A Mechanism for Congestion Control in Computer Networks

Srinivasan Keshav

Computer Systems Research Group,
Computer Science Division,
Department of Electrical Engineering and Computer Science,
University of California, Berkeley,
Berkeley, CA 94720.

We approach the problem of congestion control in computer networks from the perspective of mechanism design. The network is modeled as an economy and some simplifying assumptions are made to derive a congestion control scheme. Subsequently, some of the assumptions are relaxed and the solution is generalized.

A Mechanism for Congestion Control in Computer Networks

1. Introduction

Computer networks 1. have long suffered from congestion. The construction of wide area networks in the 70's was accompanied by research into a number of schemes to control congestion [10, 19]. The perceived failure of these efforts, as well as a rapid growth in the scale and geographical extent of computer networks has sparked renewed interest in congestion control (for a detailed survey, see [13]).

By definition, congestion occurs when an increase in the offered load results in a decrease in the effective throughput of the network [15]. The basic cause is that the short term packet arrival rate at some gateway exceeds its service rate 2. [17]. At this point, packets are buffered, leading to delays. The additional delay can cause sources to retransmit, increasing the load on the bottleneck. This feedback leads to a rapidly deteriorating situation where retransmissions dominate the traffic, and effective throughput rapidly diminishes [11, 21]. Further, if there is gateway to gateway flow control, new packets may not be allowed to enter the gateway and so packets might be delayed at a preceding gateway as well. This can lead to deadlock where all traffic comes to a standstill [26].

Existing congestion control schemes treat congestion as an isolated problem and propose ad hoc solutions that are not entirely satisfactory [12]. We approach the problem from a different perspective: a gateway is thought of as a principal that allocates throughput and delay to the conversations that pass through it. If this allocation is within the capacity constraint of the gateway, congestion cannot occur. The key to network efficiency lies in selecting an appropriate allocation of gateway resources. Since this allocation will, in general, depend on the preferences of the sources, this is naturally modeled as a mechanism design problem. Designing a congestion control scheme using the techniques of mechanism design [14] assures us that the scheme is efficient and not subject to manipulation by ill-behaved sources. Thus, we believe that this is an appropriate theoretical basis for designing congestion control mechanisms.

In this paper, we present an economic model of a computer network. We then make some simplifying assumptions to derive a mechanism for congestion control. Subsequently we relax some of the assumptions and generalize the solution. We will not present details of implementation of the scheme in real-life networks: these will be discussed in a forthcoming thesis.

2. Network Model

A network is modeled as a set of gateways and sources. All gateways are assumed to be under administrative control of the network manager, henceforth called the The users of the network, who may be human users or application programs, are called

2.1. Agent model

Agents send packets to the network to be transferred from some source to some destination along a path of gateways. This transfer is characterized by two quantities - the throughput (rate at which packets are transferred), and the delay (average time that a packets takes to travel through the network). Agents prefer a higher throughput and a lower delay. They are charged money for this transfer.

Agents are assumed to to be rational: that is, they behave in a way such that their utility is maximized. They are assumed to be able to control the rate at which they send packets to the network. They have a budget limit per packet, and a preference for a transferable good (say, money). Finally, we assume that an agent has some private information (type) that describes the agent's preferences over delay and

1. We will use the following standard terminology:- of data send information in the form of to that store and forward these packets. Gateways route and schedule incoming packets on outgoing lines, placing data in when the arrival rate exceeds the service rate. The stream of packets between a source of data and its recipient is called a The net rate at which packets enter a network is its The rate of delivery of distinct data packets summed over all recipients of data is the

2. 3. The service rate is determined by the processing time per packet and the bandwidth of the output line. Thus, the bottleneck (binding constraint) could be either the gateway CPU or the outgoing line: in either case, there will be congestion.

throughput 4.. This is meant to model the fact that some agents (such as Telnet-like applications) prefer to have low delay, whereas others (such as FTP-like bulk transfer applications) prefer to have high throughput, perhaps at the cost of higher delay. The gateway is not aware of the type of the agents.

2.2. Principal Model

The principal is charged with the duty of allocating *buffers* and *bandwidth* to the agents. It does so by observing a buffer allocation strategy and a service discipline. As network designers, our goal is to design these strategies such that network efficiency is maximized and congestion is avoided.

The principal needs to know two pieces of information to determine the optimal allocation - the type of each agent, and the price that the agent is willing to pay for the data transfer. The principal could ask each agent for its type, but an agent could lie. Thus, it is necessary to design an incentive compatible mechanism that will elicit the correct type from the agent [23].

The principal also needs to know how much an agent is willing to pay. This is done by negotiating a *contract* between each agent and the principal during conversation set up. We model two forms of contracts: guaranteed performance contracts and best effort contracts 5..

2.2.1. Guaranteed performance contracts

In virtual circuit networks, such as those assumed by the design of DASH [2], where *guarantees* of performance are offered, a contract promises a given level of performance (utility level) for a price. The agent states the throughput and delay it desires, and the principal states the price for the data transfer. If agent cannot afford the price, it will have to scale down its demand, or try at an off peak hour. The principal may need to forbid new contracts to allow it to satisfy existing contracts. If the set of contracts is such that the output line is never overloaded, there can be no congestion. (Of course, we still have to enforce the contracts.)

2.2.2. Best effort contracts

If a principal is not allowed to reject any contracts (as in most datagram networks), existing conversations may experience a performance degradation due to a new contract. They can only be given a best-effort guarantee by the principal. In many cases, this may be sufficient. The contract is thus of the form: the agent specifies the budget for the data transfer. The principal promises to allocate resources so that the agent derives maximal utility for the price it pays. The agent agrees to obey the principal's signals indicating the rate at which it should send data to the network.

Suppose some agent violates a contract, what should be done? In classical contract theory, if a principal discovers a contract violation, the principal is free to do anything (shoot the agent, for example). The idea is that in equilibrium, the contract will never be violated, so actions off the equilibrium path in the game tree can be arbitrarily defined.

In contrast, in computer networks, contracts may be violated even if the agent did not desire to do so. For example, an agent may agree to a rate limit ρ, but may temporarily send packets at a rate $\rho+\delta$ due to a timer glitch in an intermediate gateway. In this case, it is not appropriate to shoot the agent. On the other hand, the increased rate may cause disutility to the other agents.

The solution to the problem is to penalize the violator, so that the utility of the other agents is undisturbed. This could be done by increasing its delay, dropping its excess packets or by reducing its share of the bandwidth. In any case, the misbehavior of one agent should not cause disutility to any other agent (the *protection property*).

4. For example, the type could be the increase in throughput that would exactly offset a unit increase in delay.

5. 6. These two types of contracts have also been proposed for the Asynchronous Transfer Mode protocols of future Broadband ISDN networks, and for the DASH communications architecture [1].

3. Mathematical Modeling

In this section we present a mathematical model of the network. We present precise definitions of terms that have been loosely described in the previous sections.

3.1. Notation

There is one principal and I agents, indexed by i. An agent i has a utility function $U_i(d_i, \rho_i, w_i, \theta_i)$ that depends upon its allocated delay d_i, its allocated throughput ρ_i, the price that it has to pay for communication services w_i, and its private information θ_i. θ_i describes its preferences over delay and throughput.

In a Guaranteed Service (GS) Contract, an agent i of type θ_i presents the principal with the desired throughput, $\hat{\rho}_i$, desired maximum delay, $\hat{d}_i$ and its type, θ_i. The principal replies with the cost of the transfer, w_i.

In a Best Effort (BE) Contract, agent i presents its budget limit, B_i. For each agent i, the principal allocates delay d_i, and throughput ρ_i 7.. The mechanism may make it necessary to impose a monetary transfer to enforce incentive compatibility: this transfer is adjusted against the budget.

The service rate of the principal is L. We use the notation $\hat{x} \in \underset{x}{arg\max} f(x)$ to mean that $\hat{x}$ is a value that maximizes the function $f(x)$.

3.2. Definitions

Efficiency

An allocation is (Pareto) efficient if the sum of the utilities of the agents is maximized. That is, for all i,

$$(d_i, \rho_i, w_i) \in \underset{(\hat{d}_i, \hat{\rho}_i, \hat{w}_i)}{arg\max} \sum_i U_i(\hat{d}_i, \hat{\rho}_i, \hat{w}_i, \theta_i)$$

We consider efficiency to be the social welfare function for the mechanism.

Congestion Control

A scheme controls congestion at a gateway if, at all times, the rate at which traffic arrives is less than the service rate of the gateway 9..

$$\sum_i \rho_i \leq L$$

A scheme controls congestion if it controls congestion at all the gateways in the network.

Fairness

A scheme is fair if an agent does not envy the allocation given to any other agent (envy-free allocation) 10..

$$U_i(d_i, \rho_i, w_i, \theta_i) \geq U_i(d_j, \rho_j, w_j, \theta_j)$$

---

7. 8. The instantaneous throughput ρ_i is the inverse of the interpacket sending time.

9. If a gateway has K buffers, and the input rate $R = \sum_i \rho_i$ exceeds the service rate L for a time μ such that $R \times \mu > K$, then packets will be dropped and congestion can occur. To avoid introducing timing issues into this model, we choose to have a conservative congestion control constraint. Note that choosing an efficient operating point will guarantee congestion control if no agent desires infinite delay. Thus, this constraint should be trivially satisfied.

10. This is only one of a number of alternate definitions of fairness. See [18, 27] for some other definitions.

Incentive compatible to reveal type (ICT)

We assume the revelation principle [14] and hence propose a direct mechanism in which agents report their type, θ_i. Incentive compatibility implies that reporting θ_i is better than reporting any other value (to be precise, in dominant strategy equilibrium an agent will report its type as θ_i). That is, for any $\hat{\theta}_i \ \varepsilon \ \Theta_i$, the domain of all possible types for agent i,

$$U_i(d_i,\ \rho_i,\ w_i,\ \theta_i;\theta_i) > U_i(d_i,\ \rho_i,\ w_i,\ \theta_i;\hat{\theta}_i)$$

Protection from ill-behaved sources

The actions of agent j cannot adversely affect the utility of agent i. For all $j \neq i$,

$$\frac{\partial U_i(d_i,\ \rho_i,\ w_i,\ \theta_i)}{\partial r_j} \geq 0$$

Incentive compatible to obey signals (ICS)

In a BE contract, if an agent is allocated a rate ρ_i, it is in the best interest of the agent to send data at that rate. For all $\hat{\rho}_i$,

$$U_i(d_i, \rho_i,\ w_i,\ \theta_i; r_i = \rho_i) \geq U_i(d_i, \rho_i,\ w_i,\ \theta_i; r_i = \hat{\rho}_i)$$

4. Restricted Solution

In this section we make a number of assumptions to simplify the problem, and then present a solution. Section 5 removes some of these restrictions.

Assumptions

We make three sets of assumptions. The first set of assumptions (1-3) is regarding the network and serves to reduce the complexity of the problem. The second set of assumptions (4-6) is required to guarantee the existence of a solution based on mechanism design. Finally, there are some assumptions (7-9) that are made to clarify the exposition of the solution and do not affect the generality of the results. These assumptions will be removed in Section 5.

1) *Single gateway*: We consider networks with a single gateway.
2) *Quastatic solution*: The parameters describing the network are assumed to change slowly with time. The throughput and delay seen by an agent are the averages over sufficiently long intervals of time.
3) *Ignore fairness and budget balancing*: We do not require the allocation to be fair, nor do we require that the net transfer of money be zero.
4) *Additively separable U_is : We assume that the utility function of agent i can be written in the form*
$$V_i(\rho_i,\ d_i,\ \theta_i) + w_i(\theta_i)$$
where w_i is the price charged (transfer).
5) $V_i(.)$ *differentiable, continuous and concave*: We need this to assure us that the maximization problem has a well defined solution.
6) *Utility functions known to the principal*: We assume that the form of the utility function of the agent, (though not the type parameter), is known to the principal.
7) *Fixed number of agents*: We assume that the number of agents in the network is static, and that all of them transfer data through the gateway.
8) *Two agents*: We assume that I = 2.
9) *Only best effort contracts:* We assume that all agents desire only BE contracts.

Given the assumptions above, we now have to solve a much simpler problem. We state this as the following:

Maximize

$$V_1(\rho_1, d_1, \theta_1)+V_2(\rho_2, d_2, \theta_2)$$

given

$$\rho_1+\rho_2 < L,$$

$$U_i(\rho_i, d_i, w_i, \theta_i; \theta_i) \geq U_i(\rho_i, d_i, w_i, \theta_i; \hat{\theta}_i)$$
and
$$U_i(r_i=\rho_i, d_i, w_i, \theta_i) \geq U_i(r_i=\hat{\rho}_i, d_i, w_i, \theta_i)$$

In words, maximize efficiency subject to the capacity constraint (congestion control constraint), ICT and ICS.

We treat the allocation of throughputs $\bar{\rho}$ and the allocation of delays $\bar{d}$ to be public goods. Let the reported types be the vector $\hat{\Theta} = (\hat{\theta}_1, \hat{\theta}_2)$. Let $\bar{\rho}^*$ and $\bar{d}^*$ be the throughput and delay vectors that maximize the sum $V_1(.)+V_2(.)$, given $\hat{\Theta}$. Using the standard Clark-Groves scheme, we get the expression for the transfer to be (for $i \neq j$)

$$w_i(\theta_i) = V_j(\bar{\rho}^*, \bar{d}^*, \theta_j)$$

$$=> \; w_i(\theta_i) = V_j(\rho^*_j, d^*_j, \theta_j)$$

Let us check ICT for agent 1. We know that $(\bar{\rho}^*, \bar{d}^*)$ maximize

$$V_1(\bar{\rho}^*, \bar{d}^*, \hat{\theta}_1) + V_2(\bar{\rho}^*, \bar{d}^*, \hat{\theta}_2)$$

Hence, $\hat{\theta}_1 = \theta_1$ will maximize

$$V_1(\bar{\rho}^*, \bar{d}^*, \theta_1) + V_2(\bar{\rho}^*, \bar{d}^*, \hat{\theta}_2)$$

$$= V_1(\bar{\rho}^*, \bar{d}^*, \theta_1)+w_1(\theta_1)$$

$$= U_1(\bar{\rho}^*, \bar{d}^*, w_1, \theta_1)$$

ICS is easily achieved. The principal is able to observe the rate r_i at which agent i sends data. If this rate is not the same as the allocated rate ρ_i, the principal can penalize the agent, by charging it more money. This makes it incentive compatible to obey the allocated rate ρ_i. Note that in equilibrium, all agents will send at their allocated rate. If there is a deviation (which can come about if the agent unilaterally changes r_i), then the deviating agent alone is punished. In other words, the punishment scheme described above ensures the protection property.

We have verified that the solution above satisfies our requirements of ICT, ICS, protection and efficiency. However, we have not specified how to determine the optimal allocation $\bar{\rho}^*, \bar{d}^*$, and we have not verified fairness. The optimal allocation depends upon the nature of the V_is, which is specific to the network under consideration, and fairness is a property of the optimal allocation. Hence, we will not discuss these issues further.

5. Generalizations

We now consider generalizations of the solution achieved by relaxing some of the assumptions.

5.1. N agents

The solution generalizes trivially for the case where there are N agents. The transfer is then

$$w_i(\theta_i) = \sum_{j \neq i} V_j(\bar{\rho}^*, \bar{d}^*, \theta_j)$$

The tests for ICS and ICT carry through.

5.2. Allow variable number of agents in the network

In any real network, we cannot be sure of how many users will be present at any given time. However, if the number of of agents is not fixed, then the maximization problem is not well defined, since the principal may not be aware of the utility of some agent. Thus, the solution above must be modified to account for this problem.

Let us partition time into intervals such that in each interval the number of agents is fixed. As conversations start and terminate, we will move from one interval to the next. We can then solve for the optimal allocation *per interval*.

Let us consider this for some gateway G. At time zero, we assume that G has no conversations flowing through it, and there is no game to be played. When G detects the start of a new conversation through it (say, by noting a call set up packet), it increments the number of agents in the game and recomputes the optimal allocation. This information is sent to all the agents, who will then readjust their sending rates in accordance with this signal (ICS). A similar recomputation is necessary when conversations are terminated.

An interesting situation develops when a GS contract is proposed by the new agent. If preexisting GS contracts lead to resource commitments such that the new GS demands cannot be satisfied, then the agent will be turned away. The agent may renegotiate for a lower resource demand or may try when some existing GS agents have terminated. In any case, a new GS contract will be allowed only if the resource demands of the new agent can be satisfied.

5.3. Allow Guaranteed Service contracts

Agents who desire guaranteed service (GS) contracts can be modeled as agents who have infinite utility when the delay and throughput allocation made to them are exactly what they desire, else they have infinite disutility. This will constrain the maximization problem to allocate them exactly the throughput and delay they ask for. In other words, for an agent *i* who asks for guaranteed service of $(\hat{\rho}_i, \hat{d}_i)$,

$$U_i(\hat{\rho}_i, \hat{d}_i) = \infty,$$
$$U_i(\rho_i \neq \hat{\rho}_i, d_i \neq \hat{d}_i) = -\infty$$

Unfortunately, this has two adverse side effects: if one GS agent gets the desired service, and another does not, the maximization problem is undefined. Further, if a GS agent is satisfied, then the BE agents can be allocated arbitrary allocations, which may be non-optimal.

Our solution is to reduce the principal's capacity by the the allocation made to GS agents. All GS agents are given the service they desire (we know that this is possible, since if this were not so, the GS agent would not have been admitted into the game - see section 5.2). The channel bandwidth is reduced by the total bandwidth allocated to GS agents. The efficiency maximization is then done over all BE agents with this reduced capacity. This solution allows GS agents to get the service they desire, and will not cause the problems raised by infinite utilities.

The rent charged from the GS agents depends upon the fraction of the channel capacity they use. They should pay for their share of capacity costs, marginal operating costs, plus some rent to the BE agents to compensate them for their lower priority. The principal may desire to give an incentive to GS agents to use the network at off peak times. Borrowing from work in classical peak load pricing [25], this can be achieved by charging peak time GS agents for capacity and marginal costs, and off-peak time GS agents for marginal costs alone. The question of how much rent should be paid to BE agents is an interesting open problem.

5.4. Relaxing Other Assumptions

We now present the limitations imposed by each of the remaining assumptions, and how we hope to overcome these limitations.

The *single gateway* assumption is unrealistic. We made this assumption so that we would able to completely understand the dynamics of a single gateway. We do think that an approach similar to the one presented above can be adopted for the multiple gateway case, and will consider this in future work.

The *quasistaticity* assumption ignores the dynamic nature of real networks. We believe that mechanism design is not a good theoretical basis to study non quasistatic systems, since timing as such does not enter into the picture in this approach. Thus, the assumption is due to an inherent flaw in the approach.

Fairness is a property of the optimal allocation. Essentially, we would like an allocation to maximize efficiency and be equitable as well. Varian [27] has showed that for large classes of economies, efficient and equitable allocations do not exist. Thus, for simplicity, we do not require fair allocations. We still need to study the nature of the optimal allocation to see whether it is possible to have efficient and fair allocations in this economy.

Budget balancing depends upon the exact form of V_i. Since we do not model this at the moment, we are unable to check for budget balancing.

The assumptions regarding additive separability of Us, and differentiability of Vs do not strike us being very strong. The difficulty lies in the assumption that the principal is aware of the form of the V_is. This is a strong assumption to make and may not be true in all cases. In the mechanism proposed by Sanders [23], the agent communicates its marginal utility from delay and throughput to the principal, who does not need to know the form of V. It may be possible to use a variant of this approach in our problem.

6. Related Work

The notion of using game theory to model the congestion control problem, and in particular, a market solution to it, was proposed in an early paper by Nagle [17]. Sanders has discussed optimal flow control schemes using incentive compatibility to determine the marginals of utility functions of the agents [22, 23]. However, her work is limited to the case of a first-come-first-served service discipline where delay and bandwidth cannot be independently allocated. She does not consider GS and BE contracts or the protection property. Douligeris and Majumdar [6-8] have used the concepts of Nash and Stackleberg equilibria to determine optimal flow allocations. However, they try to maximize the total power [4], as opposed to the total utility. They too do not consider pricing. Shenker [24] has used a similar analysis to investigate the properties of Fair Queueing gateways.

Economic models of computer systems have been investigated by Yemini [9] and Miller [16]. However, they do not consider a game theoretic analysis of their economic system.

Economists have considered the ‘Peak Load Pricing’ (PLP) problem in the context of public utility pricing [3, 5, 20, 25, 28]. However, standard PLP solutions make several assumptions that are not valid in our case. Most important, in the PLP problem, demand is for quantity alone, whereas we need to account for a demand for quantity (throughput) as well as quality (delay).

7. Conclusions and Discussion

Thus far, congestion has been treated as an isolated problem, and ad hoc methods have been used to control it. We believe that mechanism design is a sound theoretical basis to attack the problem. Given certain assumptions, we have designed an incentive compatible and efficient mechanism to control congestion. We have discussed the consequences of relaxing some of the assumptions. We have introduced the notion of best effort and guaranteed service contracts, and have described how they can implemented in our mechanism.

The major weakness of our solution is that it makes some simplifying assumptions that may not hold in real networks. We recognize these limitations, and hope to address them in future work.

Bibliography

1. D. P. Anderson and D. Ferrari, A Software Architecture for Network Communication, *Proc. 8th International Conf. on Dist. Computing Systems*, San Jose, CA, June 1988, 376-383.
2. D. P. Anderson and D. Ferrari, An Overview of the DASH Project, Comp. Sci. Dept. Tech. Rpt. 88/406 , University of California, Berkeley, Feb. 1988.
3. E. Bailey and E. Lindenberg, Peak Load Pricing Principles : Past and Present, in *New Dimensions in Public Utility Pricing*, MSU Public Utilities Studies, 1976.
4. K. Bharath-Kumar and J. M. Jaffe, A New Approach to Performance-Oriented Flow Control, *IEEE Trans. on Communication COM-29*, 4 (April 1981), 427-435.
5. M. Boiteux, Peak Load Pricing, *Journal of Business 33* (April 1960), 157-179.
6. C. Douligeris and R. Mazumdar, An Approach to Flow Control in an Integrated Environment, CU-CTR-Tech. Rpt.-50, Columbia University, 1987.
7. C. Douligeris and R. Mazumdar, On Pareto Optimal Flow Control in a Multiclass Environment, *Proc. 25th Allerton Conference, University of Illinois* , October, 1987.
8. C. Douligeris and R. Majumdar, User Optimal Flow Control in an Integrated Environment, *Proc. of the Indo-US Workshop on Systems and Signals*, January 1988. Bangalore, India.
9. D. Ferguson, Y. Yemini and C. Nikolaou, Microeconomic Algorithms for Load Balancing in Distributed Computer Systems, 1987.
10. M. Gerla and L. Kleinrock, Flow Control : A Comparative Survey, *IEEE Trans. on Communication COM-28*, 4 (April 1980), 553-574.
11. V. Jacobson, Congestion Avoidance and Control, *Proc. ACM SigComm* , August 1988, 314-329.
12. S. Keshav, Qualifying Examination Proposal : Congestion Control in Computer Networks, April 1989.
13. S. Keshav, Congestion Control in Computer Networks, *PhD thesis (in preparation)*, University of California, Berkeley , June 1991.
14. J. Laffont, in *Fundamentals of Public Economics*, Massachussetts Institute of Technology Press, Cambridge, 1988.
15. C. Lemieux, Theory of Flow Control in Shared Networks and Its Application in the Canadian Telephone Network, *IEEE Trans. on Communication COM-29*, 4 (April 1981), 399-413.
16. M. S. Miller and K. E. Drexler, Markets and Computation : Agoric Open Systems, in *The Ecology of Computation*, Elsevier Science Publishers/North-Holland, 1988.
17. J. Nagle, On Packet Switches with Infinite Storage, *IEEE Trans. on Communications COM-35* (1987), 435-438.
18. E. Pazner, Pitfalls in the Theory of Fairness, in *Social Goals and Social Organization : Essays in the Memory of Elisha Pazner*, Cambridge University Press, New York, 1985.
19. L. Pouzin, Methods, Tools and Observations on Flow Control in Packet-Switched Data Networks, *IEEE Trans. on Communication COM-29*, 4 (April 1981), 413-426.
20. I. Pressman, A Mathematical Formulation of the Peak Load Pricing Problem, *Bell Journal of Economics and Management Science 1* (Autumn 1970), 304-326.
21. K. K. Ramakrishnan, Analysis of a Dynamic Window Congestion Control Protocol in Heterogenous Environments Including Satellite Links, *Proc. 1986 IEEE Symp. on Computer Networks*, 1986.
22. B. A. Sanders, An Asynchronous, Distributed Flow Control Algorithm for Rate Allocation in Computer Networks, *IEEE Trans. on Computers 37*, 7 (July 1988).

23. B. A. Sanders, An Incentive Compatible Flow Control Algorithm for Rate Allocation in Computer Networks , *IEEE Trans. on Computers 37*, 9 (September 1988).

24. S. Shenker, Allocating Quality Fairly (Preliminary Notes), *Private Communication*, Jan, 1989.

25. P. O. Steiner, Peak Loads and Efficient Pricing, *Quarterly Journal of Economics 71* (November 1957), 585-610.

26. A. S. Tanenbaum, in *Computer Networks*, Prentice Hall, Englewood Cliffs, NJ, 1981.

27. H. R. Varian, Equity, Envy, and Efficiency, *J. Econ. Theory 9* (1974), 63-91.

28. O. E. Williamson, Peak Load Pricing : Some Further Remarks, *Bell Journal of Economics and Management Science 5* (Spring 1974), 223-228.