iBusiness, 2011, 3, 313-322
doi:10.4236/ib.2011.34042 Published Online December 2011 (http://www.SciRP.org/journal/ib)

Scientific Research

Network Economies for the Internet-Application Models

Hans Gottinger

STRATEC, Munich, Germany.
Email: info@stratec-con.com

Received August 9th, 2011; revised September 16th, 2011; accepted October 9th, 2011.

ABSTRACT

We propose a decentralized model of network and server economies, where we show efficient QoS (*Quality of Service*) *provisioning and Pareto allocation of resources* (*network and server resources*) *among agents and suppliers, which are either network routers or servers* (*content providers*)*. Specifically, it is shown* 1) *how prices for resources are set at the suppliers based on the QoS demands from the agents and* 2) *how dynamic routing algorithms and admission control mechanisms based on QoS preferences emerge from the user classes for the network economy.*

Keywords*: *Internet Economics*, *Distributed Systems*, *Mechanism Design*, *Optimization*, *Network Economy

1. Introduction

On the basis of the conceptual framework of network economics, Gottinger [1], we will motivate and solve two problems of allocating resources and providing services (QoS) to several classes of users in view of network links and the collection of servers. In the first case we address the informational links in its supply/demand infrastructure, in the second case we focus on the transaction based aspect of the internet, recently identified with e-commerce on the business-to-consumer as well as business-to-business dimension. For both we start with some stylized examples that reflect the present internet structure.

We first consider a network economy, of many parallel routes or links, where several agents (representing user classes) compete for resources from several suppliers, where each supplier represents a route (or a path) between source and destination. Agents buy resources from suppliers based on the QoS of the class they represent. Suppliers price resources , independently, based on demand from the agents. The suppliers connect consumers to information providers who are at the destination, the flow of information is from information providers to consumers. We formulate and solve problems of resource allocation and pricing in such an environment. We then consider a server economy in a distributed system. Again we use a similar model of interaction between agents and suppliers (servers). The servers sell computational resources such as processing rate and memory to the agents for a price. The prices of resources are set independently by each server based on QoS demands from the agents. Agents represent user classes such as transactions in database servers or sessions for Web servers that have QoS requirements such as response time. Resource allocation in networks relate to computational models of networks, as developed in the works of Radner [2] ,Mount and Reiter [3], Mount and Reiter [4, Chap.4], van Zandt [5], see also Gottinger [1, Chap.9]. Here they emanate from certain types of queuing systems, Kleinrock [6], Wolff [7], on generalized networks.

The paper is organized as follows. Sections 1.1 - 1.3 briefly review issues of internet resource allocation, QoS and pricing. In Section 2 we present two examples , one of simple network routing (2.1), the other on network transactions that provide similar platforms of network allocation decisions (2.2). Based on a simple decentralized model for the network economy we apply the principles of economic optimization between agents and suppliers in Section 3. Section 3.1 outlines a structural model of the network economy with Pareto Optimality and price equilibrium for agents competing for resources from suppliers emerging. We present a routing algorithm which considers the dynamic nature of session arrival and departure. Some results for optimal allocation and for the routing mechanism are presented. In Section 3.2 we present the server economy, and show the Pareto optimal allocations and price equilibrium when agents are competing for resources from servers (suppliers). Correspondingly, we apply transaction routing policies to han-

dle the dynamics of user behaviour. Conclusions follow in Section 4.

1.1. Internet Resources

The evolution of Internet pricing poses interesting resource allocation problems. Flat-rate pricing has been a key condition that allowed the Internet to expand very fast. It is interesting to note that in the first expansion wave of the Internet flat rate pricing has been more prevalent in the US than in Europe which partially explains higher user diffusion rates in the US than in Europe or Japan among private users.

But as the net has grown in size and complexity, not discounting engineering advances in network (management) technologies, it is now becoming more obvious that other pricing schemes being able to cope with severe congestion and deadlock should come forward. New pricing schemes should not only be able to cope with a growing Internet traffic but also be able to foster application development and deployment vital to service providers. Usage-based pricing of this new kind should make the internet attractive to many new users. Casual users will find it more affordable, while business users will find a more stable environment.

Without an incentive to economize on usage, congestion can become quite serious. The problem is more serious for data networks like the Internet than for other congestible transportation network resources because of the tremendously wide range of usage rates. A single user at a modern workstation can send a few bytes of email or put a load of hundreds Mbps on the network, for example, by downloading videos demanding more than 1 Mbps.

A natural response by shifting resources to expand technology will be expensive and not a satisfactory solution in the longer run. Many proposals rely on voluntary efforts to control congestion. Many participants in congestion discussions suggest that peer pressure and user ethics will be sufficient to control congestion costs. But as MacKie-Mason and Varian [8] suggest we essentially have to deal with the problem of overgrazing the commons, e.g. by overusing a generally accessible communication network. A few proposals would require users to indicate the priority they want each of the sessions to receive, and for routers to be programmed to maintain multiple queues for each priority class. The success of such schemes would depend on the users' discipline to stick to the assigning of appropriate priorities to some of their traffic. However, there are no effective sanction and incentive schemes that would control such traffic, and therefore such a scheme is liable to be ineffective. This is why pricing schemes have gained increasing attention and various approaches and models have been discussed in the network community.

1.2. 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 [9]) 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 can tolerate significant distortion. Real-time video broadcasts have very low tolerance for delay and distortion. 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.

Network pricing could be looked at as a mechanism design problem (Hurwicz and Reiter, [10]). 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.

1.3. Pricing Congestion

The social cost of congestion is a result of the exis-tence 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, [11, Chap. 10]). 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 [8] was 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. Two Examples

2.1. Network Routing

The first example shows a network representing many user classes or types of users wishing to access specific content providers. The users have a choice of routes to connect to the servers. Several routes exist between the source and the destination. At the destination there are various kinds of content providers, such as databases, digital libraries and web servers. Each route is independent (parallel) and they have different amounts of resources. The resources are buffer and bandwidth. For simplicity, we assume that each route is a single link between a source and a destination, and we assume that each route has one packet switch that buffers packets and transmits them. User classes have several QoS requirements such as packet loss utility, maximum end-to-end delay and average packet delay. The QoS requirements are due to applications such as digital video libraries, access to multimedia databases and web servers. Sessions are set up between the source and the destination along one of the routes to access the content providers. The applications, for smooth operation, demand a certain QoS from the network routes and the end-nodes (content providers), for an end-to-end QoS. For example, video applications generate bursty traffic, and this can lead to packet loss in the network depending on the allocation of network resources for the video sessions. Video applications can tolerate a certain amount of packet loss, but beyond a threshold, the QoS of the video at the user workstation will deteriorate. In addition, maximum delay requirement is necessary to design buffer play-out strategies for smooth operation of the video application at the user workstation.

Let b(s) be the burstiness curve of input m(t), the source traffic rate, at fixed service rate s. Under normal circumstances b(s) is assumed to be nonnegative, convex and strictly decreasing for s smaller than the peak rate of traffic. The burstiness curve then represents the buffer size necessary to avoid cell losses at each service rate s. When a bandwidth-buffer space pair (s,b(s)) on the burstiness curve is used for resource allocation, there will be no cell loss.

From the demand side, the demand at the network changes due to random arrivals and departures of user sessions of the traffic classes. The new session arrival may require the user class to acquire more resources to ensure a certain QoS level. In addition, when a new session arrives, a decision has to be made as to which route to choose from. In the example, resources are finite, and therefore have to be used efficiently in order to provide QoS. The traffic classes are allocated resources only for a certain period of time. The main reason being that session arrival and departure rates could change, causing fluctuations in demand, and therefore, resources have to be re-allocated to meet the change in demand. The traffic classes can re-negotiate for resources once their ownership of resources expires.

From the supply side, consider that each route is a supplier, and let each traffic class be represented by an agent. The agent on behalf of the class negotiates for resources from the suppliers based on QoS requirements of the class.

Each supplier has to guarantee buffer and bandwidth resources, depending on the demand from the agents. The supplier has to ensure efficient utilization of the network resources, so that the resource limits are fully exploited given the QoS requirements of classes. The task of the agents is to represent the QoS needs of the traffic class, given a certain performance framework from the supplier. Every time a session of a certain traffic class arrives, a decision must be made on which route to take between the source and destination. This depends on the agent, who can choose a route based on preferences of the traffic class, and the available resources in the routes. Therefore, dynamic mechanisms are necessary to ensure the right decision making in routing a newly arrived session. In a dynamic network the available resources at each route could be different, and in addition there is competition from other agents who have similar tasks to perform. With many routes between source and destination, the routing or placing of sessions along a route or a link must be done in a decentralized fashion. This is necessary to handle many routes and many traffic classes, each of which could have diverse QoS requirements. A framework to decentralize the various functions or tasks in admitting and routing sessions, and scheduling to switch bandwidth and buffer among the traffic classes is a challenging problem. In addition, the framework must ensure flexible QoS provisioning and promote efficient utilization of resources.

2.2. Transaction Processing

In this example, users request services from the content providers, and users are grouped into classes. The user classes are transactions classes for databases or just sessions for computation or information retrieval, which request for access services from one or more of the servers (content providers).

Consider a transaction processing system, where transactions that arrive are routed to one of many systems in order to satisfy performance objectives such as average response time or per-transaction deadlines. In commercial online transaction processing systems, it is very common for transactions to be processed on heterogeneous servers which have different operating systems, database management systems, hardware and software platforms, and a host of various communication protocols. Transactions are grouped into transaction classes, transactions in the same class have common workload characteristics and performance objectives. Transactions arrive at random times to their respective classes and therefore need to be routed dynamically to one of the servers. Each transaction class could have different preferences over the performance objectives and they have different processing requirements from the servers.

In a transaction processing system it is quite difficult to match the quantities of resources for an efficient usage with the diverse QoS requirements of user classes. For example, a queue could be assigned to each class at each server in order to provide various service levels, or a queue at each server could be shared among the classes. For a queue that is shared by many classes the complexity of service provisioning increases as transactions from each class have to be distinguished in order to provide service levels. The allocation mechanism determines the throughput of each queue and the buffer allocation at the server. In addition, efficiency could mean a server wide performance measure of session level throughput, given the QoS requirements of the transaction classes.

In order to handle many transaction classes and provide access to various services, the control of resources must be decentralized for reasons of efficiency and transparency. Each server (supplier) has to offer resources such as processing, memory and input/output, and services such as average response time and throughput. This cannot be done in a centralized fashion, if we consider all the servers, instead decentralized mechanisms are needed to distribute user sessions (transaction sessions) among the servers and provide the QoS needs of classes. In addition, each server has to implement practical mechanisms, such as processor scheduling, to partition resources among the various transaction classes or provide priority services among the classes.

In this example, when a user session arrives, the problem of choosing a server in order to provide a service is needed. Consider each class is represented by an agent. If a new session arrives the agent has to know if there are enough available resources to provide the required QoS. Consider that agents represent transaction classes, and they compete for resources from the various databases, and remove this burden from the servers. The problem for the agents are to choose the right server and to make sure QoS is guaranteed to the class it represents, and use the allocated resources judiciously. This implies mechanisms for optimal routing need to be designed.

With random arrival and departure of user sessions, the agent must handle routing and admission of sessions in a dynamic way. The problem of efficient resource management by the agents and optimal allocation of resources by the servers, due to changing demand is challenging. The allocation of resources cannot be static and time-periods of renegotiation of resources and services will affect the way routing and admission of sessions is done. In addition, servers will have to adapt to changing demand in order to reflect the new allocations. For example, consider that depending on the time of day, demand at the servers fluctuate, and demands could be independent from server to server. The challenge is in determining the time-intervals based on demand.

3. A Model of the Network and Server Economy

3.1. The Network Economy

The network consists of V nodes (packet switches) and N links. Each node has several output links with an output buffer. The resources at output link are transmission capacity (or link capacity) and buffer space. The link controller at the output link schedules packets from the buffer. This is based on how the buffer is partitioned among the traffic classes and the scheduling rule between the traffic classes. Sessions are grouped into traffic classes based on similar traffic characteristics and common QoS requirements. Sessions that belong to a class share buffer and link resources, and traffic classes compete for resources at a packet switch. Each session arrives to the network with a vector of traffic parameters Tr, vector of QoS requirements and wealth. A session is grouped or mapped to a corresponding traffic class. A traffic class has common QoS requirements, and we consider QoS requirements per traffic class rather than per session. Once a session is admitted along a path (a route), it will continue along that path until it compeletes.

Each agent k performs the following to obtain the demand set on each link. The allocations are buffer (b) and bandwidth (c) on each link for each agent. The wealth is distributed across the links by each agent to buy re-

sources.

That is, the problem is to find pairs $\{c^*_k, b^*_k\}$ such that max $U_k = f(c_k,b_k,Tr_k)$, constraints $p_b b_k + p_c c_k = w_k$.

In the above formulation, each agent k buys resources from each link. The allocation for agent k is $c^*_k = \{c_k^{*1}, c_k^{*2}, \cdots, c_k^{*N}\}$ and $b^*_k = \{b_k^{*1}, b_k^{*2}, \cdots, b_k^{*N}\}$. An agent can invest wealth in either some or all the links. We assume that at each link there is competition among at least some of the agents for buying resources. As previously, Gottinger [1], we show a general utility function which is a function of the switch resources: buffer (b) and bandwidth (c). A utility function of the agent could be a function of:

- Packet loss probability $U_t = g(c,b,Tr)$
- Average packet delay $U_d = h(c,b,Tr)$
- Packet tail probability $U_l = v(c,b,Tr)$
- Max packet delay $U_b = f(b)$
- Throughput $U_c = g(c)$

We consider that an agent will place demands for resources based on a general utility function, which is a combination of the various QoS requirements:

$$U = f(c,b,Tr) = x_l U_l + x_d U_d + x_b U_b + x_c U_c + x_t U_t$$

where U_l is the packet loss probability utility function, U_d is the average delay utility function, U_t is the packet tail probability, U_b is the utility function for max-delay requirements, and U_c is for bandwidth (throughput) requirements. x_l, x_d, x_b, x_c, x_t are constants. Agents could use such a utility function. As long as the convexity property with respect to buffer b and bandwidth c holds. Pareto optimal allocations and price equilibria exist. However, if they are not convex, then depending on the properties of the functions, local optimality and price equilibrium could exist. To show the main ideas for routing and admission control, we use packet loss probability as the main utility function (U_l), which means we assume that x_l from the above equation are the only constant and the rest are zeros. For doing this, we need first some further specifications of the loss probability. We later show results for Pareto optimality and price equilibrium, and then we propose routing and admission control algorithms. In general, one can assume that agents, on behalf of user classes, demand for resources from the link suppliers based on the utility function shown above. The agent uses the utility function to present the demand for resources over the whole network of parallel links.

Loss Probability Specifications. At each output link j the resources are buffer space B^j and link capacity C^j.

Let $\{c^j_k, b^j_k\}$ be the link capacity and buffer allocation to class k on link j where k ε [1,K]. Let p^j_c and p^j_b be the price per unit link capacity and unit buffer respectively at link j, and w_k be the wealth (budget) of a traffic class k. For a link j from the source to the destination, the packet loss probability (utility) for traffic class k is given by the following

$$U_{lk} = P_{loss} = 1 - \Pi_{j=1}^{N} (1 - P_k^j) \tag{3.1}$$

where P_k^j is the packet loss probability at link j of agent k.

The goal of the agent is to minimize the packet loss probability under its wealth or budget constraints. If the traffic classes have smooth convex preferences[1] with respect to link capacity and buffer allocation variables at each link, then the utility function U_{lk} is convex with respect to the variables.

3.1.1. Price Equilibrium

Let each TC (represented by an agent) 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 (packet loss probability for M/M/1/B queues as in Wolff [1989]) U for each TC at each link is given by

$$U = f(c,b,\lambda) = \begin{cases} \dfrac{\left(1-\frac{\lambda}{c}\right)\left(\frac{\lambda}{c}\right)^b}{1-\left(\frac{\lambda}{c}\right)^{+1+b}} \\ \dfrac{1}{b+1} \\ \dfrac{(-1)+\left(\frac{\lambda}{c}\right)\left(\frac{\lambda}{c}\right)^b}{-1+\left(\frac{\lambda}{c}\right)^{1+b}} \end{cases}$$

$$\text{for} \quad 1 < c, 1 = c, 1 > c \text{ resp.} \tag{3.2}$$

The allocation variables at each node for each traffic class are c (link capacity) and b (buffer space). The utility function is continuous and differentiable for all c ∈ [0,C], and for all b ∈ [0,B]. We assume that b ∈ ℜ for continuity purposes of the utility function.

With agents competing for resources in a network of parallel links, the overall utility function U can be obtained by using the utility function above. We have the following theorem.

Proposition 3.1 The packet loss probability function for agent k shown in (3.1), assuming an M/M/1/B model for each link, is decreasing convex in c_k^j for $c_k^j \in [0,C_j]$, and decreasing convex in b_k^j, $\forall\ b_k^j \in [0,B_j]$.

Proof. See Appendix.

The goal of each agent is to maximize the preference (which is minimizing packet loss probability) under the budget constraint. Each traffic class computes a demand set using the wealth constraints and the current prices.

[1]Under the assumption that general queueing systems have convex packet loss probability functions (see Harel and Zipkin [12], which means convex preferences in link capacity and buffer space. Once proved, this can be a very useful property in designing resource allocation mechanisms for general network topologies.

The demand set can be computed using Langrange multiplier techniques. Using the utility function given by (3.2) and the first order equilibrium conditions, the price ratio at each link j is given by the following:

$$\frac{\partial U/\partial c_k^j}{\partial U/\partial b_k^j} = \frac{p_c^j}{c_b^j} = \frac{N_k^j - b_k^j}{c_k^j \log \rho_k^j},\ N_k^j = \frac{\rho_k^j \times \left(1 - P_k^j\right)}{1 - \rho_k^j \times \left(1 - P_k^j\right)} \quad (3.3)$$

where function N_k^j is the ratio of the effective queue utilization $\left(\rho_k^j \times \left(1 - P_k^j\right)\right)$ to the effective queue emptiness $1 - \rho_k^j \times \left(1 - P_k^j\right)$ and $\rho_k^j = \lambda_k^j / c_k^j$.

Consider K traffic classes of M/M/1/B type competing for resources (link and buffer) in a network of parallel links. Then the following theorem is stated:

Proposition 3.2 Let each traffic class k have smooth convex preferences represented by the utility function shown in (3.2) Given that $\sum_1^K c_i = C$ and $\sum_1^K b_i = B$ for all i,k $\in$ [1,K], then the Pareto surface exists. Given the wealth constraint w_k of the traffic classes, the Pareto optimal allocation and the price equilibrium exist.

The proof is based on the fact that the utility functions are decreasing convex and smooth in the resource space (preferences are convex and smooth). The proof is essentially the same as in Gottinger [1, Chap.9] for Pareto optimality, except that the preferences are shown here to be convex in link capacity and buffer space at each link (given the traffic parameters of each traffic class at each link) in the network of parallel links using the M/M/1/B model.

3.1.2. Agent Routing and Admission

For a session that arrives to a traffic class in the network, the agent has several routes to choose from between the source and the destination. The agent can choose a route that benefits the traffic class it joins. This means that an agent is searching for the right set of service providers (or links) to reach the destination. Several interesting questions arise in a market economy with many users and suppliers: will the network economy be efficient in service provisioning? What are the negotiation protocols between the users and the suppliers so that services are guaranteed? What is the session blocking probability per class, given session arrival and average session holding time per class?

The static description of the problem is simply to find the best allocation of sessions among the suppliers. The allocation can satisfy efficiency criteria such as throughput of the number of sessions per class admitted to the overall network. For example, consider the static case that A_i is the session arrival rate (Poisson with distribution) for class i, and Ω_i is the average session holding time of sessions of class i. Let Agent be allocated c_{ij} link capacity on link j. Let the maximum number of sessions that can be admitted per link j for class i, such that certain QoS level is satisfied. Let the space be $\{n_{i1}, n_{i2}, \cdots, n_{iN}\}$. Then the problem for the agent is simply to determine the flow of sessions among the network of links, given the above parameters. Formally, the agent has to find the following: find $\{\rho_{i1}, \rho_{i2}, \cdots, \rho_{iN}\}$, minimize $1 - \Pi_1^N (1 - P_{block})$ given that $\{c_{ij}, b_{ij}\}$ and constraints $\sum_1^N \rho_{ij} = \rho_i$ where $\rho_{ij} = A_i/\Omega_i$ for all j $\in$ [1,N] is the session level utilization, and $\sum_1^N A_{ij} = A_i$ and $\rho_i = A_i/\Omega_i$. For the main goal of agent i is to maximize the throughput of the sessions through the network. This is one of the many efficiency requirements of agent i.

We now discuss dynamic routing algorithms for each agent i that routes a session over the network along one of the routes (or links). The dynamic routing algorithms depend on the state of the network portion owned by agent i. For example, the routing decisions will be made based on the number of sessions currently active on each of the links for class i.

Consider a parallel link network as explained in Sect. 2 where each link is a supplier. The routing algorithm is as follows: The agent representing the traffic class will choose the supplier which can give a better QoS for the overall class. This means that the suppliers in the network are ordered in the decreasing order of preference by the agent based on the utility derived by joining them. This routing algorithm is descrobed as follows for a network consisting of several parallel links between a source and a destination. If a session of type TC_k arrives, then it is routed to a link j which gives the maximum preference (maximum QoS) to the class from among the set of suppliers. The routing mechanism yields the guideline: route to j such that max U_k (Φ(p)) for all j. where (Φ(p) is the demand at supplier j. U_k (Φ(p)) is the overall utility derived by traffic class k if the session joins supplier j. This mechanism essentially states that the agent will choose the service provider which gives the maximum utility (or in this case minimal packet loss probability) to the traffic class. The routing algorithm (by the agent) first computes P_k^j (Φ(p)) for all the suppliers, where P_k^j (Φ(p)) is the packet loss probability of traffic class k at link j in the parallel link network. The agent then ranks the class utility derived by joining a supplier, and then ranks them in decreasing order of preference.

Admission Control: The agent will admit the session on one of the many routes (links) provided the QoS of the traffic class it joins is honored. If the agent has the same preference over a subset of the suppliers (a tie for suppliers), then one of them will be chosen at random. If all the sessions of a traffic class are identical (same traf-

fic load and parameters), then the agent can compute the admission space, and the number of sessions that can be admitted without violating the QoS constraints of the class over all the links. Formally, find $\{n_1^*, n_2^*, \cdots, n_N^*\}$ given that $\{c_{ij}^*, b_{ij}^*\}$ with constraints $q_i^c = \{q_{i1}^c, q_{i2}^c, \cdots\}$.

The agent has to find the maximum (n_j^*) of admissible sessions at each link, given the Pareto allocation for agent i on each link j, and given the QoS constraints of the class i (q_i^c) which, for example, could be packet loss probability, max-delay and average delay requirement per class.

Several interesting questions that arise under the class welfare based routing: what is the session level blocking probability per class, given the session arrival rate and average holding time per class? How does it depend on the session arrival rate and holding time? Does this routing algorithm balance the loads in such a fashion that a traffic class benefits in the long run by the routing algorithm?

We study some of the questions numerically. We use simulations where 2 traffic classes (2 agents) compete for resources in a two node (link) parallel network, *i.e.* just two suppliers. The sessions of traffic class k arrive to the network at a Poisson rate of γ_k. The session holding time is exponentially distributed with mean ν_k ($k \in [1,2]$. Each session of class k arriving has average packet arrival rate λ_k (traffic parameters are Poisson arrivals) and the mean service time is exponentially distributed with mean one. The state space is basically a Markov chain with four parameters $\{n_1^1, n_2^1, n_1^2, n_2^2\}$ representing the number of sessions of traffic class k at each of the links. However, for each agent the state space is 2 dimensional. Numerical studies indicate that the routing algorithms are stable. The results can be obtained using simulations and Markov chain models of 2 user classes and 2 suppliers (links). The session blocking probability is in the order of $1/10^7$ for an offered load (at the session or call level) $\gamma_k/\nu_k = 2.0$. It is evident that the dynamic algorithm is better than the static one as we increase $\rho = A/\Omega$. The agent that routes a session can choose one of the links dynamically based on the state of the link. Let class 1 be allocated bandwidth and buffers in such a way that 10 sessions can be admitted to supplier 1 and 15 to supplier 2 for agent 1. This admission region assumes that a packet loss probability of $1/10^8$ is the QoS requirement by class 1.

3.2. The Server Economy

We now discuss the server economy where servers offer processing resources and memory to agents representing user classes. The agents compete for these resources and buying as much as possible from suppliers. The agents perform load balancing based on the OoS preferences of the class it represents.

The economic model consists of the following players: Agents and Server Suppliers, Consumers or user classes and Business. User sessions within a class have common preferences. User classes have QoS preferences over average delay and throughput, and in some cases completion times of sessions (deadlines). Users within a class share resources at the servers.

Agents and Network Suppliers: Agents represent user classes. An agent represents a single user class. Agents negotiate with the supplier and buy resources from service providers. Agents on behalf of user classes demand for resources to meet the QoS needs. Suppliers compete to maximize revenue. Suppliers partition and allocate resources (processing rate and memory) to the competing agents.

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

The agent and network supplier become service providers in the market. The role of the supplier is to provide technologies to sell resources (buffer and bandwidth units) and to partitioning them flexibly based on the demand by the agents. The agents transform the goods (buffer and bandwidth) and provide QoS levels to the user-classes. The agents strive to maximize profits (minimize buying costs) by using the right utility functions and the right performance models in order to provide QoS to the user-class. More users within a user-class implies more revenue for the agent. The agent is decoupled from the traffic class and the supplier.

In this economy, user classes are transaction classes that send transactions to database servers for processing. The transaction processing time at each of the server is based on the type of transaction. Consider K classes of transactions and each class is represented by an agent (economic agent). In the economy, the agents negotiate with the servers for server capacity. We assume that transactions of any class can run on any of the database servers. Therefore, agents negotiate with all the servers for server throughput (or processing speed). A model where K agents compete for services in a transaction processing system, each class could do the following based on its preferences on average delay and throughput:

1) each agent i can minimize its average response time under throughput constraints, 2) each agent i can maximize throughput of its transactions under an average delay constraint, 3) each agent i can look at a combination of QoS requirements and have preferences over them.

Therefore, each class can choose either one of these preferences and let the agent control the flow of transacttions through the system. The problem now becomes a multi-objective optimization problem as every agent is trying to maximize its benefit in the system based on the class of QoS preferences. Consider that the classes wish to choose various objectives, the the utility function assumes $U = x_d U_d + x_l U_l$ where U_d is the utility function for average delay and U_l is the utility function for throughput, and x_d and x_l are constants. Consider that there are requirements for transaction completion time. Instead of scheduling transactions to meet deadlines, we try to minimize the number of transactions that have missed the deadlines (in a stochastic sense). Consider that each transaction class is assigned a service queue at each server , then we try to minimize the probability of the number of transactions of a class exceeding a certain threshold in the buffer. This is the tail probability P(X > b) where X is the number of transactions of a class in a queue at a server, and b is threshold is threshold for the number in the queue, beyond which transactions miss deadlines. If we include this QoS requirement, then the above utility function will be $U = x_d U_d + x_l U_l + x_t U_t$ where U_t is the tail probability utility function and x_t is a constant.

Pareto Optimality: We now have a simple formulation for classes competing for server capacity (processing rate) in order to minimize average delay (or average response time). The utility function is simply $U = x_d U_d$ as the rests of the constants are zero. Let p_j be the price per unit processing rate at server j. The maximum processing rate at server j is C_j. The problem therefore for each agent is the following: find $\{c_{ij}^*\}$ such that min $U_d = \left\{\sum_{j=1}^{N} W_{ij}\right\}$ with constraints $\sum_{j=1}^{N} \lambda_{ij} = \gamma_i \forall i$, $\sum_{1}^{N} c_{ij} \times p_j \le w_i \forall i$.

In the above problem definition, each agent will try and minimize the utility function under the wealth conbstraint and under the throughput constraint. This constraint is necessary to make sure that positive values of throughput are obtained as a result of the optimization. The transaction agents compete for processing rate at each server, and transaction servers compete for profit. The objectives of the transaction classes are conflicting as they all want to minimize their average response time. In the above formulation $W_{ij} = \lambda_{ij}/(c_{ij} - \lambda_{ij})$. This is the average number of class i transactions in queue at system j. The average delay in the system for each class i is simply the average number in the system divided by the overall throughput $\sum_{j=1}^{N} \lambda_{ij}$.

The main goal of the agent representing the transaction class is to minimize a utility function which is simply the average number in the overall system. This will also minimize the average delay or average response time of the transaction class.

Proposition 3.3 The utility function U_d is convex with respect to the resource allocation variable c_{ij} where $\lambda_{ij} \in [0,c_{ij})$, and $c_{ij} \in (0, C_j]$.

The proof follows from Gottinger [1, Chap.9].

The utility function U_d is discontinuous when $\lambda_{ij} = c_{ij}$.

Demand Set. The demand set for an agent i, given the prices (p_j of server j) of the processing rates (or capacities) at the servers is $\{c_{i1}, c_{i2}, \cdots , c_{iN}\}$ over all the servers. We use the standard techniques of optimization to find the demand set, which is given as follows for all $j \in [1,N]$.

$$c_{ij} = \lambda_{ij} + \left(\left(w_i - \sum_{j=1}^{N} \lambda_{ij} p_j \right) \Big/ \sum_{j=1}^{N} \sqrt{\lambda_{ij} p_j} \right) \sqrt{\lambda_{ij} p_j} \ .$$

Price Equilibrium: Once the demand set is obtained, then using the wealth constraints, we can solve for the equilibrium price. This is not easily tractable. However, numerical results can be computed using the tatonnement process whereby agents compute the demand set, given the processing rate prices by each server.

An iteration process between the agents and the servers takes place. This will converge to an equilibrium price, when demand equals the supply which is $\sum_{i=1}^{K} c_{ij} = C_j$.

We now state formally the result for K agents competing for processing resources from N servers.

Proposition 3.4 Consider K agents competing for processing resources from N servers. If the utility function of these agents is U_d and the performance model at the servers is an M/M/1 model, then price equilibrium and Pareto optimality exist.

The proof of this proposition is the same as described in Gottinger [1, Chap.9]. The utility function U_d is continuous and decreasing convex with respect to the allocation variables c_{ij}. The function is discontinuous when $\lambda_{ij} = c_{ij}$.

Due to this, Pareto allocations or price equilibrium may not exist. However, we solve this problem by stating that the agents, when they present their demands, have to make sure that the transaction throughput rate λ_{ij} at a server has to be lower than the capacity allocation c_{ij}. If this is not met, then the price iteration process or the tatonnement process will not converge. We assume that the servers know the transaction throughput or arrival rate

from each agent during the iteration process.

Transaction Routing

The static routing problem for the agent i, once the allocation of processing rates at the N servers is done for agent i, can be formulated as:

Find $\{\lambda_{ij}\}$ such that min $\{\sum_{j=1}^{N} W_{ij}\}$ with constraints $\sum_{j=1}^{N} \lambda_{ij} = \gamma_i \forall i$. Here, W_{ij} is the average response time for agent i traffic when sent to server (supplier) j. We use a simple M/M/1 model of the queueing system, where $W_{ij} = \lambda_{ij}/(c_{ij} - \lambda_{ij})$ (average number of agent i transactions in server j). This or the average delay can be minimized (the same result will be obtained for either one of them). The optimal arrival rate vector to the servers or the optimal flow of transactions, assuming a Poisson distribution for arrivals with rate λ_{ij} is given by

$$\lambda_{ij} = c_{ij} - \frac{\sqrt{c_{ij}}}{\sum_{j=1}^{N} \sqrt{c_{ij}}} \cdot \left(\sum_{j=1}^{N} c_{ij} - \lambda_{ij} \right)$$

This result gives the optimal flow of transactions of class i to the servers, given the capacity allocation to the agent i. Using this, a simple random routing policy which can split transaction traffic optimally can be designed. This policy does not assume the current state of the servers, the number of transactions of agent i queued for service at server j. A simple, but well known routing algorithm is illustrated here.

Dynamic Routing Algorithm: The Join Shortest Queue (JSQ) algorithm routes transactions of class i to a system j found by obtaining the minimum of the following

$$\text{Min} \left\{ \frac{Q_{i1}+1}{c_{i1}}, \frac{Q_{i2}+1}{c_{i2}}, \cdots, \frac{Q_{iN}+1}{c_{iN}} \right\}$$

where Q_{ij} is the queue length of server j or the number of transactions of class i queued up for service at server j. If there are ties, then one of the queues is picked at random (or with equal probability).

4. Conclusions

We have developed a decentralized framework for QoS provisioning based on economic models. A new definition of QoS provisoning based on Pareto efficient allocation s is given. These allocations are not only efficient (from a Pareto sense) but also satisfy the QoS constraints of competing traffic classes (or users).

We have shown that Pareto optimal allocations exist in a network economy (parallel link network), and a methodology is provided for the network service provider to price services based on the demands placed by the users. Prices are computed based on the load of the traffic classes and the corresponding demand. Furthermore, a dynamic session routing algorithm is coupled with admission control mechanisms to provide QoS to the traffic classes , for a network as well as a server economy.

Future research should address several issues related to the dynamics of the overall system. For example, if we assume time is divided into intervals, and during each time interval prices of resources are stable. Then price negotiation is done between the agents and the suppliers at the beginning of the time interval. However, each supplier (server) could have time intervals which are different from the rest. This can cause the agents to negotiate with each supplier independently.

Economic models can provide several new insights into resource sharing and QoS provisioning in future networks and distributed systems which will connect millions of users and provide a large number of serves.

Pricing and competition can provide solutions to reduce the complexity of service provisioning and efficiently utilize the resources.

REFERENCES

[1] H. W. Gottinger, "Strategic Economics in Network Industries," Nova Science Publishers, New York, 2009.

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

[3] K. R. Mount and S. Reiter, "On Modeling Computing with Human Agents," Northwestern University, Evanston, 1994.

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

[5] 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

[6] L. Kleinrock, "Queueing Systems," Wiley Interscience, New York, 1975.

[7] R. W. Wolff, "Stochastic Modeling and the Theory of Queues," Prentice Hall, Englewood Cliffs, 1989.

[8] J. K. MacKie-Mason and H. R. Varian, "Pricing the Internet," MIT Press, Cambridge, 1995, pp. 269-314.

[9] S. Shenker, "Service Models and Pricing Policies for an Integrated Services Internet," MIT Press, Cambridge, 1995, pp. 315- 337.

[10] L. Hurwicz and S. Reiter, "Designing Economic Mechanisms," Cambridge University Press, Cambridge, 2006. doi:10.1017/CBO9780511754258

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

[12] A. Harel and P. H. Zipkin, "Strong Convexity Results for

Queueing Systems," *Operations Research*, Vol. 35, No. 3, 1987, pp. 405-418. doi:10.1287/opre.35.3.405

Appendix. Proofs of Pareto Optimal Allocations

We start by giving the first derivative of P with respect to the buffer variable b:

$$P' = \frac{\rho^b (1-\rho)\log[\rho]}{\left(1-\rho^b\right)^2}, \quad \lim_{c\to\lambda} P' = -\frac{1}{(1+b)^2} \qquad \text{(A1)}$$

where $\rho = \lambda/c$. This function is negative for all $b \in [0,B]$ and for all $\rho > 0$.

The second derivative with respect to be yields

$$P'' = \frac{\left(1-\rho^b\right)\rho^b \left(\log[\rho]\right)^2 (1-\rho)}{\left(1-\rho^b\right)^3}, \quad \lim_{c\to\lambda} P'' = \frac{2}{(1+b)^2} \qquad \text{(A2)}$$

This function is positive for all $b \in [0,B]$ and all $\rho > 0$. Similarly, the function P can be shown to be continuous (smooth) and decreasing convex in c for all $c \in [0,C]$, by rewriting function P to the following:

$$P = \frac{1}{1 + c/\lambda + (c/\lambda)^2 + \cdots + (c/\lambda)^b} \qquad \text{(A3)}$$

From this the first derivative can be shown to be negative and the decond derviative to be positive for all $c \in [0,C]$, hence the proof.

In the system of parallel links the overall packet loss probability (QoS parameter) for a traffic class k is given as folllows:

$$U_K = P_{loss,k} = 1 - \prod_{j=1}^{N} \left(1 - P_k^j\right) \qquad \text{(A4)}$$

where P_k^j is the packet loss probability of TC_k on link j (or supplier).

This utility function is the same as (3.2), however, this is the packet loss probability in a network consisting of parallel links rather than a route between a source and destination as considered in (3.2).

