A Pricing Scheme in Networked Computing System with Priority

Hyoun Jong Kim and Jae Ho Juhn
Internet Economy Research Team, ETRI
161 Kajong-dong Yusong-gu, 305-350 Taejon

Abstract :

The operation of a networked computing system (NCS), such as Internet, can be viewed as a resource allocation problem, and can be analyzed using the techniques of mathematical modeling. We define a general NCS and translate that setup into a model of an economy. The preferences of users are taken as primitives, and servers in the network are viewed as productive firms with priority input queues. Each server charges a rental price for its services by priority class. We characterize optimal system allocations, and derive formulae for supporting rental prices and priority premia such that the aggregated individual user demands do not exceed optimal levels and waiting-time expectations are correct. Our economic approach has the added benefit of providing a sound basis for evaluating NCS investment alternatives, using a process analogous to free entry and exit in free-enterprise economies.

1. Introduction

A networked computing system (NCS) offers the potential for harnessing the power of a large number of possibly specialized computers linked through a network. Broadly defined a NCS consists of a network of heavy-duty computers (such as mainframes and minicomputers) called "servers", and smaller user-interface processors (such as PCs and workstations) called "clients", along with other equipment (such as storage devices, printers, ethernet cabling, microwave channels, routers, etc.), the operating systems and communication protocols. Servers respond to queries or commands from clients and provide a shared computing environment, application control, distributed databases, computation management, heavy-duty computation, and network communication services.

A client, unlike a dumb terminal, is capable of modestly sophisticated processing and computation tasks for the user. Clients could also have software designed to assist the user in achieving optimal performance from the NCS, and it is this potential that we exploit in this paper. A major challenge is how to effectively manage such systems in a diverse and changing environment. The approach we take can be viewed as an extension of the literature on performance evaluation of computing systems. The computer science branch of that literature assumes fixed steady-state arrival rates of jobs and analyzes the resulting steady state queue lengths in the system. While we draw upon results from the queuing field, our contribution is in the economic analysis of NCSs. Rather than specify fixed arrival rates, the management literature has explored the potential for prices to affect the arrival rates and thereby the steady state queue length. The case of a general NCS with priority queues and general price-sensitive stochastic arrivals is the subject of our paper.

2. Description of a NCS

We consider a NCS consisting of a finite number of clients, servers and other hardware units with well-defined, commonly known capabilities. Let M denote the set of all clients, servers and hardware units, and let $m \in M$ denote a generic "machine" in the NCS.

We assume that each machine is equipped with a priority queue system. For notional simplicity, we assume that each machine offers K classes of non-interruptible priority service, $K \equiv \{1, \ldots, K\}$, with $k=1$ being the first (or highest) priority. For analytical simplicity, we assume unlimited queue capacity.

The machines are connected with each other through a network. We model shared communication devices as distinct machines. Thus, the Internet is a special case of a general NCS. Given this modeling trick, the abstract network will consist of direct connections only; that is, if m and m' are directly connected in the network representation, then there exists a dedicated physical linkage between m and m' that is not shared by any other machine. Let A_m denote the set of machines from which m can receive direct input, and let B_m denote the set of machines to which m can send direct output. We can formally represent the abstract network as $N \equiv \{(A_m, B_m), m \in M\}$. In addition, let C denote the subset of "clients" machines: the machines where users interface with the NCS.

A typical NCS will support many programming languages. Let P denote the set of all finite "programs" in the languages of the NCS system. A machine $m \in M$ can be represented by a triplet (v_m, f_m, q_m), where v_m is the processing speed in cycles per second, $f_m(p)$ gives the output when $p \in P$ is the input, and $q_m(p)$ gives the expected number of cycles

required to process input p. Then, $q_m(p)/v_m$ is the expected execution time of program p at machine m. Note that restricted access to a particular machine can be represented by a production function that performs the desired function only if p contains a specific access password, and otherwise outputs an access-denied message. In a similar vein, a machine not capable of executing some program, outputs an unable-to-read message, and if the run time exceeds the limit (if any) specified in the program, then it outputs a time-exceeded message.

More generally, given a program p started at the machine specified by the instructions, let $P_m(p)$ denote the set of intermediate programs resulting from the execution of p that are processed by machine m. For any program p, given that p is started at the machine specified by the instructions, for each machine m, we define $Q_m(p)$ to be the expected total load on machine m imposed by program p until that program is terminated and exists from the NCS. Then, $Q_m(p)= \sum p_m \in P_m(p) q_m(p_m)$. We assume the user's client interface software can estimate $\{q_m(p),\ m \in M\}$.

Given a stationary stochastic arrival process for services, let $w=\{w_{mk},\ m \in M,\ k \in K\}$ denote the vector of expected queue waiting times at the machines. As just illustrated, a program may access a given machine several times in the course of executing. Let $\mu_m(p)=\#P_m(p)$, the number of intermediate programs of p that must be processed at machine m. Then, program p of priority class k has an expected waiting of $w_{mk}\,\mu_m(p)$ at machine m.

Finally, assuming that a user chooses a single priority class, say k, for all phases of the execution of a program p started at the machine specified by the program's instructions, the total expected throughput time is

$$\tau(p, k, w)=\sum_m [Q_m(p)/v_m + w_{mk}\,\mu_m(p)]. \quad (1)$$

3. Users' Aspects

Let I denote the set of users, and for each $i \in I$ let $C_i \subset C$ denote the set of client machines to which user i has direct access. While there is a close linkage of users and clients, we will continue to employ both terms in specific contexts. The user is a human (or group of humans in an organizational team) with preferences which form the basis for choice. In contrast, the client is a machine, and as such, has no preferences of its own. While we may endow the client machines with sophisticated software to automate many of the decision-making functions, the source of value resides with the human user.

Let S denote the class of services potentially provided by the NCS. Services $s \in S$ can be viewed as subroutines which operate on the specific data, provided that the characteristics of the data are compatible with the subroutine. Different qualities are represented formally as different services. Since programs specify the client machine at which they must start, the set of feasible programs for user i is a subset of all possible programs. We let $P_i(s) \subset P$ denote the subset of programs that will successfully deliver service s for user i; that is, if a user i initiates execution of program $p \in P_i(s)$ at the appropriate machine $m_i \in C_i$, then the NCS will execute the program and return a satisfactory output.

We model the NCS service needs of a user as a stochastic process with a specific arrival rate (or "average flow" rate). One setting that makes this assumption quite plausible is that a user is a group of individuals (such as a team of engineers or accountants), so the flow of service needs are the sum of the service needs from many individuals. Let $x_i=\{x_{iskp},\ s \in S,\ k \in K,\ p \in P_i(s)\}$ denote the vector of average flow rates for user i, where x_{iskp} denotes the average flow rate by user i for program $p \in P_i(s)$ of priority class k.

We assume that the user benefits of NCS services depend only on this average flow rate. We represent the instantaneous value to user i of x_i by a continuously differentiable concave function $V_i(x_i)$. Given our definition of a service, all $p \in P_i(s)$ and $k \in K$ are perfect substitutes in terms of user benefits. Accordingly, we assume that $V_i(\bullet)$ depends only on $X_{is}=\sum_p \sum_k X_{iskp}$, the flow rate for service s, and that V_i is strictly concave in X_{is}.

The net benefit to the user is less than $V_i(x_i)$ because services take time to execute and it costs money to use the NCS. Different programs in $P_i(s)$ may have different costs, so consider each program $p \in P_i(s)$ separately. The expected throughput time of program p and priority k is $\tau(p,k,w)$, as defined in (1). Let δ_{is} denote user i's delay cost per unit time for service s, so $\delta_{is}(p,k,w)$ is the total expected cost of delay of using program p.

Expected monetary costs are determined by the expected load imposed by the program p, the priority class k, and the rental prices of the machines. Let $r(q)=\{r_{mk}(q),\ m \in M,\ k \in K\}$, where $r_{mk}(q)$ is the rental price for q units of work with priority k at machine m. This general form allows the rental prices to depend on the job size in a general non-linear manner (e.g., quadratic in a M/G/C system). Then, the expected monetary cost of program p and priority class k is $\sum_m \sum p_m \in P_m(p) r_{mk}(q_m(p_m))$. It is convenient to derive an alternative equivalent expression for this expected cost. Let $P_{mq}(p)$ denote the subset of $P_m(p)$ which generate a load of q at machine m, and let $\mu_{mq}(p)= \#P_{mq}(p)$. Then, $\sum_m \sum p_m \in P_m(p) r_{mk}(q_m(p_m)) = \sum_q \mu_{mq}(p) r_{mk}(q)$. Clearly, the user prefers the program $p \in P_i(s)$ and priority class k that minimizes total expected costs. Accordingly, we define

$$c^*_{is}(r,w)=\min\{\delta_{is}\,\tau(p,k,w)+\sum_m \sum_q \mu_{mq}(p) r_{mk}(q) \mid p \in P_i(s),\ k \in K\}. \quad (2)$$

Let $[p_{is}(r,w),\ k_{is}(r,w)]$ denote the set of pairs of programs and priority classes that minimize total expected costs for services. By standard arguments, $c^*_{is}(r,w)$ is a continuous convex function. The net benefit to user i of an average flow rate, x_i, is

$$u_i(x_i,r,w)=V_i(x_i) - \sum_s \sum_k \sum_p x_{iskp} c^*_{is}(r,w). \quad (3)$$

We assume that user i does not anticipate how his service demands and choice of programs may affect expected waiting times w and rental prices r. As the actual waiting time a job will experience

depends on what all users have submitted in the recent past and in the future, it would be impossible for the user to predict precisely how future waiting times and rental prices will differ from their current values. In a large NCS, the actual influence of a user's submittal on future waiting times and prices will be minuscule in comparison to the variance induced by other users. Thus, it is a reasonable and computational-cost-saving assumption to take expected waiting times and rental prices as fixed at their current values.

Accordingly, each user i is assumed to choose x_i to maximize $u_i(x_i,r,w)$ taking (r,w) as fixed, and we let $x_i(r,w)$ denote the set of optimal demands. Specifically, $x_i(r,w)$ is characterized by

$\partial V_i(x_i)/\partial x_{iskp} \leq c^*_{is}(r,w)$ for all s, k and $p \in P_i(s)$, and $\partial V_i(x_i)/\partial x_{iskp} < c^*_{is}(r,w)$ implies $x_{iskp}=0$. (4)

Then the demand function for services, $X_{is}(r,w)=\sum_k\sum_p x_{iskp}(r,w)$, is finite, single-valued and continuous for all non-negative (r,w).

We pause to interpret the act of choosing flow rates X_{is}. We model the arrival of potential NCS service requests as an exogenous stochastic process that is independent of the pricing of the NCS. Suppose the exogenous arrival rate of questions requiring NCS service s is X_0, with the interpretation that if the minimum cost of delivering s were 0, then X_0 would be the average flow demand for service s. A user receiving one of these questions must decide whether or not to actually request NCS service. Let λ_{is} denote the probability of accepting the question and submitting service requests. Given the exogenous arrival rate X_0, the average flow demand for s from this user will be $X_{is}=\lambda_{is}X_0$. Thus, the choice of λ_{is} is equivalent to the choice of an average flow rate X_{is}. The optimal choice maximizes (3) and equates the marginal benefit with the cost $c^*_{is}(r,w)$.

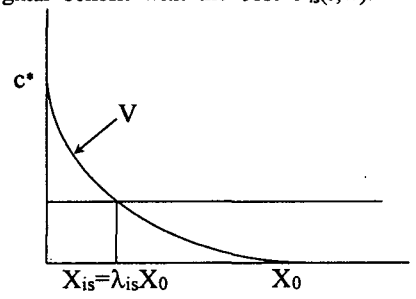

Fig.1 Optimal Choice of Average Flow Rates $\phi_m(\underline{x})$

Fig. 1 illustrates this optimization graphically. The downward-sloping "demand" curve gives the marginal benefit of service s as a function of the average flow rate of service s. The horizontal "supply" curve gives the cost c^*_{is}. An exogenous change in V_i or c^*_{is} would shift these curves, thereby causing a change in the optimal X_{is}.

Observe that we have resolved the well-known integer nature of computation problems. While each service s is an indivisible entity, we have treated the demand choice as if it could be subdivided in any matter to accommodate the certainty-equivalent flow rate of X_{is}. However, in actual implementation, the service requests enter the NCS as indivisible units. We accommodate this integer requirement by modeling the arrival process as a stochastic process with arrival rate X_{is}, and we hold input in the machine queues until ready to be processed. The resulting delays are reflected in the waiting times w.

4. Optimal Resource Allocation

The most natural definition of an equilibrium would be that average demand $z_m=\sum_i\sum_s\sum_k\sum_p x_{iskp}Q_m(p)$ at each m equals the "supply" v_m. However, with a Poisson arrival process, equality of the arrival rate and the service rate would yield infinite expected queue waiting times. But if queue waiting times are infinite, then there is no value to computation services, so demand would sink to zero, an inconsistency. Clearly, we want to have consistency between expected waiting times, demand, and actual times.

The entire array of demands for all users, services, priority classes, and programs is denoted by $\underline{x} \equiv \{x_{iskp},\ i\in I,\ s\in S,\ k\in K,\ p\in P_i(s)\}$. Given the load functions, $\{q_m(p);\ m\in M\}$, $\underline{x}$ completely determines the stochastic process in the NCS. In general, the expected waiting time at machine m will depend on the distribution of job arrival rates by priority class and job size. This distribution is given by

$\psi_{mkq}(\underline{x}) \equiv \sum_i\sum_s\sum_k\sum_p x_{iskp}\,\mu_{mq}(p)$; (5)

let $\varphi_m \equiv \{\varphi_{mkq},\ k\in K,\ q\in IN$, where IN denotes the set of integers$\}$ denote the matrix of job arrival rates at m by priority class and job size. The aggregate flow (measured in cycles per second) to m in priority class k is then $z_{mk} \equiv \sum_q q\,\psi_{mkq}$.

We assume that the expected waiting time at m given priority class k is a function of the distribution matrix $_m$ and capacity v_m

$w_{mk}=\Omega_k(_m(\underline{x});\ v_m)$, (6)

when $\Omega_k(\bullet;v_m)$ is continuously differentiable, strictly increasing and convex as long as $\sum_k z_{mk}<v_m$, and $\Omega_k(0;v_m)=0$. Further, $\Omega_k(\psi_m(\underline{x});\ v_m) \to \infty$ as $\sum_k z_{mk} \to v_m$. We also assume that $\partial\,\Omega_j/\psi_{mkq}$ for all k<k; in other words, the incremental waiting time imposed on priority j jobs is greatest for new arrivals of the highest priority jobs. $\Omega_k(\psi_m(\underline{x});\ v_m)$ gives the physical tradeoff between waiting time and throughput for priority class k, as illustrated in Fig. 2, when w_{mk} is plotted in the downward direction. Users prefer points to the northeast of this convex boundary, i.e. they prefer more throughput and less waiting time.

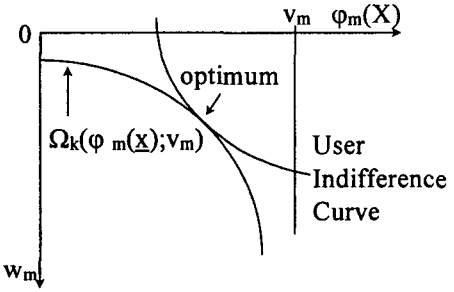

Fig. 2 Tradeoff between Waiting Time and Throughput

To derive the optimal trade-off we need to define a system-wide welfare function. It is natural to take the sum of non-pecuniary user benefits:

$W(x,w) \equiv \sum_i [V_i(x_i) - \sum_s \delta_{is} \sum_k \sum_p x_{iskp} \tau(p,k,w)]$ (7)

We assume here that there are no costs to operating the NCS and the services, so the only social costs are due to congestion. However, it would be a trivial matter to extend the analysis to include positive operating costs.

We now seek an allocation of demands, $\underline{x} = \{x_{iskp}, i \in I, s \in S, k \in K, p \in P_i(s)\}$, and waiting times w, that maximize W(x,w) subject to (6). The Kuhn-Tucker conditions of this welfare maximization are:

$\partial V_{is}/\partial x_{iskp} - \delta_{is} \tau(p,k,w) \leq \sum_m \sum_q \mu_{mq}(p) \sum_j [\partial \Omega_j / \partial \psi_{mkq}] \gamma_{mj}$ for all i, s, k and $p \in P_i(s)$;

$\partial V_{is}/\partial x_{iskp} - \delta_{is} \tau(p,k,w) < \sum_m \sum_q \mu_{mq}(p) \sum_j [\partial \Omega_j / \partial \psi_{mkq}] \gamma_{mj}$ implies $x_{iskp}=0$; (8)

and $\gamma_{mk} = \sum_i \sum_s \sum_p \delta_{is} x_{iskp} \mu_m(p)$. (9)

where γ_{mk} is the Lagrangian multiplier for (6). Equation (9) defines the shadow price of waiting time for priority class k at m. Equations (8) requires the marginal net benefit of service s at priority class k to be less than or equal to the total shadow cost of the induced incremental waiting times; if less, then the optimal $x_{iskp}=0$.

Let $\gamma \equiv \{\gamma_{mk}, m \in M, k \in K\}$. We conclude the following:

Theorem 1: *There exists a unique* (x^*, w^*, γ^*) *that maximizes* W(x,w) *subject to (6) and satisfies (8) and (9).*

We now come to the question of whether we can support this welfare-maximizing allocation with a price mechanism. That is, when will the vector of user demands functions, $x(r,w)=\{x_i(r,w), i \in I\}$, which optimize W(x,w)? Recall from Section 3 that if $x_{iskp}(r,w)>0$, then $\partial V_{is}/\partial x_{iskp} = c^*_{is}(r,w) = \delta_{is}(p,k,w) + \sum_m \sum_q \mu_{mq}(p) r_{mk}(q)$. Then, $x_{iskp}(r,w)$ will satify (8) iff $\sum_m \sum_q \mu_{mq}(p) r_{mk}(q) = \sum_m \sum_q \mu_{mq}(p) \sum_j [\partial \Omega_j / \partial \psi_{mkq}] \gamma_{mj}$. The obvious solution is to set

$r_{mk}(q) = \sum_j [\partial \Omega_j / \partial \psi_{mkq}] \gamma_{mj}$ (10)

In other words, the welfare maximizing rental price for machine m with priority k must equal to average cost of aggregate delays weighted by the waiting-time and throughput tradeoff at m. given our assumptions about $\Omega_k(\bullet; v_m)$, the rental prices are decreasing in priority k: $r_{mk} > r_{mk+1}$. Hence, we could think of r_{mk} as the base price, and $(r_{mk} - r_{mK})$ as the premium for higher priority service.

However, (10) is not an explicit formula for r_{mk}, since r_{mk} enters the right-hand side via $x_{isjp}(r,w)$ and $\psi_{mkq}(\underline{x}(r,w))$. We conclude the following:

Theorem 2: *There exists an* (r^*, w^*) *such that* $x_i(r^*,w^*)$ *maximizes* $u_i(x_i,r^*,w^*)$ *for all* $i \in I$, $(x(r^*,w^*), w^*)$ *maximizes* W(x,w), *and* $w^*_{mk} = \Omega_k[\psi_m(x((r^*,w^*)); v_m]$ *for all* $m \in M$ and $k \in K$.

In other words, given (r^*, w^*), individual users choose demands $x_i(r^*,w^*)$, which in turn generate expected waiting-times w^* satisfying (6). Furthermore, these demands and waiting times maximize welfare W(x,w).

An alternative interpretation in terms of competitive equilibrium can be made. Let z^* denote the welfare-maximizing aggregate flows from Theorem 1. Then, Theorem 2 asserts that the existence of rental price r^* and expected waiting times w^*, so (i) demands $x_i(r^*,w^*)$, equals optimal flows $\underline{x}^*$, and (ii) these demands via the queues, (6), generate expected waiting times w^*. We can call this interpretation a "stochastic" in that expected waiting times are correct and "excess demand" in terms of flow rates $(z - z^*)$ is zero, where z and z^* are calculated using (6).

5. Cocnclusion

The issue we explored is the design of a mechanism that can potentially achieve the highest level of resource allocation. To achieve this goal, we have successfully modeled a NCS with priority queues and general stochastic arrivals as an economic resource allocation problem. Our economic approach has the added benefit of providing a sound basis for evaluating NCS investment alternatives, using a process analogous to free entry and exit in free-enterprise economies. All of these theoretical advantages motivate further study. In addition, the system impact and profitability of a new service installed on a server, such as home shopping, could be investigated via simulation in much the same way as for infrastructure investments.

In future research we will consider alternative priority queue systems and incentive compatibility issues. We will also extend the theoretical model to account for organizational constraints such as the need to recover the cost of operating the NCS.

References

[1] Chaudhury, A., Stahl, D., and Whinston, A., "The Economic Theory Foundation for Neural Computing Systems," in A.Whinston and J.Johnson (eds.), *Advances in Artificial Intelligence in Economics, Finance and Management*, J.A.I. Press, 1992.

[2] Ferguson, D., Yemini, Y., and Nikalson, C., "Microeconomic Algorithms for Load Balancing in Distributed Computer Systems," Research Report, T.J. Watson Research Center, Yorktown Heights, New York, October 1989.

[3] Hyoun J. Kim, and Huh, H., A Mathematical Model for the Resource Allocation in Computer Networks, *Proceedings of 3rd Annual Int'l Conference on Industrial Engineering Applications and Practice*, December 1998.

[4] Shenker, Scott, "Service Models and Pricing Policies for an Integrated Services Internet", working paper, Xerox co. Palo Alto Research Center, September 1995.