and variance in arrival rate. We show how the traffic variability has an impact on the resource allocation, and in general the Pareto surface at a link. We assume an ATM type network where packet sizes are of fixed size (53 bytes).

On-off models are commonly used as traffic models in ATM networks (Kleinrock [23]). These traffic sources transmit ATM cells at a constant rate when active and nothing when inactive. The traffic parameters are average burst length, average rate, peak rate, and variances in burst length. The traffic models for such sources are on-off Markov sources (Ross [26]). A source in a time slot (assuming a discrete time model) is either "off" or "on". In the on state it transmits one cell and in the off state it does not transmit any cell. When several such (homogeneous or heterogeneous) sources feed into an infinite buffer queue, the tail distribution of the queue is given by the following formula:

$$\Pr(X > b) = h\left(c,b,\rho,C_v^2\right) g\left(c,b,\rho,C_v^2\right)^{-b}$$

where $h\left(c,b,\rho,C_v^2\right)$ and $g\left(c,b,\rho,C_v^2\right)$ are functions of traffic parameters and link capacity c.

Such functions are strictly convex functions in c and b. These functions are currently good approximations to packet loss probability in finite buffer systems, where packet sizes are of fixed size. These approximations become very close to the actual cell (packet) loss for very large buffers. The utility function is as follows: A TC consists of S identical (homogeneous) on-off sources which are multiplexed to a buffer. Each source has the following traffic parameters: $\left\{T,r_p,\rho,C_v^2\right)$ where T is the average on period, r_p is the peak rate of the source, C_v^2 is the squared coefficient of variation of the on period, and ρ is the mean rate. The conditions for a queue to form are: $S\ r_p > c$ (peak rate of the TC is greater than the link capacity) and $S\ r_p\ \rho < c$ (mean rate less than link capacity).

The packet tail distribution of the queue when sources are multiplexed into an infinite buffer queue then has the form

$$U = Sr_p\rho/c\left[1+2\left(c-Sr_p\rho\right)\Big/Sr_p\rho\left(1-\rho\right)^2\left(C_v^2+1\right)T\right]^{-b} \tag{6}$$

Using a numerical example, we use two traffic classes (with the same values). There are $S_1 = S_2 = 10$ sessions in each traffic class, $T = 5, r_p = 1, \rho = 0.5$. Using the constraints $c_1 + c_2 = 60$ and $b_1 + b_2 = 100$, the Pareto surface is obtained. As C_v^2 increases from 1 to 20, the Pareto surface tends to show that buffer space and link capacity are becoming more and more valuable. The equilibrium price ratios

$$p\left(c\right)/p\left(b\right)\text{vs. } C_v^2$$

increase as C_v^2 increases. A higher C_v^2 implies a higher cell loss probability and therefore more resources are required, therefore a higher price ratio (link capacity is more valuable compared to buffer).

4. Specific Cases

Now we consider some specific cases of agents with different QoS requirements.

4.1. Loss and Average Delay

Consider the following case, where two agents have different QoS requirements, one of them (agent 1) has a packet loss probability requirement and the other (agent 2) has an average delay requirement. We assume that the network supplier has finite resources, C for link bandwidth and B for buffer. Using the properties of loss probability and average delay with respect to bandwidth and buffer, the Pareto optimal solution is simply: all buffer to agent 1, as agent 2 does not compete for link buffer. The competition is for link bandwidth between agent 2 and agent 1. Let w_1 be the wealth of agent 1, annd w_2 for agent 2, then the equilibrium prices of buffer and bandwidth are the following:

$$p_b = p_b^f \text{and}\, p_c = \left(w_1 + w_2\right)/C$$

Since there is no competition for buffer space, the cost of the buffer is simply the fixed cost p_b^f. The Pareto allocations are

$$\left\{B, Cw_1/\left(w_1 + w_2\right)\right\}$$

for agent 1 and

$$\left\{0, Cw_2/\left(w_1 + w_2\right)\right\}$$

for agent 2.

4.2. Loss And Normalized Average Delay

Consider the following case, where agent 1 and agent 2 have preferences on loss probability and normalized average delay requirements (transforming average delay requirements into normalized average delay requirements). In this case, the two agents have different utility functions, however, their preferences are such that more buffer and more bandwidth is required and this causes the agents to compete for both resources.

The utility function for agent 1 is as follows:

$$U_1 = \gamma_1 U_{\text{loss}} + \left(1-\gamma_1\right)U_{\text{delay}}, \text{where } \gamma_1 \in \left[0,1\right]$$

The utility function for agent 2 is as follows:

$$U_2 = \gamma_2 U_{\text{loss}} + \left(1-\gamma_2\right)U_{\text{delay}}, \text{where } \gamma_2 \in \left[0,1\right]$$

For example, agent 1 might prefer more weight on loss probability than normalized average delay compared to agent 2 who weighs normalized average delay more than loss probability. Let agent 1 choose $\gamma_1 = 0.9$, and agent 2 choose $\gamma_2 = 0.1$. Due to the convexity properties of the loss probability function and the normalized average delay function, the resultant multi-objective utility function is decreasing convex with respect to bandwidth and buffer, respectively. Under the equilibrium condition, the equilibrium prices for the resources have the familiar prroperty that the ratio of prices is equal to the ratio of marginal utilities with respect to the resources, for each agent.

Using the resource constraints

$$c_1 + c_2 = C$$

and

$$b_1 + b_2 = B,$$

we can obtain the Pareto surface. To compute a specific Pareto allocation one uses the following parameters: agent 1 and agent 2 have the same traffic arrival rate $\lambda_1 = \lambda_2 = 10$. The performance model is the M/M/1/B model for both agents. Using the atonement process, where agents negotiate with the link supplier to buy bandwidth and buffer resources, the process converges to a price equilibrium. The Pareto optimal allocation is split evenly with respect to buffer and bandwidth among the agents. The price of link bandwidth is higher than the price of buffer.

5. Service Economy: Architecture for Interaction

Consider a large scale distributed information system with many consumers and suppliers. Suppliers are content providers such as web servers, digital library servers, multimedia database and transaction servers. Consumers request for and access information objects from the various suppliers and pay a certain fee or no fee at all for the services rendered.

Consider that third party suppliers provide information about suppliers to consumers in order to let consumers find and choose the right set of suppliers.

Access and dissemination: consumers query third-party providers for information about the suppliers, such as services offered and the cost (price). Likewise, suppliers advertise their services and the costs via the third party providers in order to attract consumers. Consumers prefer an easy and simple way to query for supplier information, and suppliers prefer to advertise information securely and quickly across many regions or domains. For example, consider a user who wishes to view a multimedia object (such as a video movie). The user would like to know about the suppliers of this object, and the cost of retrieval of this object from each supplier.

Performance requirements: users wish to have good response time for their search results once the queries are submitted. However, there is a tradeoff. For more information about services offered, advanced searching mechanisms are needed, but at the cost of increased response time. In other words, users could have preferences over quality of search information and response time. For example, users might want to know the service costs in order to view a specific information object. In large networks, there could be many suppliers of this object, and users may not want to wait forever to know about all the suppliers and their prices. Instead, they would prefer to get as much information as possible within a certain period of time (response time).

From the above example, in order to let many consumers find suppliers, a scalable decentralized architecture is needed for information storage, access and updates.

Naming of services and service attributes of suppliers becomes a challenging issue when hundreds of suppliers spread across the globe. A simple naming scheme to connect consumers, across the internet, with information about suppliers is essential. The naming scheme must be extensible for new suppliers who come into existence. A name registration mechanism for new suppliers and a de-registration mechanism (automatic) to remove non-existent suppliers is required. In addition, naming must be hierarchical, domain based (physical or spatial domains) for scalability and uniqueness. Inter-operability with respect to naming across domains is an additional challenging issue not covered in this paper.

The format of information storage must be simple enough to handle many consumer requests quickly within and across physical domains. For better functionality and more information, a complex format of information storage is necessary, but at the cost of reduced performance. For example, a consumer, in addition to current service cost, might want to know more information such as the cost of the same service during peak and off-peak hours, the history of a supplier, its services, and its reputation, in order to make a decision. This information has to be gathered when requested. In addition, the storage formats must be inter-operable across domains.

Performance: a good response time is important to make sure consumers get the information they demand about suppliers within a reasonable time period, so that decision-making by consumers is done in a timely fashion. In addition, the design of the right architectures for information storage and dissemination is necessary for a large scale market economy to function efficiently. Using the previous example, consumers and suppliers would prefer an efficient architecture to query for and post information. Consumers would prefer good response time in obtaining the information, and suppliers prefer a secure and fast update mechanism to provide up-to-date information about their services.

Security in transferring information and updating information at the bulletin boards (name servers) is crucial for efficient market operation and smooth interaction between consumers and suppliers. For this the third party suppliers (naming services) have to provide authentication and authorization services to make sure honest suppliers are the ones updating information about their services.

6. Conclusions

We show some applications of mathematical economics and operations research to resource management problems in distributed systems and computer networks. These concepts are used to develop effective market based control mechanisms, and to show that the allocation of resources are Pareto optimal.

We propose novel methodologies of decentralized control of resources, and pricing of resources based on varying, increasingly complex QoS demands of users. We bring together economic models and performance models of computer systems into one framework to solve problems of resource allocation and efficient QoS provisioning matching large-scale e-commerce applications. The work can be applied to pricing services in ATM networks and (wireless) Integrated Services Internet of the future. We address some of the drawbacks to this form of modelling 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 consumers between suppliers. 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. As in real economies, the resources in a computer system may not easily be 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.

In addition to models for QoS and pricing in computer networks, we are also working towards designing and building distributed systems using market based mechanisms to provide QoS charge users either in a commercial environment or in a private controlled environment by allocating quotas via fictitious money (charging and accounting) by central administrators.

REFERENCES

[1] R. Radner, "The Organization of Decentralized Information Processing," *Econometrica*, Vol. 61, No. 5, 1993, pp. 1109-1146. doi:10.2307/2951495

[2] T. Van Zandt, "The Scheduling and Organization of Periodic Associative Computation: Efficient Networks," *Review of Economic Design,* Vol. 3, No. 2, 1998, pp. 93-127. doi:10.1007/s100580050007

[3] K. R. Mount and S. Reiter, "Computation and Complexity in Economic Behavior and Organizatiion," Cambridge University Press, Cambridge, 2002. doi:10.1017/CBO9780511754241

[4] X. Deng and F. C. Graham, "Internet and Network Economics," 3rd International Workshop, WINE 2007, Springer, San Diego, Berlin, New York, 2007.

[5] D. Neumann, M. Baker, J. Altmann, O. Rana, "Economic Models and Algorithms for Distributed Systems," Birkhaeuser, Basel, 2010. doi:10.1007/978-3-7643-8899-7

[6] H. W. Gottinger, "Economies of Network Industries," Routledge, London, 2003. doi:10.4324/9780203417997

[7] S. Shenker, "Service Models and Pricing Policies for an Integrated Services Internet," In: B. Kahin and J. Keller, Eds., *Public Access to the Internet*, MIT Press, Cambridge, 1995, pp. 315-337.

[8] D. F. Ferguson, C. Nikolaou, J. Sairamesh and Y. Yemini, "Economic Models for Allocating Resesources in Computer Systems," In: S. Clearwater, Ed., *Market-Based Control*: *A Paradigm for Distributed Resource Allocation*, World Scientific, Singapore City, 1995. http://brahms.di.uminho.pt/discip/MInf/ac0203/ICCA03/EconModAlloc.pdf

[9] M. Macias, G. Smith, O. Rana, J. Guitart and J. Torres, "Enforcing Service Level Agreements Using an Economically Enhanced Resource Manager," In: D. Neumann, M. Baker, J. Altmann and O. Rana, Eds., *Economic Models and Algorithms for Distributed Systems*, Birkhäuser, Basel, 2010, pp. 109-125.

[10] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," *Games and Economic Behavior*, Vol. 35, No. 1-2, 2001, pp. 166-196. doi:0.1006/game.1999.0790

[11] N. Nisan, T. Roughgarden, E. Tardos and V. V. Vazirani, "Algorithmic Game Theory," Cambridge University Press, Cambridge, 2007.

[12] A. B. Mohammed, J. Altmann and J. Hwang, "Cloud Computing Value Chains: Understanding Business and Value Creation in the Cloud," In: D. Neumann, M. Baker, J. Altmann and O. Rana, Eds., *Economic Models and Algorithms for Distributed Systems*, Birkhäuser, Basel, 2010, pp. 187-208.

[13] J. Feigenbaum, M. Schapiro and S. Shenker, "Distributed Algorithmic Mechanism Design," In: N. Nisan, T. Roughgarden, E. Tardos and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, Cambridge, 2007, pp. 363-384.

[14] R. B. Myerson, "Fundamental Theory of Institutions: A Lecture in Honor of Leo Hurwicz," 2006. http://home.uchicago.edu/~rmyerson/hurwicz.pdf

[15] R. Wilson, "Nonlinear Pricing," Oxford University Press, Oxford, 1993.

[16] J. K. MacKie-Mason and H. R. Varian, "Pricing the Internet," In: B. Kahin and J. Keller, Eds., *Public Access to the Internet*, MIT Press, Cambridge, 1995, pp. 269-314.

[17] H. W. Gottinger, "Telecommunication, Internet, Regulation and Pricing," In: M. Takashima, H. W. Gottinger and C. L. Umali, Eds., *Economics of Global Telecommunications and the Internet*, Nagasaki University, 1997, pp. 107-127.

[18] R. Srikant, "The Mathematics of Internet Congestion Control," Birkhaeuser, Basel, 2004. doi:10.1007/978-0-8176-8216-3

[19] S. Low and P. Varaiya, "A New Approach to Service Provisioning in ATM Networks," *IEEE Transactions on Networking*, Vol. 1, No. 5, 1993, pp. 547-553.

[20] R. Vohra, "Mechanism Design: A Linear Programming Approach," Cambridge University Press, Cambridge, 2003.

[21] M. Weinard, "Deciding the FIFO Stability of Networks in Polynomial Time," In: T. Calamoneri, I. Finochi and G. F.

Italiano, Eds., *Algorithms and Complexity*, Springer, Berlin, 2006, pp. 81-91.

[22] H. Scarf, "Computation of Economic Equilibria," Yale University Press, New Haven, 1973.

[23] L. Kleinrock, "Queueing Networks, Vol. II," Norton, New York, 1996.

[24] L. Kleinrock and R. Gail, "Queueing Systems: Problems and Solutions," Wiley, New York, 1996.

[25] H. W. Gottinger, "Strategic Economics for Network Industries," NovaScience, New York, 2010.

[26] S. Ross, "Applied Probability Models with Optimization Applications," Dover, New York, 1970.