Microeconomic Algorithms for Flow Control in Virtual Circuit Networks (Subset in Infocom 1989)

September 13th, 1995

Donald Ferguson*,**
Christos Nikolaou*
Yechiam Yemini**

*IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598
and
**Computer Science Dept.
Columbia University,
New York, NY 10027

Abstract

This paper presents a flow control mechanism based on an economic model of optimum decentralized decision making. The virtual circuits (VCs) of the network are considered as economic agents competing for link capacity resources. Each VC is equipped with a preference relation defining its individual throughput-delay goals. VCs are endowed with budgets which they use to purchase link resources in a purely competitive economy. The prices of link capacities are iteratively updated by the network nodes in an attempt to compute equilibrium prices in which supply equals demand for all links.

The paper makes several contributions to the study of flow control protocols. First, it presents simple fully decentralized flow control algorithms that allocate capacity to the virtual circuits. The allocation computed reflects the individual throughput delay goals of the diverse VCs as well as their relative priorities. Second, we establish that the allocation of capacity computed by the algorithms is Pareto-optimal and we prove the existence of a feasible, Pareto-optimal operating point. This demonstrates that the algorithms accomplish optimally fair allocation of resources given the set priorities and preferences. Third, experimental evidence is provided that the algorithms converge rapidly.

Applying economic algorithms to problems in computer systems is a growing area of interest. The main contribution of this paper relative to work in this area is demonstrating the power of mathematical economics to provide insight into computer systems that are modelled as economies.

1.0 Introduction

This paper presents an economy that implements flow control in a virtual circuit based computer network. There are two types of agents in the economy. The first are the virtual circuits (VCs). Each VC is characterized by its path (route) through the network, a user defined throughput-delay trade-off and an endowment of money. The VC agent uses it endowment to purchase link resources that optimize its individual throughput-delay goal. The second type of agents are the network nodes. Each network node iteratively updates the prices for the communication links it controls based on the total VC demand and the link capacity. The node′s goal is to compute an equilibrium price at which demand equals supply. We present simple decentralized algorithms for the agents in our economy.

Flow control in a computer network has two goals [3, 8]. The first is obtaining an optimal trade-off between the conflicting goals of maximizing throughput and minimizing delay. Most previous work on flow control has either assumed that all circuits have the same definition of an optimal throughput-delay trade-off or has tried to optimize a system wide throughput-delay goal. (A comparative survey of previous work is given later in this paper). The processing power of network nodes and the bandwidth of network links is consistently increasing. These increases make it possible to support progressively more diverse applications and users. This evolution makes the assumption that all users have the same throughput-delay goal unrealistic. It also becomes difficult, if not impossible, to adequately define a system wide throughput-delay goal that accurately reflects the the heterogeneity of the individual users and their relative priorities. Furthermore, it has been shown that some desirable global optimality criteria cannot be computed by a decentralized algorithm [13].

This work makes several contributions to modeling VC throughput-delay goals in a network supporting diverse users. First, we allow the user of each VC to define an individual throughput-delay goal by specifying a *preference relation* [9] that imposes a partial ordering on the space of resource allocations. Second, we propose a simple preference relation that users can tailor to model the throughput-delay goals of a broad class of applications. Third, we give simple algorithms that network nodes use for updating prices and that VCs use to compute the optimal affordable resource allocation under our preference. Fourth, we prove that the resource allocation computed by our algorithms is *Pareto-optimal* [9]. This is a new definition of optimality for the flow control problem and we compare it with previous definitions. We also use techniques from mathematical economics to prove the existence of a Pareto-optimal operating point for arbitrary networks. Finally, a detailed simulation study demonstrates that our algorithms rapidly converge over a broad range of parameters.

The second goal of flow control is fairness [3, 8, 12]. Optimizing the performance of individual users (or the system as a whole) should not force unfair allocations on other individuals. We present a categorization of fairness metrics and show how the fairness properties of our algorithms fit into this framework. Our algorithms′ properties encompass many previous metrics as special cases.

In the flow control economy, the link controllers set prices based on supply and demand. Link prices can supply important information for new VCs considering entering the system. A new VC knows its throughput-delay goals, and can use the current prices to determine if it can afford to achieve these goals. If not, the VC may either postpone its entry until prices drop or may modify its goals. So, the economy can implement *admission control*. The prices charged for communication resources in the economy can also provide important information for making future circuit routing and capacity planning decisions. These are areas for future research.

A growing body of research has applied concepts from economics to problems in computer systems. A brief survey is presented later in this paper and a detailed survey can be found in [6]. The main contribution of this paper is demonstrating the power of mathematical economics to provide insights into computer system in which there is competition for scarce resources among users with conflicting goals. We demonstrate this by providing practical solutions to a realistic problem.

This paper is structured as follows. Section 2 presents the network model and the economy. This section also describes the agents′ algorithms. Section 3 contains a formalization of the flow control problem and optimality and fairness metrics. In this section, we prove Pareto-optimality of the resource allocations computed by our algorithms. In section 4, mathematical economics is used to prove the existence of a Pareto-optimal operating point for arbitrary networks. Section 5 contains the results of a detailed simulation study demonstrating the rapid convergence of the new algorithms. Section 6 is a comparative survey of previous work on flow control and on applying economic concepts to problems in computer systems. Section 7 contains conclusions and directions for future work. Some lemma used in sections 2, 3 and 4 are proved in an appendix.

2.0 Network and Economy

In this section, we define the network model, the economy and our algorithms. We model the network as a graph $G = \langle V, E \rangle$, where $V = \{1, 2, \ldots, K\}$ is a set of nodes and $E = \{1, 2, \ldots, M\}$ is the set of edges representing the network links. We associate two numbers with each edge $i \in E$: A *capacity* C_i (measured in data units per second), and a *link supply* S_i such that $S_i < C_i$. The link supply sets an upper bound on the link utilization to assure that the queuing delays are bounded. This does not limit generality because S_i arbitrarily close to C_i is allowed. The goal of the flow control economy is to allocate the link supplies among virtual circuits (VCs).

Let $A = \{1, 2, \ldots, N\}$ denote the set of VCs. For now, we assume that A is fixed. We deal with adding and deleting VCs in section 5. For each VC $a \in A$ there is an associated path of edges in G,

$$P_a = l_{a,1}, l_{a,2}, \ldots, l_{a,\, m(a)}, \text{ with } l_{a,\, i} \in E.$$

VCs play the roles of *consumer agents* in the economy and seek allocations of link capacities. An *allocation* of capacity is a vector $\dot{x} = \langle x_1, x_2, \ldots, x_M \rangle$, such that $x_i \geq 0$. Each link $i \in E$ has a *supplier agent* (network node) that charges VCs using i. A *price system* is a vector $\dot{p} = \langle p_1, p_2, \ldots, p_M \rangle$, where $p_i \geq \varepsilon > 0$. We restrict prices to be strictly positive in order to avoid pathological behaviors. The assumption that $p_i > 0$ is used in later sections. This assumption does not limit generality because p_i arbitrarily close to zero is permitted.

Each VC a has a binary *preference relation* [9] denoted $\bullet_a$ over the allocations; $\dot{x} \bullet_a \dot{y}$ denotes that VC a prefers allocation $\dot{x}$ at least as much as allocation $\dot{y}$. If $\dot{x} \bullet_a \dot{y}$ and $\dot{y} \bullet_a \dot{x}$, VC a is *indifferent* between $\dot{x}$ and $\dot{y}$ (denoted $\dot{x} \sim_a \dot{y}$). If $\dot{x} \bullet_a \dot{y}$ but not $\dot{x} \sim_a \dot{y}$, then VC a strictly prefers $\dot{x}$ ($\dot{x} \bullet_a \dot{y}$).

We define $\bullet_a$ to reflect two features of an allocation $\dot{x}$: Its throughput $T(\dot{x})$ and its worst case average delay $D(\dot{x})$. A VC a attempts to maximize $T(\dot{x})$ until it reaches its individual throughput goal γ_a chosen by the VC′s user, and then tries to minimize $D(\dot{x})$. The relation $\bullet_a$ is defined to capture this goal as follows:

1. If $T(\dot{x}) \leq \gamma_a$ and $T(\dot{y}) \leq \gamma_a$, then if $T(\dot{y}) \geq T(\dot{x})$, $\dot{y} \bullet_a \dot{x}$.

2. If $T(\dot{x}) < \gamma_a$ and $T(\dot{y}) \geq \gamma_a$, then $\dot{y} \bullet_a \dot{x}$.

3. If $T(\dot{x}) \geq \gamma_a$ and $T(\dot{y}) \geq \gamma_a$, then if $D(\dot{x}) \geq D(\dot{y})$, $\dot{y} \bullet_a \dot{x}$.

This preference relation can be tailored to model the diversity of throughput-delay goals in a computer network supporting many types of VCs. A circuit supporting a throughput oriented application (e.g. - file transfer) is modeled by setting $\gamma_a = \infty$. The preference relation also models applications with low throughput demands and tight delay constraints such as a VC connecting a workstation to a database. In this case, the throughput goal γ_a is the mean transaction request/response packet size divided by the user's "think time." A VC carrying voice can set γ_a to the value needed to carry the voice signal (64 Kbits/s) and then try to minimize the delay.

This preference relation does not model all situations, however. For example, a VC may have a very strict throughput goal and will be indifferent between all allocations $\dot{x}$ and $\dot{y}$ with $T(\dot{x}) < \gamma_a$ and $T(\dot{y}) < \gamma_a$. Similarly, a VC may have a strict delay constraint and is indifferent between any allocations not meeting the constraint. The economic paradigm deals with these, and other models, but the results of sections 3, 4 and 5 assume the preference relation above.

To compute $\bullet_a$, we use the following definitions for $T(\dot{x})$ and $D(\dot{x})$. The throughput attainable is defined by the link in the path with the smallest allocated capacity, and

$$T(\dot{x}) = \min \{x_i | e_i \in P_a\}.$$

To define $D(\dot{x})$ we assume that VC a only transmits at rate γ_a. Any excess capacity allocated to VC a beyond its throughput goal is unused. VC a hoards the excess capacity to avoid the increase in delay it would experience if this capacity were used by some other VC. Assuming an M/M/1 link queuing model, the worst case average delay on link i is given by

$$\frac{1}{C_i - S_i + (x_i - \gamma_a)}.$$

This assumes that all the remaining link supply $S_i - x_i$ is consumed by the throughput of other users of the link. So, $(C_i - S_i) + (x_i - \gamma_a)$ is the unused link capacity. Under the independence assumption [15],

$$D(\dot{x}) = \sum_{i \in P_a} \frac{1}{C_i - S_i + (x_i - \gamma_a)}.$$

$D(\dot{x})$ is a worst case analysis because other VCs may also be hoarding capacity (effectively lowering utilization). The actual average delay experienced may be lower. The preference relation we are using is *monotonic* [9] in delay and lower delay is always strictly preferred. Dealing with nonmonotonic preferences (i.e. - all delay below some goal D is equal) is an area for further work. In this case, average delay would be more important than worst case average delay. Under the worst case model, a VC always tries to minimize delay whenever possible which will decrease the actual average delay experienced.

At a given price system $\dot{p}$ we associate with each VC a a *demand set* [9] of allocations $\Phi_a(\dot{p})$. We assume that VC a is endowed with a budget $W_a > 0$. The VCs′ endowments are chosen by some policy external to the economy and define the relative priorities of the VCs. Given a price system $\dot{p}$, the set of allocations affordable by VC a is

$$B_a(\dot{p}) = \{\dot{x} \mid \dot{p} \bullet \dot{x} \le W_a\}.$$

This is called the *budget set* [9] of VC a. The demand set is defined as the set of allocations in $B_a(\dot{p})$ that are optimally preferred by $\bullet_a$. That is,

$$\Phi_a(\dot{p}) = \{\dot{x} \mid \dot{x} \in B_a(\dot{p}),\ \text{and if } \dot{y} \in B_a(\dot{p}) \text{ then } \dot{x} \bullet_a \dot{y}\}.$$

2.1.1 Computing the Demand Set

Virtual circuit a wants to send data from its source to its destination at rate γ_a. Since p_i is the cost per unit capacity of edge i, the total cost per transmission unit on a′s path is

$$Q_a = \sum_{i \in P_a} p_i.$$

So, the total cost of achieving its throughput goal is $\gamma_a \bullet Q_a$, and VC a can achieve its throughput goal if $W_a \ge Q_a \bullet \gamma_a$. If this is not the case, VC a attempts to maximize its throughput subject to its budgetary constraint. The following lemma states properties of $\Phi_a(\dot{p})$:

Lemma 2.1 : If $W_a \le Q_a \bullet \gamma_a$ at prices $\dot{p}$, then:

1. $\Phi_a(\dot{p})$ is non-empty.
2. $\Phi_a(\dot{p})$ has a single element $\dot{\phi}^a$.
3. $\phi_i^a = \dfrac{W_a}{Q_a}$, if $i \in P_a$.
4. $\phi_i^a = 0$, if $i \notin P_a$.

This lemma is proved in the appendix. Stated simply, if $W_a \leq Q_a \bullet \gamma_a$, a VC simply divides its wealth over the total cost of the path.

If VC a can afford to meet its throughput goal, it purchases γ_a capacity of each edge in P_a and uses the remaining $W_a - (Q_a \bullet \gamma_a)$ to minimize the worst case average delay of its allocation. We ignore the edges not in P_a for the remainder of this section. Define $u_i = x_i - \gamma_a$, and $K_i = (C_i - S_i)$. The delay minimization problem is to minimize

$$F(\dot{u}) = \sum_{i \in P_a} \frac{1}{K_i + u_i}$$

subject to

1. $u_i \geq 0$.
2. $\dot{p} \bullet \dot{u} \leq W_a - Q_a \bullet \gamma_a$.

The set of feasible $\dot{u}$ is convex and compact, and the function F is continuous and strongly convex on this set. These facts are proved in the appendix. This implies that there exists a unique, feasible $\dot{u}^*$ minimizing $F(\dot{u})$[1] .

Formulating and simplifying the Kuhn-Tucker conditions for this minimization problem gives the following form for the optimal $\dot{u}^*$: For all i,

1. $u_i^* = 0$, or
2.

$$\frac{1}{(K_i + u_i^*)^2 \bullet p_i} = \lambda \qquad (1)$$

where λ is the Lagrange multiplier. The first condition enforces the throughput goal ($u_i^* = x_i - \gamma_a \geq 0$). For all links i with $x_i > \gamma_a$, the second condition states that the marginal decrease in delay per unit price is the same. If this were not the case, buying marginally less of one link and more of another would decrease delay without violating the budget constraint.

The algorithm for minimizing $D(\dot{x})$ computes λ and the links i with $u_i^* > 0$. For $u_i^* > 0$, equation (1) gives

$$u_i = \frac{1}{\sqrt{p_i \bullet \lambda}} - K_i.$$

The algorithm is a binary search on the possible values of λ. A VC agent uses the following algorithm to compute $\dot{u}^*$:

Algorithm 1 :

1. $\lambda^b = 0$ {The lower bound on λ}
2. {An upper bound on λ from equation (1) and $u_i^* \geq 0$}

$$\lambda^u = \max \left\{ \frac{1}{(K_i)^2 \bullet p_i}, i \in P_a \right\}$$

3. $\lambda = (\lambda^u - \lambda^b)/(2.0)$. {Test the midpoint of the range}
4. For all $i \in P_a$ {Compute the u_i}:
   a.

$$u_i = \frac{1}{\sqrt{p_i \bullet \lambda}} - K_i.$$

   b. If $u_i < 0$, $u_i = 0$. {Enforce throughput goal}
5. If $\dot{p} \bullet \dot{u} = W_a - Q_a \bullet \gamma_a$; Stop, $\dot{u}$ is optimal.
6. If $\dot{p} \bullet \dot{u} > W_a - Q_a \bullet \gamma_a$; $\lambda^b = \lambda$. {Too many $u_i > 0$, increase λ}
7. If $\dot{p} \bullet \dot{u} < W_a - Q_a \bullet \gamma_a$; $\lambda^u = \lambda$. {Too many $u_i = 0$, decrease λ}
8. Go to 3 {Test new lambda}

For $p_i \geq \varepsilon = 0.01$ and $K_i = C_i - S_i = 0.8$, algorithm 1 computes λ with maximum error 0.001 in 25 iterations. This algorithm was implemented for the simulation study described in section 5 and this convergence behavior was confirmed.

After computing $\dot{u}^*$, VC a's demand set has a single element $\dot{\phi}^a$ with

1. $\phi_i^a = \gamma_a + u_i^*$, if $i \in P_a$.
2. $\phi_i^a = 0$, if $i \notin P_a$.

This algorithm for computing the optimal affordable allocation is specific to our preference relation. Other optimization algorithms must be defined when other preferences are added to the system. The demand allocations output by the optimization algorithms are reconciled by the price update algorithm described below.

2.1.2 Competitive Equilibrium and Price Updates

The total demand for a resource i at prices $\dot{p}$ is the sum of the individual demands, or

$$d_i(\dot{p}) = \sum_{a \in A} \phi_i^a(\dot{p}),$$

where $\phi_i^a(\dot{p})$ is VC a's demand for link i at prices $\dot{p}$. The *excess demand function* $\dot{Z}(\dot{p})$ [9] represents the imbalance between supply and demand. Its i-th component is simply $Z_i(\dot{p}) = d_i(\dot{p}) - S_i$. The economy is in a *competitive equilibrium* at *equilibrium prices* $\dot{p}^*$ if $Z_i(\dot{p}^*) = 0$ for all i [9]. The flow control economy attempts to compute a competitive equilibrium. Equilibrium has the following properties: 1) No resources are unused. 2) Each VC's demand can be satisfied. Furthermore, we prove in section 3 that the equilibrium allocation is Pareto-optimal.

For the flow control economy, this definition for competitive equilibrium is not adequate because S_i can be strictly greater than $d_i(\dot{p})$ no matter how low p_i falls. Figure 1 on page 9 depicts an example of this phenomenon. There are two VCs with $W_1 = W_2 = 1$ and $\gamma_1 = \gamma_2 = \infty$. All links have $S_i = 1$. By symmetry and lemma 2.1, the VCs evenly divide link 3, and VC 1 gets $\langle \frac{1}{2}, 0, \frac{1}{2} \rangle$ and VC 2 gets $\langle 0, \frac{1}{2}, \frac{1}{2} \rangle$. So, in equilibrium $\dot{Z}(\dot{p}^*) = \langle -\frac{1}{2}, -\frac{1}{2}, 0 \rangle$.

To avoid this problem, we present a new definition of competitive equilibrium. The flow control economy is in equilibrium at prices $\dot{p}^*$ if for all i either: 1) $Z_i(\dot{p}^*) = 0$, or 2) $Z_i(\dot{p}^*) \leq 0$ and $p_i = \varepsilon$. This means that excess supply is allowed if the minimum price is charged for the resource.

Each data link i has a *supplier agent* (network node) that updates the p_i based on supply and demand. The algorithm for updating prices is a *tatonement process* [9] and follows directly from the definition of equilibrium. The change in p_i is proportional to the imbalance between S_i and $d_i(\dot{p})$. The rule for updating p_i is:

$$p_i = \text{Max}\left[p_i + \alpha \bullet p_i \bullet \frac{Z_i(\dot{p})}{S_i}, \varepsilon\right].$$

The parameter α controls the magnitude of the updates. A value of $\alpha = 1$ was used for the experiments of section 5, but in other experiments $\alpha < 1$ was needed to obtain convergence.

The behavior of the flow control economy as a whole is an iteration described by the following:

Algorithm 2 :

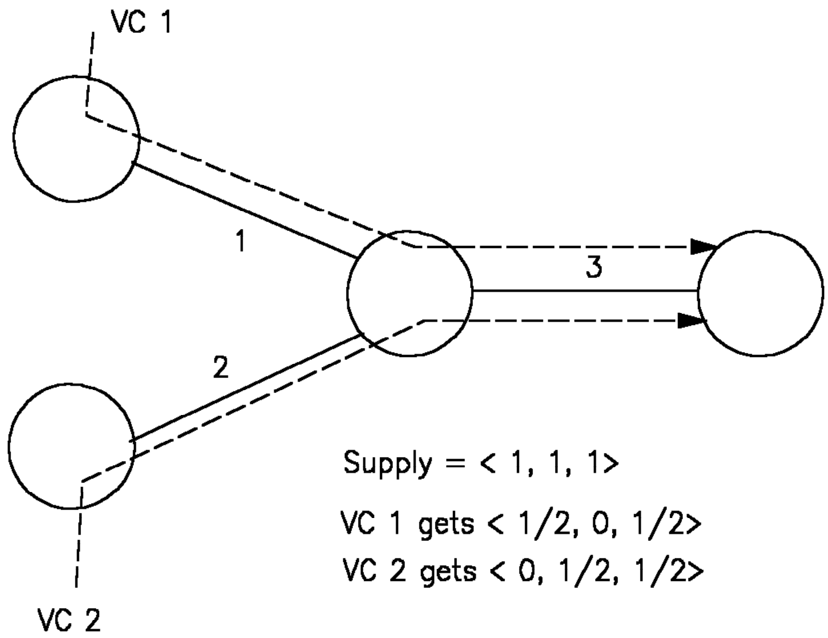

Figure 1. Supply Exceeds Demand at all Prices

---

1. Choose an initial price $\dot{p}$.

2. For all $a \in A$, compute $\dot{\phi}^a(\dot{p})$ {As describe in the preceding subsection}.

3. If for all $i \in E$, $(Z_i(\dot{p}) = 0)$ or $(Z_i(\dot{p}) \leq 0$ and $p_i = \varepsilon)$, an equilibrium has been reached and the iteration stops.

4. Otherwise, for all $i \in E$,

$$p_i = \text{ Max } \left[p_i + \alpha \bullet p_i \bullet \frac{Z_i(\dot{p})}{S_i}, \varepsilon \right].$$

5. Go to 2.

In our implementation, we chose an initial price system with $p_i = \varepsilon$ for all i. The effects of the choice of an initial price system on the equilibrium computed is an open problem in our economy. In step 2, all VCs compute their demand allocations at the current price system in parallel. To perform this computation, each VC requires information about the prices of resources on its path, and no other prices matter. In step 4, the supplier for link i needs to know the identifiers of the circuits using link i and their demands for link i in order to compute $Z_i(\dot{p})$ (in parallel). This means that the VCs can compute their demand vectors and the link suppliers can update prices using a decentralized algorithm as defined by Jaffe [13]. In the simple algorithm above, global information is required to detect equilibrium. We believe that there is a decentralized algorithm for equilibrium detection, but this is left for future work.

3.0 Optimality and Fairness

The goal of this section is to establish the degree to which the flow control economy accomplishes optimal and fair allocation of capacities. We proceed first with a formalized model of the flow control problem and fairness and optimality criteria.

A *flow control problem* is defined by:

1. A graph $G = < V,E>$ with a capacity C_i for each $i \in E$. (As in section 2)
2. A set of virtual circuits $A = \{1, 2, \ldots, N\}$. VC $a \in A$ is characterized by:
    a. $P_a \subseteq E$: VC a's path.
    b. $\mathbf{X}^a$: Set of all possible allocations to a.
    c. An *Admissibility Correspondence*

$$F^a : \left[\prod_{(b \neq a) \wedge (b \in A)} \mathbf{X}^b \right] \rightarrow 2^{\mathbf{X}^a}.$$

For a given set of allocations to the other VCs, F^a defines the allocations that can be assigned to VC a. This correspondence reflects feasibility criteria such as not exceeding link capacity as well as any constraints imposed by system administration, etc.

    d. A preference $\bullet_a$ on $\mathbf{X}^a$.

A *solution* to a flow control problem is a set of allocations $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ with the following property: For all $a \in A$

$$\dot{x}^a \in F^a(\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\} - \{\dot{x}^a\}).$$

The set of allocations is admissible with respect to each VC. For the flow control economy, the admissibility function is simply the budget constraint (and is not directly dependent on the allocations to other VCs).

The flow control problem and solution have been defined. We now present a set of norms that define *fair* solutions. The simplest condition is

(F1) A solution $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ to a flow control problem is fair if for all $a \in A$, $T(\dot{x}^a) > 0$.

This condition is not always satisfied by flow control algorithms that attempt to optimize a global performance metric. For example, in [4] it is shown that maximizing certain global metrics based on power may yield solutions that are not fair by F1. Condition F1 states that no VC should be involuntarily assigned 0 throughput. A VC may decide to terminate if the throughput it is assigned does not meet its requirements.

A stronger fairness condition is to allocate the same throughput to all VCs, but differing link capacities can make this impossible. Jaffe [12] proposed a fairness condition that takes differing link capacities into consideration. A link $i \in P_a$ is a *bottleneck link* for VC a if $T(\dot{x}^a) = x_i^a$. Link i constrains the throughput of VC a. Jaffe′s fairness condition is

> (F2) If link i is a bottleneck link of VC a, a solution $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ is fair if $x_i^a \geq x_i^b$ for all $b \in A$.

This means VC a should get at least as much of its bottleneck link as any other VC. This condition maximizes the minimum throughput and is referred to as *Bottleneck* or *Max-Min* flow control [3, 12]. F2 is based on the assumption that all VCs want to maximize throughput and does not consider delay objectives.

The third definition of fairness is related to a *Nash Equilibrium* [30].

> (F3) A solution $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ is fair if for all $a \in A$, $\dot{x}^a \bullet_a \dot{y}$ for all $\dot{y} \in F(\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\} - \{\dot{x}^a\})$.

This simply states that a given allocation is fair if no VC can unilaterally improve its allocation by choosing another admissible allocation given the allocations to the other VCs. This approach to fairness and optimality was proposed by Hsiao and Lazar [10, 11]. Unlike F1 and F2 which assume that all VCs are throughput oriented, F3 allows each VC to choose its individual preference relation, which may include both throughput and delay.

F3 is pessimistic because it does not allow VC cooperation. F3 states that a solution is fair if no single VC can improve on its allocation. The fourth condition states that a solution is fair if no *group* of VCs can cooperatively improve their allocations. To formalize cooperation, we borrow from Microeconomics. Define a *coalition* S as a subset of A. S can improve on solution $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ if there is another solution $\{\dot{y}^1, \dot{y}^2, \ldots, \dot{y}^N\}$ with

1. $\dot{y}^a = \dot{x}^a$, $a \notin S$ (The non-members are not affected).

2. $\dot{y}^a \bullet_a \dot{x}^a$, $a \in S$ (No one in the coalition is worse off).

3. There exists $a \in S$ with $\dot{y}^a \bullet_a \dot{x}^a$ (At least one member is strictly better off).

Given this definition,

> (F4) A solution $\{\dot{x}^1, \dot{x}^2, \ldots, \dot{x}^N\}$ is fair if no coalition can improve on their allocations.

Condition F4 is the definition of *Pareto-optimality* [9] and it also permits individual preferences that include throughput and delay.

The sets of fair solutions defined by these conditions are not disjoint. The following propositions state two obvious relations:

Proposition :

1. Condition F2 implies condition F1.
2. Condition F4 implies condition F3.

The fairness condition supplied by the flow control economy is based on the price system $\dot{p}$. The feasibility operator F^a is defined by the budget set. The following lemma shows that the flow control economy computes solutions that are fair by F1 and F3:

Lemma 3.1 : Let (1) $\dot{p}$ be a price system, (2) $\dot{\phi}^a(\dot{p})$ be VC a's demand at $\dot{p}$, (3) $\dot{x} \neq \dot{\phi}^a(\dot{p})$ be any allocation. Then,

1. $\dot{\phi}^a(\dot{p}) \bullet \dot{p} = W_a$.
2. $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$ implies that $\dot{x} \bullet \dot{p} \geq \dot{\phi}^a(\dot{p}) \bullet \dot{p}$.
3. $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$ implies that $\dot{x} \bullet \dot{p} > \dot{\phi}^a(\dot{p}) \bullet \dot{p}$.

The lemma is proved in the appendix.

By definition, the link prices and W_a are non-zero. The fairness of $\dot{\phi}^a(\dot{p})$ by condition F1 follows from part 1 of the lemma; Condition F3 follows from parts 2 and 3.

Lemma 3.1 can be used to prove that the flow control economy computes solutions that are fair by F4. The following theorem proves this assertion:

Theorem 3.1 : Let $\dot{p}$ be an equilibrium price in the flow control economy. Then, the set of allocations

$$\dot{\phi}^1(\dot{p}),\ \dot{\phi}^2(\dot{p}),\ \dots,\ \dot{\phi}^N(\dot{p})$$

is pareto-optimal and unique.

Proof :

(unique) We showed in section 2 that each $\dot{\phi}^i(\dot{p})$ is unique. So the aggregation of the choices in unique.

(Pareto-optimality) Assume a coalition $S = \{1, 2, \dots, s\}$ chooses a set of allocation with

1. $\dot{\phi}'^i(\dot{p}) \ \bullet_i \ \dot{\phi}^i(\dot{p}),\ \forall i \in S.$
2. $\dot{\phi}'^j(\dot{p}) \ \bullet_j \ \dot{\phi}^j(\dot{p})$, for at least 1 $j \in S$.

We know from lemma 3.1 that

1. $\dot{\phi}'^i(\dot{p}) \bullet \dot{p} \geq \dot{\phi}^i(\dot{p}) \bullet \dot{p},\ \forall i \in S.$
2. $\dot{\phi}'^j(\dot{p}) \bullet \dot{p} > \dot{\phi}^j(\dot{p}) \bullet \dot{p}$, for j.

So,

$$\sum_{i=1}^{s} \dot{\phi}'^i(\dot{p}) \bullet \dot{p} > \sum_{i=1}^{s} \dot{\phi}^i(\dot{p}) \bullet \dot{p} = \sum_{i=1}^{s} W_i.$$

The coalition cannot afford the better set of allocations at prices $\dot{p}$.

This proof only relies on the properties of the preference relation established in lemma 3.1. Theorem 3.1 is not specific to the preference relation of section 2. Any other preference relation possessing these properties could be added to the flow control economy without decreasing the fairness properties or Pareto-optimality of the equilibrium allocation.

The equilibrium allocation is optimal because an allocation that improves the lot of any VCs forces unfair allocations on others. This depends on the assumption that the tatonement process sets prices that accurately reflect the relative importance of the data links. It is intuitively accepted that prices in an economy accurately reflect the relative value of resources. Formally analyzing the relationship between prices, demand and wealth in the flow control economy is an important area for continuing work.

In the proof of theorem 3.1, we have used the budget constraint as the admissibility correspondence. If instead we use not exceeding the link supplies as our definition of feasible, and ignore pricing, budgets and endowments, it is easy to show that the allocations are still Pareto-optimal. However, uniqueness no longer holds.

The examples in Figure 2 on page 16 show the relationship between conditions F2 and F4. In example 1, assume VC 1 and 2 use only throughput for their preference ($\gamma_a = \infty$), and consider the allocations $<3, 0>$ to VC 1 and $<2, 1>$ to VC 2. VC 2 can increase VC 1′s throughput by giving it one unit of capacity on link 1. This does not decrease VC 2′s throughput because $S_2 = 1$. So, the allocation is not Pareto-optimal. It is fair by definition F2 because VC 1 gets more of link 1 than VC 2. Jaffe′s algorithm computes a solution that is more fair than he claims. The solution is in fact fair by F4. Other definitions of fairness equivalent to F2 include special rules to ensure that the solution is also Pareto-optimal [3]. The main advantage of the flow control economy over this previous work is that diverse preferences are allowed. The previous work assumes that all VCs are throughput oriented.

Example 2 in Figure 2 on page 16 shows that the flow control economy computes solutions that are not fair by F2. In this example, for $i = 1, 2, 3$, $\gamma_i = 1$, and $W_i = 1$. At prices $\dot{p} = <1.5, 1.5>$, the economy is in equilibrium with allocations $< \frac{2}{3}, 0>$ to VC 1, $<0, \frac{2}{3}>$ to VC 2, and $< \frac{1}{3}, \frac{1}{3}>$ to VC 3. VC 3 is bottlenecked on links 1 and 2 and has less capacity than the other VCs. Bottleneck flow control tries to equalize throughputs while the flow control economy tries to equalize the dollar values of the allocations (assuming equal wealth).

This example shows that the flow control economy can be unfair to long VCs under definition F2. An alternative argument can state that condition F2 is unfair to short VCs. Under F2, the total amount of capacity assigned to a VC increases with its length. This decreases the amount assigned to shorter VCs. External policies can cause the flow control economy to reach a mean between these conflicting goals by the amount of money assigned to long VCs compared to short VCs.

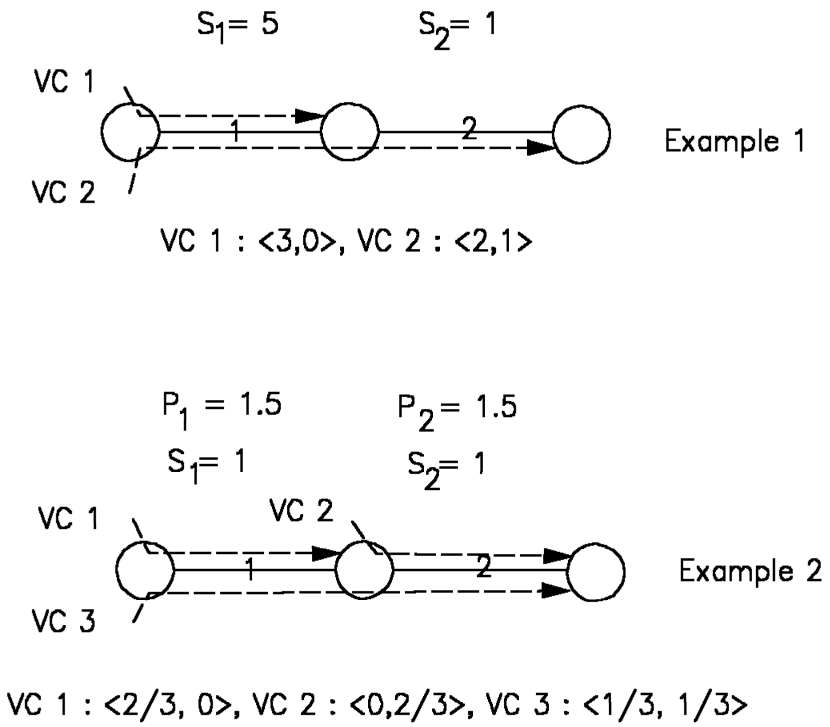

Figure 2. Fairness Conditions F2 and F4

4.0 Existence of Equilibrium

The preceding section demonstrated the optimality and fairness conditions of the equilibrium allocation. In this section we show that it is meaningful to try to compute an equilibrium by proving that a competitive equilibrium always exists. We also show that there may be infinitely many equilibrium price vectors. The traditional approach in economic theory for proving the existence of an equilibrium price vector is based on fixed point theorems [1, 2, 9]. A point $\dot{p}^0$ is a fixed point of a function $\dot{f}$ if $f(\dot{p}^0) = \dot{p}^0$. A function $\dot{f}(\dot{p}) = \dot{p} + \dot{Z}(\dot{p})$ is defined and shown to have a fixed point $\dot{p}^0$. This implies that $\dot{Z}(\dot{p}^0) = \dot{0}$ and $\dot{p}^0$ is an equilibrium price vector.

We must modify these tools to prove existence of equilibrium in the flow control economy because of the non-standard definition of equilibrium that may have $\dot{Z}(\dot{p}^*) \neq \dot{0}$ in equilibrium. Economic theory also assumes that the demand functions of the individual agents are continuous with $\dot{p}$. We cannot make this assumption, and we prove that the demand function defined in section 1 is continuous.

We define a new function $\dot{H}(\dot{p})$ with the property that $\dot{H}(\dot{p}) = \dot{0}$ if, and only if, $\dot{p}$ is an equilibrium price system. We then show that $\dot{f}(\dot{p}) = \dot{p} + \dot{H}(\dot{p})$ has a fixed point.

We define the function $\dot{H}(\dot{p})$ with i-th component

$$H_i(\dot{p}) = (p_i - \varepsilon) \bullet Z_i(\dot{p}) + Z_i(\dot{p}) + |Z_i(\dot{p})|.$$

By our definition of equilibrium, if $\dot{p}^*$ is an equilibrium price vector, then for every i

1. $Z_i(\dot{p}^*) = 0$, or
2. $p_i = \varepsilon$ and $Z_i(\dot{p}^*) \leq 0$.

If $Z_i(\dot{p}^*) = 0$, then $H_i(\dot{p}^*) = 0$. If $p_i = \varepsilon$, then the first term in $H_i(\dot{p}^*)$ is 0. If $Z_i(\dot{p}^*) \leq 0$, then the second and third terms cancel and $H_i(\dot{p}^*) = 0$. So, if $\dot{p}^*$ is an equilibrium price, $H_i(\dot{p}^*) = 0$ for all i, which implies $\dot{H}(\dot{p}^*) = \dot{0}$.

Assume that $\dot{H}(\dot{p}) = \dot{0}$. We must show that $\dot{p}$ is an equilibrium price vector. We know by definition that $p_i - \varepsilon \geq 0$. So, $H_i(\dot{p}) = 0$ implies that $Z_i(\dot{p}) \leq 0$. This, in turn, implies that

$$H_i(\dot{p}) = (p_i - \varepsilon) \bullet Z_i(\dot{p}).$$

This term is 0 only if $Z_i(\dot{p}) = 0$ or $p_i = \varepsilon$. So, $H_i(\dot{p}) = 0$ implies that $Z_i(\dot{p}) = 0$ or $Z_i(\dot{p}) \leq 0$ and $p_i = \varepsilon$. The following theorem has been shown.

Theorem 4.1 : $\dot{H}(\dot{p}) = \dot{0}$ if, and only if, $\dot{p}$ is an equilibrium price vector.

Define the function $\dot{f}(\dot{p}) = \dot{p} + \dot{H}(\dot{p})$. We apply the following theorem to show that $\dot{f}$ has a fixed point.

> (Brouwer′s fixed point) **Theorem :** Let X be a compact, convex subset of $\bullet^{M}$, and let $\dot{f}$ be a continuous function mapping X into X. Then there exists $\dot{x}^0 \in X$ such that $\dot{f}(\dot{x}^0) = \dot{x}^0$.

Brouwer′s theorem is discussed in economic contexts in [1, 2, 9].

To apply the fixed point theorem, we must

1. Define the set X.
2. Prove $\dot{f}$ maps X into itself.
3. Prove continuity of $\dot{f}$ on X.

Let $W_T = \sum_{i=1}^{N} W_i$. This is the total amount of money possessed by the VCs. Define the set X as

$$X = \left\{ \dot{x} \mid \varepsilon \le x_i \le \frac{2W_T}{\varepsilon} \right\}.$$

X is simply a cube in $\bullet^{M}$ and contains its boundary. So, it is is compact and convex.

We must show that $f(\dot{p}) \in X$ for all $\dot{p} \in X$. To do so, we make two assumptions that do not limit generality. First, we assume that $0 \le S_i \le 1$ for all i. If this is not the case, an equivalent economy can be created by dividing all S_i by Max $\{S_i \mid 1 \le i \le M\}$. Secondly, we assume that $\varepsilon \le S_i$ for all i. This is possible because the only constraint is $\varepsilon > 0$.

Since $p_i \ge \varepsilon > 0$, f_i takes its minimal value at $\dot{p}^0$ when $Z_i(\dot{p}^0)$ is minimal. Our assumptions that $S_i \le 1$ implies that $Z_i(\dot{p}) \ge -1$, and we have

$$f_i(\dot{p}^0) \ge p_i^0 + (\dot{p}_i^0 - \varepsilon) \bullet (-1) + (-1) + |-1| = \varepsilon.$$

To prove that $f_i(\dot{p}) \le \frac{2W_T}{\varepsilon}$ we define a function

$$g(p_i) = p_i + (p_i - \varepsilon)\frac{W_T}{p_i} + \frac{2W_T}{p_i}.$$

The budget constraint implies that $Z_i(\dot{p}^0) \le \frac{W_T}{p_i^0} - S_i$. We know that $S_i > 0$ and we have $g(p_i^0) \ge f_i(p_i^0)$. The second derivative of $g(\dot{p})$ is

$$g''(\dot{p}) = \frac{2(2-\varepsilon)W_T}{p_i^3},$$

which is strictly positive on $[\varepsilon, \frac{2W_T}{\varepsilon}]$ provided that $\varepsilon < 2$. So, $g(p_i)$ takes its maximal value at either $p_i = \varepsilon$ or $p_i = \frac{2W_T}{\varepsilon}$. At $p_i = \varepsilon$,

$$f_i(\dot{p}^0) \leq \varepsilon + \frac{2W_T}{\varepsilon} - 2S_i \leq \frac{2W_T}{\varepsilon}.$$

At $p_i^0 = \frac{2W_T}{\varepsilon}$, $Z_i(\dot{p}^0) \leq 0$ because of the budget constraint and

$$f_i(\dot{p}^0) \leq \frac{2W_T}{\varepsilon}.$$

This shows that for all $\dot{p} \in X$, $\varepsilon \leq f_i(\dot{p}) \leq \frac{2W_T}{\varepsilon}$ for all i which means that $\dot{f}(\dot{p}) \in X$. So, $\dot{f}$ maps X into itself.

The function $\dot{H}(\dot{p})$ is continuous if the excess demand function $\dot{Z}(\dot{p})$ is continuous. The excess demand function is simply

$$\dot{Z}(\dot{p}) = \left(\sum_{i=1}^{N} \dot{\phi}^i(\dot{p})\right) - \dot{S},$$

where $\dot{\phi}^a(\dot{p})$ is VC a's demand at $\dot{p}$ and $\dot{S}$ is the vector of link supplies. We will show that $\dot{\phi}^a(\dot{p})$ is continuous, which implies continuity of $\dot{Z}$.

Lemma 2.1 shows that for VC a, if

$$\gamma_a \sum_{j \in P_a} p_j \geq W_a$$

the agent's demand is a continuous function of $\dot{p}$. If the agent can afford its throughput goal, and attempts to minimize delay, we have to show that the $\dot{u}^*$ solving the minimization problem in the previous section is a continuous function of $\dot{p}$.

Let $U(\dot{p})$ be defined as follows: $\dot{u} \in \bullet^M$ is in $U(\dot{p})$ if,

1. $u_i \geq 0$

2.

$$\sum_{j \in P_a} (p_i \bullet u_i) \leq W_a - (\gamma_a) \sum_{j \in P_a} p_i.$$

Define the function $F{:}U(\dot{p}) \rightarrow \bullet$ as

$$F(\dot{u}) = \sum_{i \in P_a} \frac{1}{(K_i + u_i)}$$

as in section 2.

The following lemmas are needed to prove continuity and are proved in this chapter′s appendix.

Lemmas 4 :

1. For any $\dot{p}$, F is *continuous* and *strongly convex* on $U(\dot{p})$.
2. For any $\dot{p}$, $U(\dot{p})$ is *compact* and *convex*.
3. A sequence of points $\{\dot{x}^m\}$ in $\bullet^N$ converges to $\dot{x}^0$ if, and only if, every sub sequence $\{\dot{x}^{m_k}\}$ contains a subsequence $\{\dot{x}^{m'_k}\}$ that converges to $\dot{x}^0$.

We prove the following theorem:

Theorem 4.2 : For all $\dot{p}$, let $\dot{x} = M(\dot{p})$ denote a vector in $U(\dot{p})$ minimizing $F(\dot{x})$. Then $M(\dot{p})$ is a continuous function of $\dot{p}$.

Proof:

1. $M(\dot{p})$ is a *function*:
   a. $U(\dot{p})$ is compact, and F is continuous on $U(\dot{p})$, so there exists an $\dot{x} \in U(\dot{p})$ minimizing F [1].
   b. So $M(\dot{p})$ is defined for all $\dot{p}$.
   c. Furthermore, *convexity* of $U(\dot{p})$ and *strong convexity* of F on $U(\dot{p})$ imply that $\dot{x}$ is *unique* for any $\dot{p}$.
   d. So, M is not multivalued and is a function.

2. $M(\dot{p})$ is *continuous*:

a. Pick a point $\dot{p}^0$. Let $\dot{p}^m$ be a sequence converging to $\dot{p}^0$.

b. Denote $\dot{x}^m = M(\dot{p}^m)$ and $\dot{x}^0 = M(\dot{p}^0)$.

c. We must show that as $\dot{p}^m \to \dot{p}^0$, $\dot{x}^m \to \dot{x}^0$.

d. Define two sequences $\dot{w}^m$ and $\dot{u}^m$ with the following properties:

1) $\dot{w}^m$:

a) $\dot{w}^m \in U(\dot{p}^m)$.

b) For all $\dot{z} \in U(\dot{p}^m)$,

$$||\dot{w}^m - \dot{x}^0|| \leq ||\dot{z} - \dot{x}^0||.$$

$\dot{w}^m$ is the point in $U(\dot{p}^m)$ closest to $\dot{x}^0$.

2) $\dot{u}^m$:

a) $\dot{u}^m \in U(\dot{p}^0)$.

b) For all $\dot{y} \in U(\dot{p}^0)$,

$$||\dot{u}^m - \dot{x}^m|| \leq ||\dot{y} - \dot{x}^m||.$$

$\dot{u}^m$ is the point in $U(\dot{p}^0)$ closest to $\dot{x}^m$.

$\dot{u}^m$ and $\dot{w}^m$ exist because $U(\dot{p}^m)$ and $U(\dot{p}^0)$ are compact. Figure 3 on page 22 shows an example of the sequences $\dot{x}^m$, $\dot{w}^m$, $\dot{u}^m$ and their relationship to $\dot{p}^0$ and $\dot{x}^0$ in the two dimensional case.

e. As $\dot{p}^m \to \dot{p}^0$,

1) $||\dot{w}^m - \dot{x}^0|| \to 0$

2) $||\dot{u}^m - \dot{x}^m|| \to 0$.

This is intuitively clear. It is formally proved in an appendix.

f. Let $\dot{u}^{m_k}$ be a subsequence of $\dot{u}^m$. Since $U(\dot{p}^0)$ is compact, $\dot{u}^{m_k}$ has a convergent subsequence $\dot{u}^{m'_k} \to \dot{u}^0 \in U(\dot{p}^0)$.

g. Assume that $\dot{u}^0 \neq \dot{x}^0$.

h. By construction $||\dot{u}^m - \dot{x}^m|| \to 0$. By f., $\dot{u}^{m'_k} \to \dot{u}^0$. So, $\dot{x}^{m'_k} \to \dot{u}^0$.

i. $\dot{w}^m \to \dot{x}^0$ which implies that $\dot{w}^{m'_k} \to \dot{x}^0$.

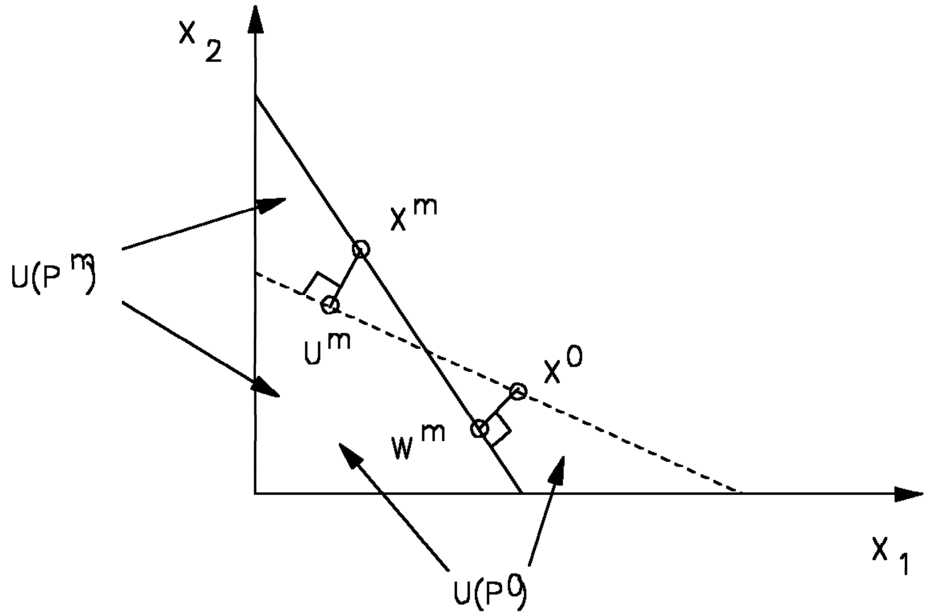

Figure 3. The two dimensional example: $\dot{u}^m$ is the point in $U(\dot{p}^0)$ closest to $\dot{x}^m$. $\dot{w}^m$ is the point in $U(\dot{p}^m)$ to $\dot{x}^0$.

j. By definition, $F(\dot{w}^{m'k}) \geq F(\dot{x}^{m'k})$.

k. By the assumption that $\dot{u}^0 \neq \dot{x}^0$ and uniqueness of the optimal, $F(\dot{u}^0) > F(\dot{x}^0)$.

l. By continuity of F, $F(\dot{x}^{m'k}) \to F(\dot{u}^0)$, and $F(\dot{w}^{m'k}) \to F(\dot{x}^0) < F(\dot{u}^0)$. However, $F(\dot{w}^{m'k}) \geq F(\dot{x}^{m'k})$ implies that $F(\dot{x}^0) \geq F(\dot{u}^0)$ which contradicts $F(\dot{x}^0) < F(\dot{u}^0)$

m. So, $\dot{u}^0 = \dot{x}^0$ and every subsequence of $\{\dot{u}^m\}$ contains a subsequence that converges to to $\dot{x}^0$, which by the lemma above means $\dot{u}^m \to \dot{x}^0$.

n. $||\dot{u}^m - \dot{x}^m|| \to 0$, and by m., $\dot{x}^m \to \dot{x}^0$.

3. This proves the theorem.

Lemma 2.1 shows that $\dot{\phi}^a(\dot{p})$ is continuous at any $\dot{p}$ with $\gamma \sum_{i \in P_a} p_i \geq W_a$. Theorem 4.2 shows that $\dot{\phi}^a(\dot{p})$ is continuous at $\dot{p}$ when $\gamma \sum_{a \in P_a} p_i \leq W_a$. When $\gamma \sum_{i \in P_a} = W_a$, the feasible vector minimizing F is $\dot{0}$. This implies that $\dot{\phi}^a(\dot{p})$ is continuous when $\gamma \sum_{i \in P_a} = W_a$. So, the VCs demand is continuous at all prices.

The demand functions of the individual VCs are continuous, which means that $\dot{H}$ is continuous. Brouwer's fixed point theorem can be applied to prove that $f(\dot{p}) = \dot{p} + \dot{H}(\dot{p})$ has a fixed point. This proves the main result of this section stated by the following theorem.

Theorem 4.3 : There is an equilibrium price vector $\dot{p}^*$ for the flow control economy.

The theorem states that there is at least 1 equilibrium price vector. The following example demonstrates that there may be infinitely many equilibrium price vectors. Figure 4 on page 24 depicts the example. There are two links with $S_1 = S_2 = 1$ and one VC using the links. The VC has $W = \$2$ and $\gamma = \infty$. At any price vector with $p_1 + p_2 = 2$, the VC demands $\langle 1, 1 \rangle$ by lemma 2.1. In this case, supply equals demand for both links and the economy is in equilibrium. So, there are infinitely many equilibrium prices.

This example illustrates that there are infinitely many equilibrium *prices*. However, in this example there is only 1 equilibrium allocation. We have not yet determined if all equilibria prices yield the same allocations, or if there are multiple allocations as well. This is left for future work.

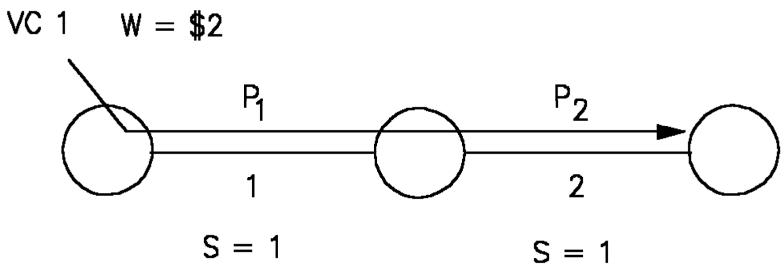

Figure 4. Infinite Equilibria: If $p_1 + p_2 = 2$, the economy is in equilibrium.

5.0 Convergence to Equilibrium

This section presents the results of experiments to determine the convergence behavior of the economy (Algorithm 2). In general, the tatonement process in an economy does not always converge to an equilibrium [27]. Economists have developed algorithms for computing equilibrium price vectors, but these algorithms are extremely complicated, have high computational overhead and are inherently centralized [27]. We conducted several hundred simulation experiments on algorithm 2, and the flow control economy always converged. Exploiting special properties of this specific economy to prove convergence or finding a non-converging example is left for future work.

We implemented the flow control economy and simulated its behavior on randomly generated networks. In each simulation, the network had 50 virtual circuits and 25 data links. The capacity of link i, C_i, was a random variable with uniform distribution on the interval [1, 5]. S_i was $0.8 \times C_i$ for all links.

Figure 5 on page 26 and Figure 6 on page 27plot the convergence to equilibrium of the flow control economy for various network definition parameters. The X-axis represents the number of iterations of the economy (the number of price updates in step 4 of algorithm 2). The Y-axis represents the relative distance from the eventual equilibrium allocation of resources. Denote $\dot{d}(\dot{p}^k)$ as the total demand vector for all agents at price $\dot{p}^k$ of iteration k; $\dot{d}(\dot{p}^*)$ is the total demand in equilibrium. In this notation, the Y-axis represents

$$\frac{||\dot{d}(\dot{p}^*) - \dot{d}(\dot{p}^k)||}{||\dot{d}(\dot{p}^*)||}.$$

The Y-axis represents the normalized error between $\dot{d}(\dot{p}^k)$ and $\dot{d}(p^*)$.

In Figure 5 on page 26, the average convergence for three sets of random economies is presented. These economies differ by the probability of link i being in agent a's path. If this probability is denoted $Pr\{i \in P_a\}$, the three curves plot average convergence for random economies with $Pr\{i \in P_a\} = 0.1, 0.2,$ and 0.3 for all i and a. These values define random economies in which the mean length of a virtual circuit is 2.5, 5.0, 7.5, respectively. The mean VC throughput goal for these experiments was uniformly distributed on the interval $[0, 2]$. The first conclusion drawn from this graph is that the economy converges very rapidly. After 15 iterations, all three economies are within 2% of the equilibrium allocation. The second conclusion is that the mean VC length has little effect on the rate of convergence. The explanation for this fact is given below.

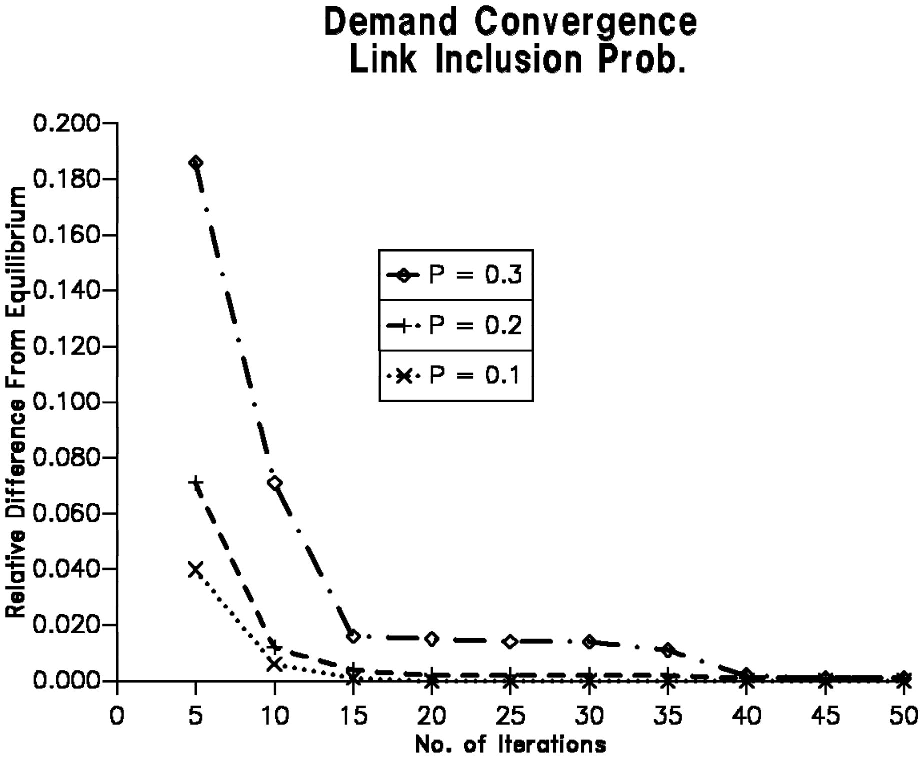

Figure 5. Convergence to Equilibrium Demand: This figure plots the economy's convergence to equilibrium. The three curves represent random economies in which the probability of link i being in VC *a*'s path is 0.1, 0.2, and 0.3. These yield mean VC lengths of 2.5, 5 and 7.5 links respectively. The mean VC throughput demand was 1.0.

Figure 6 on page 27 plots the convergence behavior of the economy for various VC throughput goals. In this experiment, each virtual circuit's throughput demand γ_a was a random variable uniformly distributed on the interval $[0,T]$. The three curves represent the cases of T = 0.5, 2.0, and 3.0. In these economies, the mean VC length was 5 links. The same conclusion can be drawn from these experiments as from the previous. The economy converges rapidly and after 20 iterations all three economies are within 2% of the equilibrium allocation. Varying the mean throughput goal has little effect on the convergence behavior of the economy.

The flow control economy converges very rapidly to the equilibrium allocation of capacity to agents. The economy takes many more iterations to identify that an equilibrium has been computed, however. In the simple economy of Figure 7 on page 28, there are two VCs (VC 1 and VC 2). $P_1 = 1,3$ and $P_2 = 2,3$, and both VCs have $W = 1$ and $\gamma = 1$. The supply vector is $\dot{S}$ = <1, 1, 1.98>. By symmetry and lemma 2.1, the equilibrium allocation for this economy is:

1. VC 1 gets <0.99, 0, 0.99>.

2. VC 2 gets <0, 0.99, 0.99>.

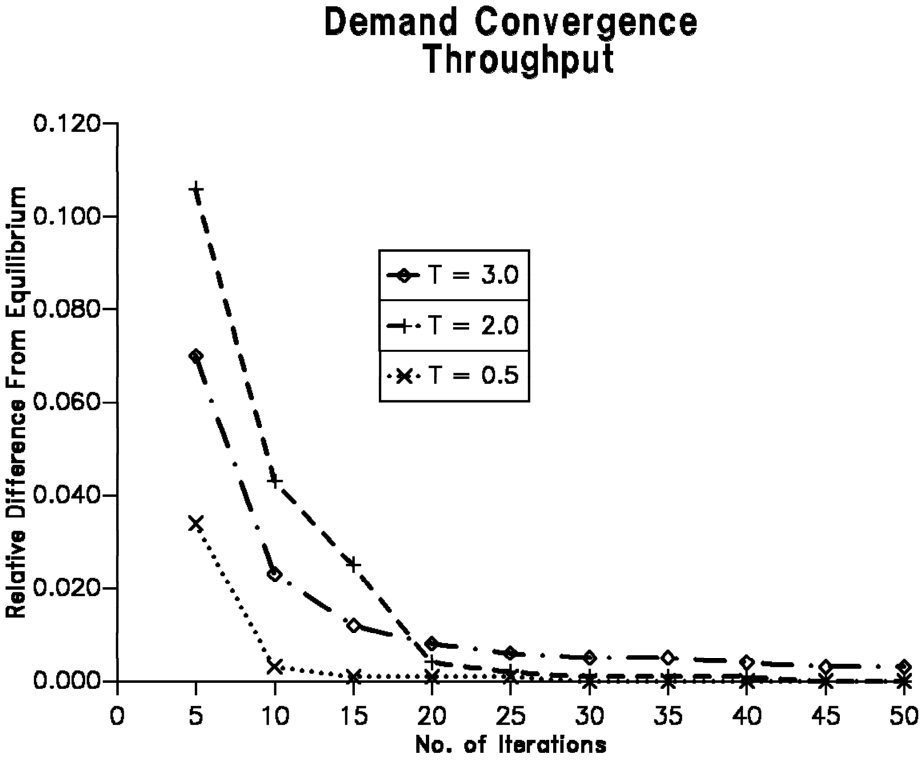

Figure 6. Convergence to Equilibrium Demand: This figure plots the economy's convergence to equilibrium. The three curves represent random economies in which the mean VC throughput demand was uniformly distributed in the interval [0, *T*], where T = 0.5, 2.0, and 3.0. The mean VC length was 5 links.

In the simulation of this simple economy, the minimum price was $\varepsilon = 0.001$. The economy required over 1000 iterations to detect that it had computed an equilibrium. The economy *did* converge very rapidly to the equilibrium, but did not realize that it had. After only four iterations,

$$\frac{\|\dot{d}(\dot{p}^4) - \dot{d}(\dot{p}^*)\|}{\|\dot{d}(\dot{p}^*)\|} = 0.002.$$

The detection problem is caused by the fact that in equilibrium some resource may have demand slightly less than supply. In this case, the price of the resource is slowly driven down to ε. Let i be the most utilized link whose equilibrium demand is less than supply. That is, the i that minimizes:

$$\Delta = \frac{(S_i - d_i(\dot{p}^*))}{S_i}$$

and has $p_i^* = \varepsilon$. Suppose that the equilibrium allocation for this link is computed in iteration L. The economy must drive the price of edge i to ε. The price p_i^K in iteration $K > L$ is simply

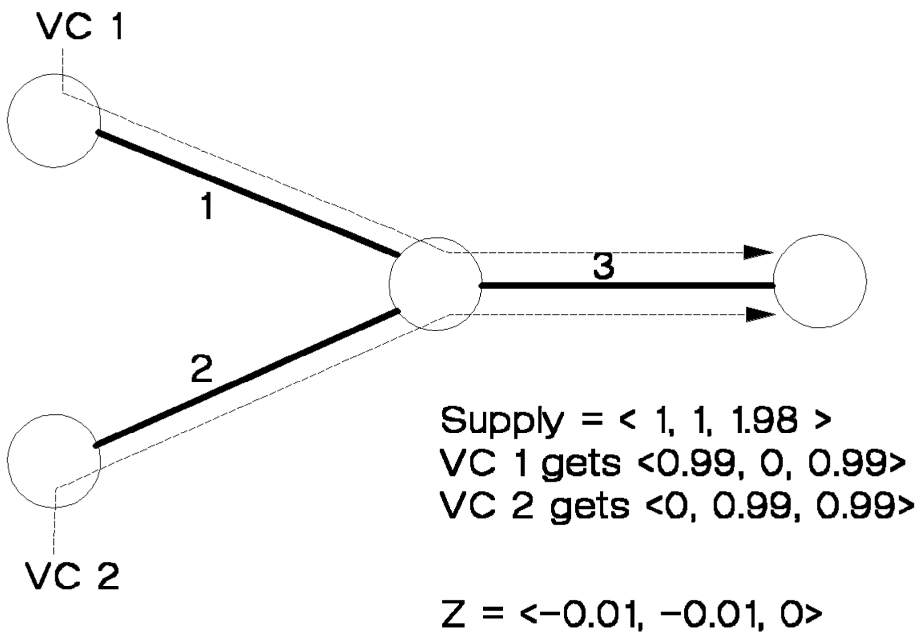

Figure 7. Convergence detection problem.

$$p_i^L \bullet (1 - \Delta)^{(K - L)}.$$

For $p_i^K \leq \varepsilon$, we have

$$K = L + \frac{\log \varepsilon - \log p_i^L}{\log(1 - \Delta)}. \tag{2}$$

As $\Delta \to 0$, the value of K in equation (2) goes to infinity. Factors such as mean VC throughput goal and mean VC length only indirectly effect the detection of equilibrium. The detection of equilibrium is solely determined by the most utilized, non-saturated link.

There are heuristic solutions to the equilibrium detection problem. One approach is to arbitrarily stop the economy after some threshold K iterations. Our experiments suggest that $K = 25$ yields allocations within 2% of optimal for realistically large networks. Another approach is to monitor the change in demand in successive iterations. In this case, the economy stops iterating if

$$\frac{\|\dot{d}(\dot{p}^{k+1}) - \dot{d}(\dot{p}^{k})\|}{\|\dot{d}(\dot{p}^{k})\|} \leq \delta,$$

where δ is a parameter. We implemented and tested this second heuristic for the economies of Figure 5 on page 26, Figure 6 on page 27 and Figure 7 on page 28, setting δ = 0.01. The heuristic detects equilibrium in the pathological economy of Figure 7 on page 28 after 2 iterations. The relative error in the demand vector when the heuristic guesses that equilibrium was found was 0.002. For the six sets of random economies of Figure 5 on page 26 and Figure 6 on page 27, on the average the heuristic guesses that equilibrium has been found after 10.1 iterations. The average relative error if the economy stopped when equilibrium was guessed was 0.021.

5.1.1 Network Updates

The simulation experiments in the preceding discussion demonstrated that the flow control economy can rapidly compute an equilibrium allocation of resources when starting from an initial state in which no resources have been allocated. These experiments modelled network start-up with prebuilt routing tables between source and destination pairs. An equally interesting situation is the dynamic behavior of the network between start-ups. In this section, we study the convergence of the economy to a new equilibrium when a VC leaves the economy, or when a new VC is started up. Figure 8 on page 30 contains the results of simulation experiments testing the convergence. In this figure, a network with 25 links and 50 random VCs is generated and the equilibrium is computed. Then VC additions and deletions are simulated. After each change, the new equilibrium is computed. The curves represent the average number of iterations required to reach the new equilibrium. The three cases are: 1) Adding 50 VCs, 2) Deleting 50 VCs, and 3) a random ordering of 50 additions and deletions.

Three main conclusions can be drawn from these experiments. First, the economy rapidly computes the new equilibrium after an update. The economy is within 5% of the new equilibrium after 15 iterations in all cases. The second conclusion is that moving from one equilibrium to the new equilibrium after an update requires approximately as many iterations as the computation of the initial equilibrium. This is due to the same phenomenon described above. The third conclusions is that the flow control economy can effectively deal with dynamic network environments.

The flow control economy can also implement *admission control*. A user planning to activate a new VC can examine the link prices to determine the quality of service it can afford given its endowment. The user can then scale down its goals or defer entry into the system if it is too poor

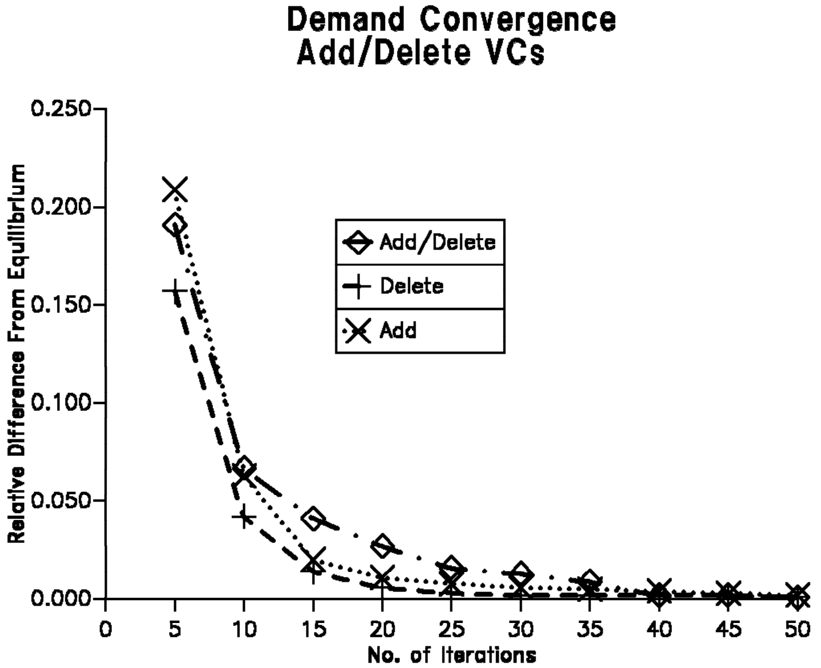

Figure 8. Network Updates: This figure plots the average convergence of the economy to the new equilibrium after a new VC is activated or a VC terminates. The three curves represent adding 50 VCs 1 at a time, deleting 50 VCs and a random ordering of 50 adds and deletes.

to meet its performance objectives. There are two disadvantages to this approach, however. First, if the user decides to enter the system, resource demands increase but the supply is fixed. This means that prices rise when the user enters. So, the actual quality that can be purchased is less than initially estimated. The second disadvantage occurs because the price increases may prevent other users already in the system from continuing to meet their objectives. Several users may leave the system even though if only one user terminated the remaining users could meet their goals. Enhancing our protocols to deal with these problems is an area for future work.

In the simulations in this section, the path chosen for a new VC is determined before the economy starts iterating to a new equilibrium. The price information can help the VC agent choose the "best" path. However, the VC cannot change its routing decision after the tatonement process begins. That is, dynamic routing after each price update cannot be allowed. If this is allowed, there are cases in which an equilibrium will never be computed.

Figure 9 on page 31 depicts an example of this problem. In this figure, there are three links e_1, e_2 and e_3. Each link has $S_i = 1$ and initial price $p_i = \varepsilon$, and we assume $\varepsilon < 0.5$. There is a single VC agent that wants to route traffic from node 1 to node 3. This agent has throughput goal $\gamma = \infty$ and wealth $W = \$1$. The agent can choose either path $\{e_1, e_2\}$ or simply $\{e_3\}$. To maximize throughput,

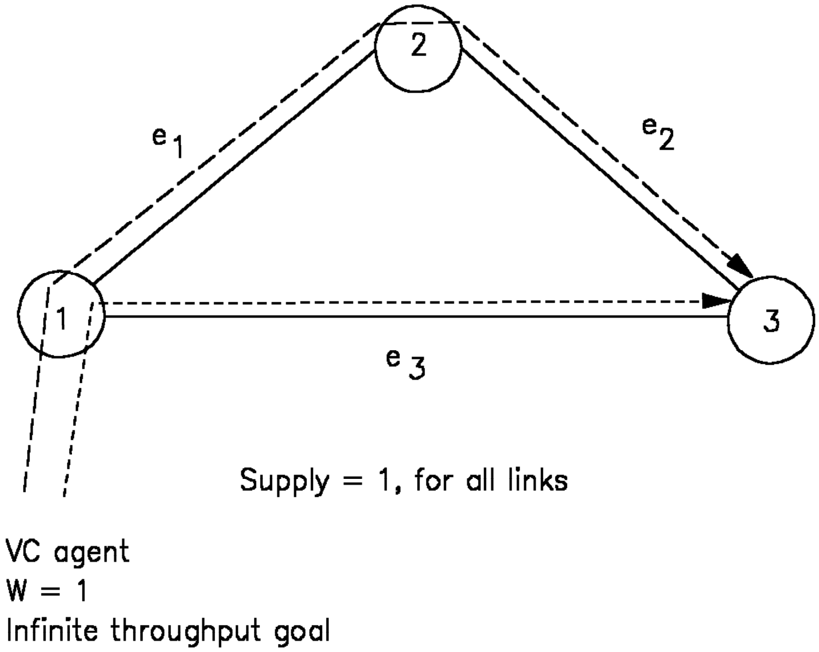

Figure 9. Routing Pathology: If the VC agent can choose a new path after each price update, the economy does not converge.

the agent chooses the cheaper path $\{e_3\}$ and demands capacity $\frac{1}{\varepsilon}$ which is greater than S_3. So, p_3 increases. As soon as $p_3 > 2 \bullet \varepsilon$, the agent switches to the cheaper path $\{e_1, e_2\}$ and demands capacity $\frac{1}{2 \bullet \varepsilon}$ of each link. This causes the price of p_3 to fall because demand is now 0, and causes and the prices of p_1 and p_2 to rise. As soon as $p_1 + p_2 > p_3$, the VC switches back to path $\{e_3\}$. So, the VC oscillates between paths $\{e_3\}$ and $\{e_1, e_2\}$. The economy never reaches an equilibrium.

6.0 Comparison with Related Work

The flow control problem for virtual circuit networks has been extensively studied. Comprehensive surveys of this work can be found in [3, 8, 28]. In this section, we compare previous work to the flow control economy. Flow control has two goals. The first is obtaining an optimal trade-off between throughput and delay [3]. The second is fairly allocating the communication resources to the set of VCs. The economy is compared to previous work with respect to meeting these goals.

Window flow control is the most common flow control algorithm used in practice [3, 28]. Each VC has a source A that sends data units and a destination B that accepts the data units and sends acknowledgements back to A. Flow control is implemented by assigning a window size W to A. Source A is allowed to have at most W unacknowledged data units sent to B. When the window is full, A must wait until receiving an acknowledgement before sending any additional data.

The idea behind window flow control is that when the network is becoming congested, the delays of routing a data unit from A to B and routing the acknowledgement back increase. This decreases the rate at which A can transmit. This occurs over all VCs, and the system wide average delay is improved by decreasing the throughputs of the senders.

The main advantage of window flow control relative to the flow control economy is that window flow control is dynamic. At most W data units can be transmitted before throughput is decreased to compensate for a transient period of congestion [3]. The flow control economy computes static transmission rates and does not react to transient conditions. A second advantage is that the window sizes can be chosen to guarantee that buffer overflows do not occur at any routing nodes.

One problem with window flow control is choosing the window sizes. The throughput of a VC can partially be determined by the choice of its window size, e.g. - large windows increase throughput. However, the throughput and delay of the VC are also determined by the window sizes of other VCs. This makes the choice of window sizes complicated. The flow control economy has a similar problem with the initial endowment of wealth to VCs. Exploring how the choice of the W_a's effects the relative throughputs and delays of the VCs is left for future work.

Another problem with window flow control is that it implements a system wide throughput-delay trade-off. For example, the set of VCs experiencing delays due to congestion may be throughput oriented, and should not have their window sizes decreased. The flow control economy does not have this problem.

Window flow control can be unfair to short VCs [3]. The window size of a VC must be an increasing function of the number of links in its path to meet its throughput goals during periods with light contention. During periods of heavy congestion, this makes it likely that a long VC gets more capacity of a heavily loaded link than a shorter VC sharing the same link [3]. The fairness of the flow control economy with respect to long VCs versus short VCs was described in section 3.

The flow control problem can be combined with circuit routing to form a single optimization problem [3]. The problem is to choose a path P_a and a transmission rate (throughput) r_a for each VC a that minimizes:

$$\sum_a D_a + \sum_a e_a(r_a).$$

D_a is VC a's average delay given the routing and transmission rates. $e_a(r_a)$ is a *penalty function* for the VC. As r_a decreases, this function increases and defines the penalty for flow controlling this circuit.

The main disadvantage to this approach is the it requires computationally intensive, centralized computing to determine the routing and transmission rates. The flow control economy also performs a minimization problem when the VCs try to minimize delay. We have an efficient decentralized algorithm that is specific to our optimization problem. It is not clear how the computational overheads of the two approaches compare.

A second disadvantage of the combined routing-flow control problem is that the optimization must be recomputed each time a new VC is activated or an existing VC terminates. This incurs overhead. Furthermore, the new paths may be different from the existing paths. So, additional overhead is incurred terminating the existing paths and starting the new ones.

The definition of the penalty functions adds complexity to the flow control problem. e_a must reflect VC a's need for throughput. The rate actually assigned to VC a also depends on the penalty functions of other VCs. So, the interactions between the penalty functions increase problem complexity. The assignment of wealth to a VC in the flow control economy also should consider the values assigned to other VCs. This is straight forward, however, and does not require comparisons of throughput-delay goals.

Finally, combining routing and flow control in a single optimization problem may assign 0 throughput to some VCs. So, it is not fair be condition F1.

The main advantage of the combined flow control-routing approach is that it can improve on the allocations given to the VCs compared to the flow control economy. We have proven that the economy computes pareto-optimal resource allocations given a fixed set of VC paths. However, choosing an alternate set of paths may result in an allocation the improves the lot of all VCs. We demonstrated that the economy cannot implement dynamic routing in addition to flow control. This is a major shortcoming of the economy. Quantifying the performance degradation due to not dynamically assigning routes is an area for future work.

A third class of flow control algorithms are *Bottleneck* or *max-min* algorithms [3]. These algorithms are based input rate adjustment. The algorithms′ primary goals is to compute allocations which are fair by condition F2. An efficient decentralized algorithm for computing rate assignments that are fair by condition F2 has been proposed by Jaffe [12]. Other algorithms are presented in [3].

The main disadvantage of bottleneck flow control is the assumption that all VCs are throughput oriented. This approach is less general than the flow control economy. We compared the fairness properties of the economy with bottleneck flow control in section 1.

A major benefit of the flow control economy is that it supports decentralized choice of individual throughput-delay trade-offs. Some previous work on the flow control problem has allowed a VC to define its throughput-delay goals independently of other VCs. Sanders [25, 26] presented an algorithm for computing optimal transmission rate allocations under a model in which each VC has an individual *utility function* that defines its throughput-delay trade-off. Possible utility functions were not defined. Sanders algorithm is decentralized and asynchronous, and maximizes the sum of the individual utilities. The algorithm can assign transmission rate 0 to a circuit and is unfair by condition F1. The algorithm also requires some coordination between choice of utility functions.

Kadaba and Jaffe [4] proposed a generalized form of power for for representing throughput-delay trade-offs. This work focused on maximizing a global objective function based on the power obtained by individual VCs, e.g. - the sum of the individual powers. The individual VCs did not choose their own definition of power and were assumed to be homogeneous, however. In some examples, their algorithm computes rate allocations that are a Nash Equilibrium. It was shown in section 2 that a Nash Equilibrium is not Pareto-optimal. In their work, Kadaba and Jaffe observed that their algorithm computed an allocation that could be improved upon by all VCs.

Kadaba and Jaffe use power to model throughput-delay trade offs. Let γ_k be VC k′s throughput, and D_k is its delay. VC k chooses a constant β_k and attempts to maximize its power

$$\frac{(\gamma_k)^{\beta_k}}{D_k}.$$

Power is an awkward representation for a individual VC throughput-delay goals. It would be very difficult for the user of VC k to choose the constant β_k that achieves its throughput-delay goals. The model used in the flow control economy makes specifying throughput-delay goals simpler.

Some previous work has applied game theoretic models to the flow control problem. Hsiao and Lazar [10, 11] were the first to observe the similarities between the flow control problem and the theory of non-cooperative games [30]. These similarities are decentralized decision making and conflicting individual goals. Hsiao and Lazar proposed Nash Equilibrium as an optimality criteria for the flow control problem and derived the set of flow control problems in Markovian networks [14] for which a Nash Equilibrium exists. Bovopoulos and Lazar [5] continued this work by presenting algorithms for computing a Nash Equilibrium in restricted environments (2 players; singer server queue). Douglieris and Mazumdar [23, 24] have studied Nash Equilibria and Pareto-optimality in single server queues with power as the user preference relation. Recently, Shenker has come up with the most generalized version of the game theoretic approach [29].

The main difference between this paper and the game theoretic work is that our algorithms compute Pareto-optimal allocations, which is a stronger definition of optimality than Nash Equilibrium. Our network model is also more general than most previous work. Finally, there is no use of resource prices or agent wealth in the previous work. The main disadvantage of this paper is that we have not proved convergence of our algorithm, while the game theoretic work has shown convergence in restricted domains.

A growing body of research has studied applying economic algorithms and concepts to problems in computer science. This research has been motivated by the similarities between human economies and large, distributed computer systems. The hope of this research is that structures that human societies have evolved over thousands of years to allocate resources will also prove effective in other types of large systems.

The problem of CPU load balancing has been studied by Ferguson, Nikolaou and Yemini [7], by Malone et al. in the *Enterprise System* [21, 22] and in the *Spawn* prototype (Waldspurger et al. [31]). In addition to load sharing, other resource allocation problems have been addressed using economic algorithms. Kurose and Simha [18-20] have studied the File Allocation Problem. Finally, Yemini and Kleinrock [33], Yemini [32] and Kurose, Schwartz and Yemini [16, 17] have examined economies that allocates transmission probabilities in broadcast networks.

The main contribution of this paper relative to previous work is demonstrating the power of mathematical economics to provide insight into computer systems structured as economies. We have shown the benefits of Pareto-optimality as a definition of optimality in heterogeneous environments and have been able to prove optimality and existence of an equilibrium operating point. This paper is also the first to derive implementable algorithms from the theory.

7.0 Conclusions

In this paper, we have applied concepts from mathematical economics to develop an effective flow control algorithm for virtual circuit (VC) networks. The immediate advantages of our algorithm compared to previous flow control mechanisms, are that: 1) it captures in a precise manner both diversity of priorities among virtual circuits in a network as well as diversity of their performance objectives (throughput vs. delay). 2) it accomplishes resource allocations that are optimally fair (in a formal sense). We have compared the fairness and optimality properties of our algorithms with previous work.

Our algorithms are decentralized and are simple to implement. Additionally, simulation studies demonstrated that the economy efficiently computes optimal resource allocations for realistically large networks. We showed that the economy rapidly computes the new optimal allocations when a new virtual circuit is activated or an existing circuit terminates. The prices of the communication links set by our algorithms can provide important information for new VCs entering the system. First, the VC′s user can determine if its performance objectives are affordable. If they are not, the user can modify its goals or defer entry and *admission control* is implemented. Second, the prices can be used to make routing decisions for the new VC. We demonstrated that there are limitations with the admission control and dynamic routing capabilities of our economic model, however. Overcoming these disadvantages is an important are for future research.

In this paper, Pareto-optimality was shown to provide a powerful new definition for optimal, fair resource allocations in heterogeneous network environments. Economic techniques were used to prove optimality of the resource allocations computed by the economy, and that an optimal allocation does exist for all networks under our model. This paper illustrated that two main challenges must be overcome when applying economic theory to resource management in distributed computer systems. First, the underlying economic models must be modified to accurately reflect the problem being studied. Secondly, and most importantly, it must proven that the system possesses fundamental properties assumed by economic theory, and it cannot be assumed that the economy possesses these properties. For example, continuity of demand with respect to prices is assumed in economic theory. In this paper we had to prove that the agents′ diverse demand functions were continuous.

Several areas for future work remain. First, two theoretical questions were raised. One is uniqueness of the resource allocations our algorithms compute. The second is a proof of convergence or a counter example. Another area is expanding the economy to incorporate other models of VC goals.

This would increase the generality of the results. Finally, the use of link prices to determine long term routing decisions, to guide capacity planning and to set demand based rate setting in public access networks are potentially fruitful areas for further work.

Acknowledgements : The authors would like to thank Mon-Song Chen and Bharath Kadaba for their initial comments on this work, and Leo Georgiadis for verifying the mathematical results.

Appendix A. Proof of Lemmas

This section proves lemmas on which results in this paper depend. Let $G = \langle V, E \rangle$ be the graph of the underlying network, and let the current price vector be $\dot{p} = \langle p_1, p_2, \ldots, p_m \rangle$ with $0 < \varepsilon \le p_i < \infty$ for all i. A VC agent a in this economy has a non-empty path $P_a = \{l_{a,1}, l_{a,2}, \ldots, l_{a,m(a)}\}$, wealth $W_a > 0$ with which it purchases allocations and a throughput goal γ_a. To make the notation simpler, the $\dot{p}$ will be dropped from the allocation vectors when the context is clear, i.e. - $\dot{\phi}(\dot{p})$ is written simply $\dot{\phi}$.

For the proof of lemma 2.1, define a special allocation $\dot{\beta}^a$ for VC a with:

1. $\beta_i^a = 0$, if $i \notin P_a$.
2. $\beta_i^a = \dfrac{W_a}{Q_a}$, if $i \in P_a$.

where $Q_a = \sum\limits_{i \in P_a} p_i$.

Lemma 2.1 : If $W_a \le Q_a \bullet \gamma_a$ at prices $\dot{p}$, then:

1. $\Phi_a(\dot{p})$ is non-empty.
2. $\Phi_a(\dot{p})$ has a single element $\dot{\phi}$.
3. $\phi_i = \dfrac{W_a}{Q_a}$, if $i \in P_a$.
4. $\phi_i = 0$, if $i \notin P_a$.

Proof:

All four propositions will be proven if we can show $\dot{\beta}^a$ is VC a's unique optimal choice at prices $\dot{p}$.

First,

$$\dot{\beta}^a = \sum_{i \in P_a} p_i \bullet \frac{W_a}{Q_a} = \frac{W_a}{Q_a} \bullet Q_a = W_a.$$

So $\dot{\beta}^a$ is affordable and $B_a(\dot{p})$ is not empty. The preference relation $\bullet_a$ is a partial order, and has a maximal element. This means that $\Phi_a(\dot{p})$ is not empty.

Suppose there exists another allocation $\dot{\alpha}^a \bullet \dot{\beta}^a$. Then we have the following:

1. $\alpha_i^a \geq 0$ implies that $\alpha_i^a \geq \beta_i^a$ for $i \notin P_a$.

2. $T(\dot{\alpha}^a) \geq T(\dot{\beta}^a)$, which in turn implies $\alpha_i^a \geq \frac{W_a}{Q_a}$, $i \in P_a$.

If $\dot{\alpha}^a \sim_a \dot{\beta}^a$, then $T(\dot{\alpha}^a) = T(\dot{\alpha}^a) = \frac{W_a}{Q_a}$ and $\dot{\alpha}^a = \dot{\beta}^a$. If $\dot{\alpha}^a \bullet_a \dot{\beta}^a$, then $T(\dot{\alpha}^a) > T(\dot{\alpha}^a) = \frac{W_a}{Q_a}$. This means $\alpha_i^a > \frac{W_a}{Q_a}$, and $\dot{\alpha}^a \bullet \dot{p} > W_a$. So, $\dot{\alpha}^a$ is not affordable.

Lemma 3.1 : Let (1) $\dot{p}$ be a price system. (2) $\dot{\phi}^a(\dot{p})$ be agent a's demand at $\dot{p}$. (3) $\dot{x} \neq \dot{\phi}^a(\dot{p})$ be any allocation. Then,

1. $\dot{\phi}^a(\dot{p}) \bullet \dot{p} = W_a$.

2. $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$ implies that $\dot{x} \bullet \dot{p} \geq \dot{\phi}^a(\dot{p}) \bullet \dot{p}$.

3. $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$ implies $\dot{x} \bullet \dot{p} > \dot{\phi}^a(\dot{p}) \bullet \dot{p}$.

Proof:

If $T(\dot{\phi}^a(\dot{p})) \leq \gamma_a$ the results are obvious. So, assume that $T(\dot{\phi}^a(\dot{p})) \geq \gamma_a$.

(1) If $\dot{\phi}^a(\dot{p}) \bullet \dot{p} < W_a$, then VC a has a surplus of $\delta = W_a - \dot{\phi}^a(\dot{p}) \bullet \dot{p}$, with $\delta > 0$. Buying $\frac{\delta}{p_i}$ of any link $i \in P_a$ improves the delay because delay decreases monotonically as a function of x_i when $x_i \geq \gamma_a$. So, $\dot{\phi}^a(\dot{p})$ was not optimal, and this is a contradiction.

(2,3) Assume the opposite: $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$, and $\dot{x} \bullet \dot{p} < \dot{\phi}^a(\dot{p}) \bullet \dot{p}$. Perform the following transformations

1. $u_i = \phi_i^a(\dot{p}) - \gamma_a$, for $i \in P_a$.

2. $w_i = x_i - \gamma_a$, for $i \in P_a$.

3. $u_i = \phi_i^a(\dot{p})$, $i \notin P_a$.

4. $w_i = x_i$, $i \notin P_a$.

This means that $\dot{p} \bullet \dot{w} \leq W_a - Q_a \times \gamma_a$, and $\dot{w}$ is feasible (affordable). $\dot{x} \sim_a \dot{\phi}^a(\dot{p})$ means that in the minimization problem of section 2, $F(\dot{u}) = F(\dot{w})$. This contradicts the strong convexity of F. If $\dot{x} \bullet_a \dot{\phi}^a(\dot{p})$, then $F(\dot{w}) < F(\dot{u})$. This contradicts the optimality of $\dot{u}$ and the definition of $\dot{\phi}^a(\dot{p})$.

lemma 4.1 : $F(\dot{u})$ is continuous and *strongly convex*.

Proof: :

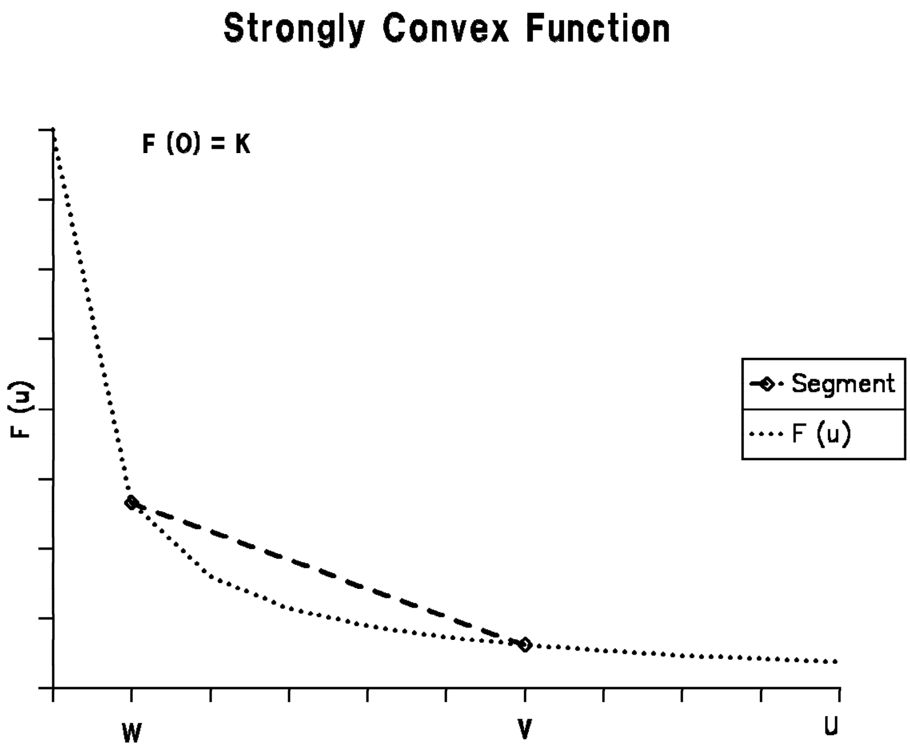

Figure 10. Strong convexity of F_i

F is the sum of m functions

$$F_i(\dot{u}) = \frac{1}{K_i + u_i}$$

each of which is strongly convex and continuous ($K_i > 0, u_i \geq 0$). Figure 10 shows the form of F_i. It is clear that the segment connecting $F_i(w)$ and $F_i(v)$ always lies above the curve.

Strong convexity of the F_i's means that for all $u, w \in U$,

$$\alpha \bullet F_i(\dot{u}) + (1 - \alpha) \bullet F_i(\dot{w}) > F(\alpha \bullet \dot{u} + (1 - \alpha) \bullet \dot{w}).$$

This in turn implies,

$$\sum_{i \in P_a} \alpha \bullet F_i(\dot{u}) + (1-\alpha) \bullet F_i(\dot{w}) > \sum_{i \in P_a} F_i(\alpha \bullet \dot{u} + (1-\alpha) \bullet \dot{w}).$$

This means $\alpha \bullet F(\dot{u}) + (1-\alpha)F(\dot{w}) > F(\alpha \bullet \dot{u} + (1-\alpha) \bullet \dot{w})$ and F is strongly convex.

Lemma 4.2 : U is convex and compact.

Proof :

We re-write $W_a - C_a \bullet \gamma_a$ as K_a to simplify the notations.

(Compact) : U is compact if it is bounded and closed. By definition, U is bounded below by $\dot{0} = <0, 0, \ldots, \ldots, 0>$. We know that $\dot{p} \bullet \dot{u} \leq K_a$. This requires that $p_i \bullet u_i \leq K_a$ which in turn means that $u_i \leq \frac{K_a}{p_i}$. We know that $p_i > 0$ which means that $\frac{K_a}{p_i} < \infty$. So, U is bounded from above by the vector

$$< \frac{K_a}{p_1}, \frac{K_a}{p_2}, \ldots, \frac{K_a}{p_M} > .$$

For U to be closed, any sequence of vectors $\dot{u}^n \in U$ that converges must converge to an element $\dot{u}^*$ of U. Assume the sequence $\dot{u}^n$, $n = 1, 2, \ldots$ converges to $\dot{u}^*$. This means that for all $i = 1, 2, \ldots, M$, u_i^n converges to u_i^*. Assume that $u_i^* < 0$. This means that $u_i^* = 0 - \delta, \delta > 0$. By definition, $u_i^n \geq 0$, for all n. This means that $\lim |u_i^n - u_i^*| \geq \delta$. This contradicts the assumption of convergence, and $u_i^* \geq 0, i = 1, 2, \ldots, M$.

$|\dot{p} \bullet \dot{u}^n - \dot{p} \bullet \dot{u}^*| =$

$$\sum_{i=1}^{M} |u_i^n - u_i^*| \bullet p_i.$$

We know that as $n \to \infty$, $|u_i^n - u_i^*| \to 0$ that in turn implies $|\dot{p} \bullet \dot{u}^n - \dot{p} \bullet \dot{u}^*| \to 0$. We know that $\dot{p} \bullet \dot{u}^n \leq K_a$, so $\dot{p} \bullet \dot{u}^* \leq K_a$.

We have shown that $u_i^* \geq 0$ and $p \bullet \dot{u}^* \leq K_a$. So, $\dot{u}^* \in U$ and U is closed. U is closed and bounded which means that it is compact.

(Convex) : For U to be convex, for any two vectors $u, v \in U$ and any α, $0 \leq \alpha \leq 1$, the vector

$$\dot{w} = \alpha \bullet \dot{u} + (1 - \alpha)\dot{v}$$

must be in U. By definition, $u_i \geq 0$, $v_i \geq 0$, $\alpha \geq {}_0$ and $1 - \alpha \geq 0$, so $w_i = \alpha \bullet u_i + (1 - \alpha) \bullet w_i \geq 0$.

$\dot{p} \bullet \dot{w} = \dot{p} \bullet (\alpha \bullet \dot{u}) + \dot{p} \bullet ((1 - \alpha) \bullet \dot{v})$. This in turn means, $\dot{p} \bullet \dot{w} = \alpha \bullet (\dot{p} \bullet \dot{u}) + (1 - \alpha) \bullet (\dot{p} \bullet \dot{v})$. By definition, $\dot{p} \bullet \dot{u} \leq K_a$ and $\dot{p} \bullet \dot{v} \leq K_a$. So, $\dot{p} \bullet \dot{w} \leq \alpha \bullet K_a + (1 - \alpha) \bullet K_a = K_a$.

Lemma 4.3 : : A sequence of points $\{\dot{x}^m\}$ in $\bullet^N$ converges to $\dot{x}^0$ if, and only if, every sub sequence $\{\dot{x}^{m_k}\}$ contains a subsequence $\{\dot{x}^{m'_k}\}$ that converges to $\dot{x}^0$.

Proof :

The if part follows directly from the definition of convergence

For the only if part, assume that $\dot{x}^m \not\to \dot{x}^0$. Also assume that every subsequence contains a subsequence converging to $\dot{x}^0$.

$\dot{x} \not\to \dot{x}^0$ implies that there exists a $\varepsilon > 0$ such that for any $k \geq 1$, there is an m_k such that $||\dot{x}^{m_k} - \dot{x}^0|| > \varepsilon$. For $k = 1, 2, \ldots$ take the sequence of points $\dot{x}^{m_k}$. By definition, this sequence cannot contain a subsequence converging to $\dot{x}^0$ because for all k, $||\dot{x}^{m_k} - \dot{x}^0|| > \varepsilon$.

This contradicts the assumptions, and the lemma is shown.

We now prove one of the assertions made in the proof of continuity of a VC′s demand function with respect to prices. This was theorem 4.2. Let $\dot{x}^0, \dot{p}^0, \dot{p}^m$ be defined as for theorem 4.2. For the sequence of points $\dot{p}^m \to \dot{p}^0$, define a sequence of points $\dot{w}^m \in U(\dot{p}^m)$ where

$$||\dot{x}^0 - \dot{w}^m|| \leq ||\dot{x}^0 - \dot{z}^m||$$

for all $\dot{z} \in U(\dot{p}^m)$. We prove the following lemma:

Lemma 4.4 : As $\dot{p}^m \to \dot{p}^0$, $||\dot{x}^0 - \dot{w}^m|| \to 0$.

Proof :

We construct a sequence $\dot{y}^m$ that converges to $\dot{x}^0$. This will prove convergence of $\dot{w}^m$.

If $\dot{x}^0 = \dot{0}$ then $\dot{y}^m = \dot{0}$.

Assume that $\dot{x}^0$ is a vector in $U(\dot{p}^0)$. Let $S \subseteq \{1, 2, \ldots, M\}$ be the links for which $\dot{x}_i^0 > 0$. Define e_m as

$$e_m = \min \{(p_s^0 - p_s^m) \mid s \in S\}.$$

We have $e_m \to 0$.

We know that

$$p_s^m + e_m \leq p_s^0. \tag{1}$$

The budget constraint tells us that

$$\sum_{i=1}^{M} p_i^0 x_i^0 \leq W - \gamma \sum_{i=1}^{M} p_i.$$

Plugging the inequality (1) into the budget constraint gives

$$\sum_{s \in S} (p_s^m + e_m) x_s^0 \leq W - \gamma(\sum_{i=1}^{N} p_i^m) - \gamma e_m M.$$

Rearranging terms gives

$$\sum_{s \in S} p_s^m x_s^0 + e_m \sum_{s \in S} x_s^0 + \gamma e_m M \leq W - \gamma \sum_{i=1}^{M} p_i^m.$$

We multiply and divide the second term in this inequality by the cardinality of S, denoted $|S|$, and get

$$\sum_{s \in S} p_s^m x_s^0 + |S| e_m \left(\frac{\left(\sum_{s \in S} x_s^0 \right) + \gamma M}{|S|} \right) \leq W - \gamma \sum_{i=1}^{M} p_i^m. \tag{2}$$

For simplicity of notation, let

$$A = \frac{\left(\sum_{s \in S} x_s^0\right) + \gamma M}{|S|}.$$

By definition of the cardinality of a set we have

$$|S| e_M A = \sum_{s \in S} e_M A,$$

and multiplying and dividing each i-th term p_i^m gives

$$|S| e_M A = \sum_{s \in S} p_s^m \left(\frac{e_M A}{p_s^m} \right)$$

This means we can rewrite inequality (2) as

$$\sum_{s \in S} p_s^m \left(x_s^0 + \frac{e_m A}{p_s^m} \right) \leq W - \gamma \sum_{i=1}^{M} p_i^m. \tag{3}$$

We know that $e_m \rightarrow 0$, and $p_s^m \rightarrow p_s^0$. By definition, $p_s^0 > 0$, and by the definition of the set S, we know $x_s^0 > 0$. This implies that we can find $K \in \bullet$ such that for all $s \in S$, if $m > K$,

$$x_s^0 \geq -\frac{e_m}{p_s^m} \bullet \frac{A}{|S|}. \tag{4}$$

The sequence $\{\dot{y}^m\}$ is defined by components as follows:

1. $y_i^m = 0$, if $m \leq K$

2. $$y_i^m = x_i^0 + \frac{e_m}{p_s^m} \bullet \frac{A}{|S|}, \text{ if } m > K \text{ and } i \in S$$

3. $y_i^m = 0$, if $m > K$ and $i \notin S$

From inequality (4), we know that $y_i^m \geq 0$, and inequality (3) shows that $\dot{y}^m$ is affordable. So, $\dot{y}^m \in U(\dot{p}^m)$.

We know that:

1. $e_m \to 0$.
2. $p_i^m \to p_i^0 > 0$.
3. $|S| > 0$.

This means that $y_i^m \to x_i^m$. This proves the lemma

Bibliography

[1] Kenneth J. Arrow and Michael D. Intrilligator, editor. *Handbook of Mathematical Economics*.
North-Holland Publishing Co., Amsterdam, 1981.

[2] K. Arrow and F. Hahn.
General Competitive Analysis.
Holden Day Publisher, San Francisco, 1971.

[3] Dimitri Bertsekas and Robert Gallager.
Data Networks.
Prentice-Hall, Englewood Cliffs, NJ, 1987.

[4] Kadaba Bharath-Kumar and Jeffrey Jaffe.
A New Approach to Performance Oriented Flow Control.
IEEE Transactions on Communication, COM-29(4), April 1981.

[5] Andreas Bovopoulos and Aurel Lazar.
Decentralized Algorithms for Optimal Flow Control.
Proceedings of the 25th Annual Allerton Conference on Communication, Control and Computing, September 1987.

[6] Donald Ferguson.
The Application of Microeconomics to the Design of Resource Allocation and Control Algorithms.
PhD thesis, Computer Science Department, Columbia University, New York, NY. 1989.

[7] Donald Ferguson, Yechiam Yemini and Christos Nikolaou.
Microeconomic Algorithms for Load Balancing in Distributed Computer Systems.
Proceedings of the 8th International Conference on Distributed Computing Systems, June 1988.

[8] Mario Gerla and Leonard Kleinrock.
Flow Control: A Comparative Survey.
IEEE Transactions on Communication, COM-28(4), April 1980.

[9] W. Hildenbrand and A.P. Kirman.
Introduction to Equilibrium Analysis.
North-Holland Publishing Co., Amsterdam, 1976.

[10] Man-Tung Tony Hsiao.
Optimal Decentralized Flow Control in Computer Communication Networks.
PhD thesis, Center for Telecommunications Research, Columbia University, New York, NY. 86.

[11] Man-Tung Hsiao and Aurel Lazar.
A Game Theoretic Approach to Optimal Decentralized Flow Control of Markovian Queueing Networks with Multiple Controllers.
*Proceedings of Performance '*87, Brussels, Belgium, December 1987.

[12] Jeffrey Jaffe.
Bottleneck Flow Control.
IEEE Transactions on Communication, COM-29(7), July 1981.

[13] Jeffrey Jaffe.
Flow Control Power is Nondecentralizable.
IEEE Transactions on Communication, COM-29(9), September. 1981.

[14] Leonard Kleinrock.
Queueing Systems: Volume I: Theory.
Wiley Interscience, New York, 1975.

[15] Leonard Kleinrock.
Queueing Systems: Volume II: Computer Applications.
Wiley Interscience, New York, 1975.

[16] James F. Kurose.
Time-Constrained Communication in Multiple Access Networks.
PhD thesis, Columbia University, 1984.

[17] James F. Kurose, Mischa Schwartz and Yechiam Yemini.
A Microeconomic Approach to Optimization of Channel Access Policies in Multiaccess Networks.
Proceedings of the 5th IEEE Distributed Computer Systems Conference, pages 70-80, May 1985.

[18] James F. Kurose and Rahul Simha.
A Microeconomic Approach to Optimal File Allocation.
Proceedings of the 6th IEEE Distributed Computer Systems Conference, 1986.

[19] James F. Kurose and Rahul Simha.
Second Derivative Algorithms for Optimal Resource Allocation in Distributed Computer Systems.
Proceedings of the 7th International Conf. on Distributed Computing Systems, 1987.

[20] James F. Kurose and Rahul Simha.
A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems.
IEEE Transactions on Computers, 38(5), May 1989.

[21] Thomas W. Malone, Richard E. Fikes, Kenneth R. Grant and Michael T. Howard.
Market Like Load Sharing in Distributed Computing Environments.
MIT, 1987.

[22] Thomas W. Malone, Richard E. Fikes, Kenneth R. Grant and Michael T. Howard.
Enterprise: A Market-Like Task Scheduler for Distributed Computing Environments.
in B. A. Huberman, editor, *The Ecology of Computation*, North-Holland, Amsterdam, 1988.

[23] Christos Douglieris and Ravi Mazumdar.
On Pareto Optimal Flow Control in a Multiclass Environment.
Center for Telecommunications Research, Columbia University, 1987.

[24] Christos Douglieris and Ravi Mazumdar.
User Optimal Flow Control in an Integrated Environment.
CTR Technical Report, Columbia University, 1987.

[25] Beverly Sanders.
An Incentive Compatible Flow Control Algorithm for Fair Rate Allocation in Computer/Communication Networks.
Proceedings of the 6th IEEE Distributed Computer Systems Conference, pages 314-320, May 1986.

[26] Beverly Sanders.
An Asynchronous, Distributed Flow Control Algorithm for Rate Allocation in Computer Networks.
IEEE Transactions on Computers, C-37(7), July 1988.

[27] Herbert Scarf.
The Computation of Economic Equilibria.
Yale University Press, New Haven, CN, 1973.

[28] Mischa Schwartz.
Telecommunication Networks: Protocols, Modeling and Analysis.
Addison-Wesley Publishing Co., Reading, Mass., 1987.

[29] Scott Shenker.
Making Greed Work in Networks: A Game-Theoretic Analysis of Gateway Service Disciplines.
XEROX Palo Alto Research Center, 1989.

[30] Martin Shubik.
Game Theory in the Social Sciences.
The MIT Press, Cambridge, Mass., 1983.

[31] Carl A. Waldspurger, Tad Hogg, Bernardo Huberman, Jeffrey O. Kephart and Scott Stornetta.
Spawn: A Distributed Computational Ecology.
Xerox Palo Alto Research Center, 1989.
Submitted to CACM.

[32] Yechiam Yemini.
Selfish Optimization in Computer Networks.
Proceedings of the 20th IEEE Conference on Decision and Control, pages 281-285, IEEE, San Diego, December 1981.

[33] Yechiam Yemini and Leonard Kleinrock.
On a General Rules for Access Control or Silence is Golden....
Proceedings of the International Conference on Flow Control in Computer Networks, pages 335-347, North Holland Press Amsterdam, Versailles, February 1979.