Making Greed Work in Networks:
A Game-Theoretic Analysis of Switch Service Disciplines

Scott Shenker
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304-1314
415-812-4840
shenker@parc.xerox.com

Abstract

This paper discusses congestion control from a game-theoretic perspective. There are two basic premises: (1) users are assumed to be independent and selfish, and (2) central administrative control is exercised only at the network switches. The operating points resulting from selfish user behavior depend crucially on the service disciplines implemented in network switches. This effect is investigated in a simple model consisting of a single exponential server shared by many Poisson sources. We discuss the extent to which one can guarantee, through the choice of switch service disciplines, that these selfish operating points will be efficient and fair. We also discuss to what extent the choice of switch service disciplines can ensure that these selfish operating points are unique and are easily and rapidly accessible by simple self-optimization techniques. We show that no service discipline can guarantee optimal efficiency. As for the other properties, we show that the traditional FIFO service discipline guarantees none of these properties, but that a service discipline called *Fair Share* guarantees all of them. While the treatment utilizes game-theoretic concepts, no previous knowledge of game theory is assumed.

1 Introduction

Congestion has long been a problem in computer networks. During the past decade, much effort has been devoted to understanding the nature of congestion and developing techniques for its control. Many different congestion control mechanisms have been proposed, and numerous studies have been published evaluating their relative performance. However, buried beneath these detailed mechanistic proposals is a fundamental disagreement about network design philosophy. Most of the earlier proposed congestion control schemes assume the cooperation of network users[1], requiring them to implement a particular flow control algorithm at the end hosts [11, 13, 27]. In this approach, users adopt a centrally mandated algorithm, and the role of the designer is to make the resulting system-wide behavior achieve certain systemic goals such as high utilization or low delay. This is the usual paradigm in the study of distributed systems; while evaluating the relative merits of the various proposals is often technically difficult, it poses no particular paradigmatic challenge.

Some of the more recent work takes a rather different approach. This approach not only concedes that it is impossible to centrally mandate the behavior of end users, but actively contends that such centralized administrative control is not advisable. Rather than following some mandated algorithm, in this approach users are assumed to act "selfishly" to further their own individual interests. The role of the designer here is confined to mandating the behavior of network switches (which are assumed to be under centralized administrative control). The goal is to design these switch service disciplines so that the network will deliver good service in spite of selfish user behavior. Reference [3] is an example of this approach; it focuses on designing effective service disciplines in network switches and then letting end hosts do whatever is in their own best interest.

This approach, with its emphasis on individual incentives, does not fit the typical paradigm used in the study of distributed systems. If users are not following some centrally mandated algorithm, how can one model user behavior? How can one describe the eventual network operating point in such noncooperative systems? If users are acting to further their own interests, rather than that of the system, by what criteria does one evaluate the resulting system-wide behavior? These questions, which involve individual incentives in a fundamental way, are largely foreign to computer science; however, they are the very core of game theory. The purpose of this paper is to illustrate how game theory can be used to formulate and answer, at least on a theoretical level, these incentive questions. Other than reviewing some of the arguments briefly in Section 2.2, we are not revisiting the basic debate between the two approaches to congestion control. Rather, we are merely exhibiting how one can formalize and analyze this second approach using a game-theoretic perspective.

We will apply this game-theoretic perspective to a very simple system: a single switch shared by N users. Each user sends a Poisson stream of packets through the switch. The rate of the i'th user's Poisson stream is denoted by r_i; the algorithm by which the user controls this rate is called the flow control algorithm. The switch is serviced by an exponential server with preemption. One measure of the congestion experienced by a user is the average number of that user's packets that

[1] The term user here is purposely ambiguous. While the human user ultimately controls the implementations used on the network device, it is the device which typically controls the network behavior in the short term. To avoid continually tripping over this distinction, we will use the term user to refer to whatever entity is controlling the behavior.

are waiting in the server's queue[2]. This congestion, or average queue length, will be denoted by c_i and depends on the set of rates r_j and on the switch service discipline. In this paper we explore the possibilities and limitations of designing service disciplines that produce desirable results in the presence of selfish user behavior. The game-theoretic implications of switch service disciplines, and their relevance to network congestion control, was first articulated by Nagle in [26].

In the next section we present the basic premises of the game-theoretic approach, and justify its relevance. Section 3 contains the mathematical details of the simple single-switch model. In Section 4 we define the properties one might desire, and then present technical results on the possibility of achieving these properties. Due to their length, the proofs of the technical results are presented in the Appendix. We conclude in Section 5 by discussing related work and generalizations of the model.

2 The Game-Theoretic Perspective

In this section we first discuss the basic premises of our game-theoretic approach. We then argue that this approach is indeed somewhat relevant to real design issues and is not merely a theoretical whimsy.

2.1 Basic Premises

Our analysis is based on four main principles:

1. **User satisfaction is a function of the amount and quality of service provided by the switch.**

   The amount of service is given by r_i. The quality of service is measured in terms of congestion, which here we are equating with the average queue length, c_i; note that *increasing* c_i reflects a *decreasing* quality of service. The content of the first principle is formalized via utility functions $U_i(r_i, c_i)$ which express the user's degree of satisfaction with a particular level of service. A user's preference for allocation[3] (r_i, c_i) over allocation $(\bar{r}_i, \bar{c}_i)$ is represented by having $U_i(r_i, c_i) > U_i(\bar{r}_i, \bar{c}_i)$. These utility functions allow us to express the fact that different users may have very different preferences; some users are very sensitive to congestion and others are more sensitive to throughput. Utility functions are private, known only by the individual user, and not by other users or the switch. Furthermore, users are aware only of their own quantities r_i and c_i and not those of others.

2. **Users are selfish.**

[2]We could equivalently, and perhaps more naturally, use the average delay of packets. However, this would require the convexity condition on U_i introduced in Section 3.2 to be significantly more complicated. Note that since $c_i = r_i d_i$, we have lost no generality by using the average queue size instead of the delay.

[3]In keeping with its economic usage, we are using the term allocation to describe the quantity and quality of service given to a user by the network. Contrary to standard networking usage, we are not using the term allocation to refer to reserved resources such as bandwidth or buffers.

Each user employs a flow control algorithm that maximizes their individual utility by varying r_i. When each user acts in this selfish way, the stable operating points of the system are Nash equilibria (to be defined later).

3. **The performance of the switch is evaluated solely in terms of the level of user satisfaction it provides.**

   Traditional discussions of network congestion focus on switch-centered quantities, such as *power* [13, 14, 27], line utilization [12], or total queueing delay. This neglects the fact that users could have very different preferences. In contrast, we use only the collection of utility functions U_i in assessing the performance of the switch.

4. **The switch algorithms are under central administrative control.**

   While users are independent entities, the switch is a shared resource. As such, it is under central administrative control. Our focus is on the switch service disciplines, which control the allocation of congestion.

Our challenge is to design the switch service disciplines so that, for every collection of utility functions, the system exhibits good performance in spite of the selfishness of individual users. This notion of "good" performance has three aspects. First, the Nash equilibria should be efficient and fair. Second, the Nash equilibria should be easily and rapidly accessible by simple self-optimizing techniques. Third, the system should provide some minimal performance guarantees even out of equilibrium.

In Section 4 we define these various properties more precisely and discuss to what extent one can achieve them. We show that there is no service discipline that always achieves optimal efficiency at the Nash equilibrium. However, it is possible to always achieve the other desirable properties. There is a service discipline, the *Fair Share* discipline, that guarantees all of these properties, and is the only monotonic service discipline that guarantees *any* of them (we will define monotonicity later). The Fair Share service discipline is based on the intuition that users should always receive their fair share of the service, in terms of amount and quality, regardless of what other users are doing. This insularity, or partial independence, enables the Fair Share service discipline to provide good service in the presence of selfish users. While the Fair Share service discipline is defined merely in terms of this simple M/M/1 system, it is similar in spirit to Fair Queueing [3] and other related packet scheduling disciplines [9, 15]. Our results provide further evidence for the superiority of Fair Queueing over the traditional FIFO service discipline, but this is not our focus in this paper. Instead, our intent here is to formulate the incentive issues in a more rigorous and systematic, albeit theoretical, manner.

2.2 In Defense of Selfishness

The central assumption in the game-theoretic approach described here is that users are noncooperative and selfish. Many dismiss this as an unacceptably pessimistic view of human nature, and contend that we can count on the cooperation of users. Since, as we shall show, the noncooperative algorithms presented here do not give optimally efficient (Pareto) Nash equilibria, and cooperative algorithms can achieve optimal efficiency, the question arises: if we can assume user cooperation, why not use cooperative algorithms? In cooperative solutions, users must not only be willing to cooperate, but there also must be universal agreement on what it means to cooperate. This requires a single, universally accepted flow control algorithm which, given a user's particular needs as expressed by her utility function, dictates a user's actions. There are three main problems with this cooperative approach.

First, cooperation requires that users know their utility function in the abstract; i.e., be able to make comparisons between allocations without having a chance to experience them. This is quite different than the knowledge required in a hill-climbing implementation of the selfish view[4], where users need only make local comparisons between allocations they actually receive (i.e., they only need to ask if they are happier now than they were a few minutes ago). For instance, when watching TV one cannot specify the desired level of contrast quantitatively, but one merely adjusts the knob until the picture looks best. Similarly, for some applications we expect network controls to be adjusted to achieve the best performance, rather than having the preferences specified *a priori*. Such optimizing adjustment procedures are inherently selfish.

The second problem with the cooperative view is that a universally accepted flow control algorithm chains the network to obsolete technology. The needs and desires of users will change quickly as workstation technologies and applications change. The inertial drag of universal compatibility will likely result in flow control algorithms that change slowly. Thus, we can expect that centrally mandated flow control algorithms will often lag behind the needs of users.

Lastly, there is the obvious problem of vulnerability to selfish users. Flow control algorithms based on the assumption of universal cooperation are vulnerable to isolated instances of cheating. Once users start to cheat, the performance of the system as a whole will deteriorate. In this context, cheating is not done only by those users with nefarious intent. Because flow control is often an application specific concern, many applications already bypass TCP and do their own flow control. Moreover, even seemingly benign actions such as opening several simultaneous TCP connections to achieve parallelism is a form of cheating in this context. Thus, it is hard to imagine a situation in which such cheating was not widespread.

These three objections to the cooperative model of flow control must be weighed against the inability of the noncooperative model to provide optimal efficiency. Note, however, that the noncooperative model of users is not that users are necessarily malicious, but merely that they behave in such a way as to optimize their own satisfaction. Placing the onus on the users to optimize their utility enables networks to satisfy a wide variety of service requirements. In addition, users need not have abstract knowledge of their preferences; instead, they can use simple hill-climbing techniques to find their optimal operating point. Moreover, no universal flow control algorithm is required, so innovation is unimpeded. While the game-theoretic notion of selfishness may initially appear to be, at best, a regrettable reality, in fact it may indeed be the best way to ensure good performance in the large, heterogeneous, and rapidly changing networks of the future.

[4]A hill-climbing implementation of the selfish view is where users adjust their r_i locally attempting to increase their U_i, much as in standard hill-climbing algorithms for optimization.

3 Mathematical Model of System

3.1 Switches

Our basic model is a single queue with an exponential server (of service rate 1) with each of the N users contributing an independent Poisson input stream of packets with rate r_i, $r_i > 0$. The average queue of each user's packets at the switch, c_i, depends on the service discipline used. The switch is a shared resource, and the pair of quantities $(\vec{r}, \vec{c})$ can be viewed as an allocation of that resource among the users.

This allocation cannot be arbitrary; the set of feasible allocations, those that can be realized by a nonstalling or work-conserving service discipline (one in which the server is idle only when the queue is empty), will satisfy the constraint $F(\vec{r}, \vec{c}) = 0$ where

$$F(\vec{r}, \vec{c}) \equiv \sum_{i=1}^{N} c_i - f(\vec{r})$$

with the definitions $f(\vec{r}) = g(\sum_{i=1}^{N} r_i)$ and $g(x) = \frac{x}{1-x}$ [2, 28]. This constraint merely requires that the total average queue be given by the standard M/M/1 formula. Feasible allocations will also satisfy the further constraint that, numbering the users so that the $\frac{c_i}{r_i}$ are in increasing order, the following inequalities hold for each $k \in [1, N-1]$ (these constraints must hold for all orderings of users, but it is sufficient to check the ordering where the $\frac{c_i}{r_i}$ are increasing):

$$\sum_{i=1}^{k} c_i \geq g(\sum_{i=1}^{k} r_i)$$

These subsidiary constraints embody the fact that no subset of users can have an aggregate average queue that is less than the M/M/1 result for that subset.[5]

Each service discipline gives rise to an allocation function $\vec{C}(\vec{r})$ such that the allocations $(\vec{r}, \vec{C}(\vec{r}))$ always satisfy these constraints. We will restrict the set of service disciplines considered to those that satisfy the following criteria. The allocations $(\vec{r}, \vec{C}(\vec{r}))$ must always lie in the interior of the feasible set, where the above inequalities are not saturated. Also, since the switch has no *a priori* knowledge about the users, the function $\vec{C}(\vec{r})$ must be symmetric (i.e., a permutation of the r's results in the same permutation of the c's). Furthermore, the allocation functions must be C^1 (have continuous first derivatives), with finite but possibly discontinuous second derivatives (i.e., the one-sided second derivatives always exist and are finite). Let AC denote the set of acceptable allocation functions. We will usually be restricting ourselves to the natural domain D of these functions: $D \equiv \{\vec{r} | r_i > 0 \ and \ 1 > \sum_{i=1}^{N} r_i\}$.[6] Each allocation function $\vec{C}(\vec{r})$ is realized by many service disciplines. Since we are concerned here with only the average queue length, and not higher moments thereof, we will use the terms "allocation function" and "service discipline" somewhat interchangeably.

[5]All of the results in this paper apply to any queueing system where the set of all feasible allocations can be represented by a strictly increasing and strictly convex function g in the above constraint expressions. This would include nonpremptive M/M/1 systems, as well as M/G/1 systems.

[6]However, for technical reasons arising in Section 4.2.2, we actually define the allocation functions on all of $\Re_+^n$, and allow the allocated c_i to take on infinite values.

As examples, and for future reference, let us define two sample allocation functions. First, consider the *proportional* allocation, where each user's average queue is proportional to their throughput r_i. This allocation is realized by the FIFO service discipline, and is given by the well-known formula

$$C_i^P(\vec{r}) = \frac{r_i}{1 - \sum_{j=1}^{N} r_j}$$

Another example, first introduced in [32], is the *Fair Share* allocation function. With the users labeled so that their r_i's are in increasing order, the k'th user's allocation is defined in terms of the preceding allocations: $C_k^{FS}(\vec{r})$ is the solution to

$$F((r_1\ ,\ r_2\ ,\ r_3\ ,\ \ldots,\ r_{k-1}\ ,\ r_k\ ,\ r_k\ ,\ r_k\ \ldots)\ ,\ (C_1\ ,\ C_2\ ,\ C_3\ ,\ \ldots,\ C_{k-1}\ ,\ C_k\ ,\ C_k\ ,\ C_k\ \ldots)) = 0$$

Specifically, $C_1^{FS}(\vec{r}) = \frac{g(nr_1)}{n}$, $C_2^{FS}(\vec{r}) = C_1^{FS}(\vec{r}) + \frac{g((n-1)r_2+r_1)-g(nr_1)}{n-1}$, and, more generally,

$$C_k^{FS}(\vec{r}) = C_{k-1}^{FS}(\vec{r}) + \frac{g((n-k+1)r_k + r_{k-1} + \ldots + r_1) - g((n-k+2)r_{k-1} + r_{k-2} + \ldots + r_1)}{n-k+1}$$

This is C^1 everywhere in D, and is locally C^2 whenever $r_i \neq r_j$ for all $i \neq j$. Also, $\frac{\partial C_i^{FS}}{\partial r_i} > 0$, $\frac{\partial^2 C_i^{FS}}{\partial r_i^2} > 0$, $\frac{\partial C_i^{FS}}{\partial r_j} \geq 0$, and for $i \neq j$ we have $\left(\frac{\partial C_i^{FS}}{\partial r_j} > 0\right) \Leftrightarrow (r_j < r_i)$. Thus C_i^{FS} locally depends only on those r_j such that $r_j \leq r_i$; i.e., small variations in r_j will affect C_i^{FS} if and only if $r_j \leq r_i$. The resulting triangularity of the matrix $\frac{\partial C_i^{FS}}{\partial r_j}$ is crucial in deriving the properties of the Fair Share service discipline, and reflects the partial insularity this service discipline provides.

This allocation is realized by a preemptive priority queueing algorithm best explained through the example depicted in the table below. In this example, there are four users, labeled so that the r_i are in increasing order. All of user 1's packets are in the highest priority class, and all of the other users get the same rate r_1 of packets in the highest priority class. Similarly, the rest of user 2's packets are in the second highest priority class, and all of the other users get the same rate $r_2 - r_1$ of packets in the second highest priority class. The pattern repeats until all of the throughput is assigned a priority.

FS	Priority Level			
User	A	B	C	D
1	r_1	-	-	-
2	r_1	$r_2 - r_1$	-	-
3	r_1	$r_2 - r_1$	$r_3 - r_2$	-
4	r_1	$r_2 - r_1$	$r_3 - r_2$	$r_4 - r_3$

Table 1: A priority queueing algorithm which implements the Fair Share allocation function

3.2 Users

Each user has a utility function U_i which is a function only of that user's own service allocation. We consider only those utility functions $U_i(r_i, c_i)$, which are strictly monotonic in both variables, increasing in r_i and decreasing in c_i. Furthermore, we require that U_i be a convex function and C^2 everywhere[7]. Denote by AU the set of acceptable utility functions. This is not a requirement placed on users (i.e., it is not something we require them to do, which would violate the user independence assumption) but is only a statement about what we consider a reasonable model of reality (and these are also standard assumptions in the economics literature).

The utility function U_i is merely a representation of the user's preference orderings of the various allocations (r_i, c_i). U_i contains no metrical information, so all subsequent results must be invariant under the transformation $U_i \mapsto G_i(U_i)$ for strictly monotonically increasing functions G_i. Note that this renders meaningless quantities that compare or combine utility functions from different users, such as the sum $\sum_{i=1}^{N} U_i$.

The assumption of user selfishness means that users self-optimize; each user i adjusts the rate parameter r_i so as to maximize the utility function U_i. We don't specify how this optimization occurs, but we assume that it is equivalent to holding the other r_j constant while r_i is varied; under these conditions if the system reaches an equilibrium it must be a *Nash equilibrium*.

Let $\vec{r}|^i\hat{r}_i$ denote the vector with the i'th element given by $\hat{r}_i$ and all other elements j given by r_j. Then, a Nash equilibrium can be defined as follows:

Definition 1 *A point $\vec{r}^{Nash}$ is a Nash equilibrium point if*

$$U_i(r_i^{Nash}, C_i(\vec{r}^{Nash})) \geq U_i(\hat{r}_i, C_i(\vec{r}^{Nash}|^i\hat{r}_i)) \; for\ all\ \hat{r}_i\ and\ for\ all\ i$$

Thus, at a Nash equilibrium no user is able to further increase her utility function by a unilateral change in her Poisson rate.

At a Nash equilibrium we have the necessary first derivative condition (FDC) $\frac{dU_i}{dr_i} = 0$ for all i. This condition can be reexpressed as:

$$M_i(r_i, c_i) = -\frac{\partial C_i}{\partial r_i}$$

with $M_i(r_i, c_i)$ defined as the ratio of marginal utilities

$$M_i(r_i, c_i) \equiv \frac{\frac{\partial U_i}{\partial r_i}}{\frac{\partial U_i}{\partial c_i}}$$

The Nash equilibrium describes, given our assumptions, the operating point of a set of selfish users. What are the properties of these Nash equilibria, and can we choose a service discipline that ensures certain desirable properties hold there? This is the subject of the next section.

4 Achieving Good Performance

In this section we discuss what properties constitute good performance, and then analyze if these properties can be realized through the choice of service discipline. It is extremely important to

[7] Actually, all of the results presented here hold without requiring differentiability; we merely use differentiability to simplify the presentation (we can use first derivative conditions rather than discrete differences).

note that these properties must exist for all possible sets of acceptable utility functions, and can refer only to the levels of user satisfaction and not to purely switch-oriented quantities. One can somewhat artificially divide the properties of "good" performance into three categories: in-equilibrium, getting-to-equilibrium, and out-of-equilibrium.

We discuss properties in these categories separately in the next three sections. We discuss, for each property, the possibility of finding a service discipline which provides it. For those properties which are achievable, we will also address to what extent the satisfying service disciplines are unique. These uniqueness results involve a restricted set of allocation functions, those which obey certain monotonicity conditions. We define a subset of AC, called MAC, of monotonic allocation functions as follows:

Definition 2 *An allocation function in AC is in MAC if*

1. $\frac{\partial C_i}{\partial r_j} \geq 0$ *for all i and j*
2. $\frac{\partial C_i}{\partial r_i} > 0$ *for all i*
3. $\left\{\frac{\partial C_i}{\partial r_j} = 0 \text{ at } \vec{r}^o\right\} \Rightarrow \left\{\frac{\partial C_i}{\partial r_j} = 0 \text{ for all } \vec{r} \text{ with } r_k \geq r^o{}_k\ ,\ k \neq i, \text{ and } r_i \leq r^o{}_i\right\}$

The first two conditions require that when r_i increases, c_i increases and no other user's queue decreases. Thus, no user benefits when another user consumes more throughput. While these two conditions might have some claim to reasonability, the third condition is purely technical: in essence, it states that whenever $\frac{\partial C_i}{\partial r_j} = 0$ $(i \neq j)$, the derivative remains 0 as r_i decreases and as r_k $(k \neq i)$ increases. These conditions are satisfied by such typical service disciplines as FIFO, LIFO, Processor Sharing, Polling, and HOL Priority; these conditions are also satisfied by the Fair Share allocation function. Whatever their merit, these conditions are used only for the uniqueness results; there may be other service disciplines outside of the MAC set that also provide our desirable properties (although we have yet to find such a counterexample).

While we assume that users are selfish and optimize their own performance, there will also be cases where users are not able to do so. A particular instance of a nonoptimizing user is one which does not vary r_i at all. Given a set of users and an allocation function, we can define various subsystems in which we hold one or more of the r_i's constant while the rest are allowed to vary. The induced allocation function for a subsystem is merely the same function $\vec{C}(\vec{r})$ but with some of its variables treated as constants. It is reasonable to require that these induced allocation functions have the same set of desired properties. Note that if the original allocation function $\vec{C}(\vec{r})$ is in MAC then the induced allocation function is also in MAC for the various subsystems.

We now describe the three aspects of "good" performance in more detail.

4.1 Nash Equilibria

Our basic premise is that users are selfish, and so that all equilibria of the system are Nash equilibria. Therefore, the most obvious aspect of good performance is that these Nash equilibria have certain desirable properties. Along this line, we will discuss whether or not one can guarantee that the Nash equilibria are fair and optimally efficient.

4.1.1 Efficiency

One of the central themes of this paper is that the efficiency of the algorithm should be measured strictly in terms of user satisfaction. The standard criterion for efficiency in these circumstances is that of Pareto optimality.

Definition 3 *An allocation* $(\vec{r}, \vec{c})$ *is* Pareto optimal *if there is no other feasible allocation* $(\vec{\bar{r}}, \vec{\bar{c}})$ *such that*

1. $U_i(r_i, c_i) \leq U_i(\bar{r}_i, \bar{c}_i)$ *for all* i
2. $U_i(r_i, c_i) < U_i(\bar{r}_i, \bar{c}_i)$ *for at least one* i

The set of Pareto optimal allocations is sometimes referred to as the *noninferior* set; there are no other allocations that every user would prefer. Equivalently, a point $(\vec{r}, \vec{c})$ is Pareto optimal iff there exists a vector $\vec{W}$, $W_i \geq 0$ and $\sum_{i=1}^{N} W_i = 1$, such that $\sum_{i=1}^{N} W_i U_i(r_i, c_i) \geq \sum_{i=1}^{N} W_i U_i(\bar{r}_i, \bar{c}_i)$ for all feasible $(\vec{\bar{r}}, \vec{\bar{c}})$. Thus, each Pareto allocation maximizes at least one weighted sum of utilities. Using this definition of Pareto, we can see that an interior allocation is Pareto efficient only if the following first derivative condition (FDC) holds:

$$M_i(r_i, c_i) = Z_i(r_i, c_i)$$

with the definition

$$Z_i(r_i, c_i) \equiv \frac{\frac{\partial F}{\partial r_i}}{\frac{\partial F}{\partial c_i}} = -\frac{\partial f}{\partial r_i} = -\left(1 - \sum_{j=1}^{N} r_j\right)^{-2}$$

Can we find an allocation function $\vec{C}(\vec{r})$ such that, for every admissible utility profile[8] $\vec{U} \in AU^N$, the Nash equilibrium is Pareto optimal? The answer is no (a similar result[9] is discussed in [23], page 1023).

Theorem 1 *There is no allocation function in* MAC *such that every Nash equilibrium is Pareto optimal.*

In light of the negative result of Theorem 1, one might attempt to generalize the allocation functions by including extra parameters by which users might more effectively signal their preferences. Consider, for instance, allocation functions $\vec{\hat{C}}(\vec{r}, \vec{\alpha})$ where $\vec{\alpha}$ is a vector of user specified signalling parameters. The impossibility result of Theorem 1 applies to these allocation functions as well.

Corollary 1 *Consider allocation functions* $\vec{\hat{C}}(\vec{r}, \vec{\alpha})$ *that are* C^2 *and, for every constant* $\vec{\alpha}$, *are in* MAC. *There is no allocation function of this form such that every Nash equilibrium is Pareto optimal and such that the set of all Nash equilibria has nonempty interior.*

[8] We are using the terms utility profile, utility vector, and set or collection of utility functions interchangeably, and they all refer to the set of utility functions corresponding to the set of users. These utility profiles must always be a vector in AU^N, the n-fold Cartesian product of AU.

[9] In that result, the constraint function has a different character which rules out the "separable" forms discussed in our proof and in Corollary 2.

However, if we combine this extra parameter $\vec{\alpha}$ with stalling service disciplines, where we replace the constraint $F(\vec{r}, \vec{C}(\vec{r}, \vec{\alpha})) = 0$ with $F(\vec{r}, \vec{C}(\vec{r}, \vec{\alpha})) \geq 0$, we can find allocation functions whose Nash equilibria are all Pareto optimal. See [33] for an explicit construction. This family of mechanisms, however, represents a much more general allocation mechanism than just the design of nonstalling service disciplines $\vec{C}(\vec{r})$.

The goal of guaranteeing Pareto optimality of Nash equilibria solely through the design of the service discipline $\vec{C}(\vec{r})$ might, in general, seem overly ambitious. However, the negative result of Theorem 1 is actually a property of our particular constraint function. If our problem had instead the constraint function $\hat{F}(\vec{r}, \vec{c}) \equiv \sum_{i=1}^{N} c_i - \hat{f}(\vec{r})$ with $\hat{f} = \sum_{i=1}^{N} {r_i}^2$, then every Nash equilibrium would be Pareto optimal under the allocation function $C_i(\vec{r}) = {r_i}^2$. This result generalizes to the following Corollary, where we consider the constraint $\sum_{i=1}^{N} c_i = \hat{f}(\vec{r})$ without any of the subsidiary inequality constraints of our original problem:

Corollary 2 *: Consider a convex*[10] C^2 *constraint function* $\hat{f}(\vec{r})$*:*

1. *If* $\hat{f}(\vec{r})$ *can be expressed as* $\hat{f}(\vec{r}) = \frac{1}{N-1} \sum_{i=1}^{N} h_i(\vec{r})$ *with* $\frac{\partial h_i}{\partial r_i} = 0$ *and* $\hat{f}(\vec{r}) - h_i(\vec{r}) \geq 0$*, then there is an allocation function* $\vec{C}(\vec{r})$ *such that every Nash equilibrium is Pareto optimal.*
2. *If* $\hat{f}(\vec{r})$ *cannot, in any open neighborhood, be expressed as* $\hat{f}(\vec{r}) = \frac{1}{N-1} \sum_{i=1}^{N} h_i(\vec{r})$ *with* $\frac{\partial h_i}{\partial r_i} = 0$*, then there is no* MAC *allocation function* $\vec{C}(\vec{r})$ *such that every Nash equilibrium is Pareto optimal.*

Theorem 1 states that there is no allocation function such that, for all sets of utility functions in AU (recall that AU is the set of acceptable utility functions), every Nash equilibrium is Pareto optimal. However, for a given set of utility functions, it is possible to design a service discipline whose Nash equilibria are Pareto optimal. Furthermore, if all users have the same utility function, then the Nash equilibria of the Fair Share allocation mechanism are always Pareto optimal. This result generalizes to the following theorem.

Theorem 2 *Consider an allocation function in* MAC*, and a vector of utility functions in* AU^N*.*

1. *If the Nash equilibrium is Pareto optimal, then* $r_i = r_j$ *for all* i, j*.*
2. *In all subsystems, any completely symmetric* $\vec{r}$ *that gives rise to a Pareto optimal allocation is also a Nash equilibrium of the Fair Share allocation function.*

Thus, while it is impossible to guarantee Pareto optimal Nash equilibria, there are occasions when Nash equilibria are Pareto optimal. The Fair Share allocation mechanism achieves all such possible Nash/Pareto points. In contrast, the proportional allocation, and any other allocation function that always has $\frac{\partial C_i}{\partial r_j} > 0$, never has Pareto optimal Nash equilibria.

4.1.2 Fairness

Another obviously desirable property is fairness. There are several definitions of fairness in the economics literature [36]. The most relevant for our purposes is the *envy-free* definition. User i is

[10] The set of allocations $\vec{r}, \vec{c}$ satisfying $\sum_{i=1}^{N} c_i \geq \hat{f}(\vec{r})$ must be a convex set.

said to envy user j if $U_i(r_i, c_i) < U_i(r_j, c_j)$. Note that envy does not involve a comparison between two users' utility functions, merely a comparison between their two allocations using the preference ordering of one of the users. An allocation is deemed fair if it is envy-free (i.e., no user envies another).

Given our model of user behavior, we clearly desire that Nash equilibria be envy-free. However, as we discuss in Section 4.2.2, we may not always be at a Nash equilibrium. A stronger condition is that whenever user i has chosen r_i so as to maximize U_i, user i envies no one. Essentially this states that no matter what anyone else does, a user, if she maximizes her own utility, will envy no one. We call this property being unilaterally envy-free. Unilaterally envy-free allocation functions have envy-free Nash equilibria.

Definition 4 *An allocation function is unilaterally envy-free if*

$$\left\{U_i(r_i, C_i(\vec{r})) \geq U_i(\hat{r}_i, C_i(\vec{r}|^i \hat{r}_i)) \text{ } for \text{ } all \text{ } \hat{r}_i\right\} \Rightarrow \{U_i(r_i, C_i(\vec{r})) \geq U_i(r_j, C_j(\vec{r})) \text{ } for \text{ } all \text{ } j \text{ }\}$$

Theorem 3 *:*

1. *The Fair Share allocation function is unilaterally envy-free in all subsystems.*
2. *Fair Share is the only MAC allocation function which is unilaterally envy-free.*

4.2 Converging to Nash Equilibrium

In addition to considering the properties at a Nash equilibrium, we must also address how the system reaches equilibrium. To that end, we discuss under what circumstances these equilibria are unique. We also investigate both the viability of generalized hill-climbing optimization techniques, and also the resulting rate of convergence to equilibrium using simple incremental hill-climbing optimization techniques.

4.2.1 Uniqueness of Nash Equilibrium

Clearly, if there is more than one Nash equilibrium for a given profile of utility functions, then there is an unavoidable ambiguity in the resulting noncooperative outcome. This is to be avoided, since such ambiguity leads to "super-games" whereby the users attempt to affect which equilibrium the system reaches. In [23][11], we show that the Fair Share allocation function ensures that there is always one and only one Nash equilibrium; moreover, we show that it is the only *MAC* service discipline with this property.

Theorem 4 *(from [23]):*

1. *The Fair Share mechanism always has a unique Nash equilibrium.*
2. *Fair Share is the only MAC allocation function for which every Nash equilibrium is unique.*

[11] The assumptions in that paper are slightly different than here, but the same proof techniques apply with only minor and straightforward modifications. Also, the characterization result (the result in [23] corresponding to the second part of Theorem 4) is significantly stronger than what we quote here. Note that in the economics literature, the Fair Share allocation function is referred to as the serial cost sharing method, a terminology introduced in [23].

4.2.2 Robust Convergence

Previously we have considered the existence of, and the properties of, Nash equilibria. We now address the actual dynamics of self-optimization. The most naive self-optimization algorithm is a simple incremental hill-climbing technique, with users sampling the change in U_i in response to small changes in r_i. Against a fixed environment, the dynamics of a single hill-climbing optimizer are relatively robust. However, when all the users are performing this optimization simultaneously the dynamics are considerably more complicated. In particular, the dynamics depend on the time constants used in the sampling procedure (these time constants control how long a particular value of r_i is sampled when determining its payoff). A more sophisticated user could use a longer time constant in her sampling rate, allowing the other users to equilibrate before varying her own r_i. In essence, this sophisticated user would become a leader, with the other less sophisticated users following; the leader would choose to sample a value for r_i and the "following" users, because they had much shorter time scales, would quickly equilibrate to a Nash equilibrium in the $n-1$ person subsystem. The leader could then choose her r_i based on which of these $n-1$ person subsystem Nash equilibria maximized her utility. Furthermore, simple hill-climbing users could be exploited if a sophisticated user had information about other users' utility functions, and thus knew how their flow control would react. Both of these situations give rise to another type of equilibrium, known as the Stackelberg equilibrium.

Definition 5 *Let user 1 be the leader, and let $\vec{\bar{r}}(r_1)$ be a function such that (1) the first component of the vector is given by the argument, and (2) for all $i > 1$, $U_i(\bar{r}_i, C_i(\vec{\bar{r}})) \geq U_i(\hat{r}_i, C_i(\vec{\bar{r}}|^i \hat{r}_i))$ for all $\hat{r}_i$. We then consider it a Stackelberg equilibrium with user 1 leading if*

$$U_1(r_1, C_1(\vec{\bar{r}}(r_1)) \geq U_1(\hat{r}_1, C_1(\vec{\bar{r}}(\hat{r}_1))) \quad \textit{for all} \quad \hat{r}_1$$

The leader's utility in Stackelberg equilibrium is never less than it is in the corresponding Nash equilibrium. If users benefit by becoming the leader, they might devote considerable effort to exploiting other users by either employing sophisticated optimizing strategies or attempting to discover other users' utility functions. This is not desirable, and can be avoided if the Stackelberg equilibria are also Nash equilibria. Users employing simple optimization strategies would then be protected from those using sophisticated techniques. Thus, one desires that allocation functions give rise to Nash equilibria that are also Stackelberg equilibria.

More generally, the convergence to the Nash equilibrium should be robust, in that convergence would be assured as long as users employed *any* reasonable form of self-optimization. In order to consider this question, we model optimization as a process of reducing the candidate values for r_i; each user initially considers the entire interval of values $[0, 1]$ and then, based on the achieved performance, eliminates some of them[12]. We say that such a learning algorithm is reasonable if it eventually eliminates all values of r_i which produce strictly worse performance than another candidate value for r_i. To be precise, let S_i^t denote the remaining candidate values for user i at

[12] Because users are acting independently here, we cannot guarantee that all candidate $\vec{r}$'s will stay within D; this is why we need the allocation functions defined outside of D even though there are singularities there. This requires that outside of D we assign infinite c_i's to some users, and that we extend the range of the utility functions to accommodate these infinite values for c_i. These extensions do not significantly alter any of the formalism presented here.

time t, and let $S^t = S^t_1 \times \ldots \times S^t_n$. Then, user i must eventually eliminate any value s such that there exists an $\hat{s} \in S^t_i$ such that $U_i(s, C_i(\vec{r}|^i s)) < U_i(\hat{s}, C_i(\vec{\hat{r}}|^i \hat{s}))$ for all $\vec{r}, \vec{\hat{r}} \in S^t$. This is an extremely weak requirement on the optimization procedure, in that it is extremely rare that the optimization procedure is required to eliminate candidate values. We will call these generalized hill climbing algorithms; we discuss some examples of such algorithms based on learning automata in [8].

If all users are using generalized hill climbing algorithms, then the system will eventually operate within some reduced set S^∞ in which no more values are eliminated. In order for convergence to be robust, we want S^∞ to be a single point (which, of necessity, must be a Nash equilibrium); that is, we want any combination of generalized hill climbing optimization algorithms to always eventually converge to the unique Nash equilibrium.

Note that this set S^∞ contains all Stackelberg equilibria. Thus, the condition that S^∞ is a single point is a much stronger statement than requiring that all Nash equilibria are also Stackelberg equilibria.

We have the following Theorem; the first part follows directly from a result in [8].

Theorem 5 *:*

1. *(from [8]) With the Fair Share allocation function, all generalized hill climbing algorithms converge to the Nash equilibrium.*
2. *Fair Share is the only MAC allocation function for which, in all subsystems, every Nash equilibrium is also a Stackelberg equilibrium.*

The fact that the Fair Share allocation function is the only MAC allocation function for which S^∞ is a single point follows trivially from Theorem 4, since all Nash equilibria must lie within S^∞.

A second consideration is that we want to enable more efficient protocols that would eliminate the hill-climbing technique altogether for those applications that wish to. An extreme example of this is to allow users to communicate their utility functions directly to the switch. Then, the allocation mechanism becomes a function of the reported utility functions. There is a function $\vec{B}$ that maps the set of reported utility functions into a set of (r, c) allocations: $(r_i, c_i) = B_i(\vec{\hat{U}})$ (where we denote reported utility functions, as opposed to the true utility functions, by a hat). Unless the allocation mechanism is carefully designed, users will have an incentive to lie about their utility function, and sophisticated users could again exploit naive ones. An allocation mechanism that encourages users to tell the truth is called a revelation mechanism, and has the following property.

Definition 6 *$\vec{B}$ is a revelation mechanism if $U_i(B_i(\vec{\hat{U}})) \leq U_i(B_i(\vec{\hat{U}}|^i U_i))$ for all $\vec{\hat{U}}$ and U_i.*

In [23] (therein see Theorem 2 and the comment on page 1028), we show that indeed the Fair Share allocation mechanism produces a revelation mechanism. Denote by $\vec{B}^{FS}$ the function that maps a vector of utility functions into the $\vec{r}$ and $\vec{c}$ resulting from the unique Nash equilibrium of the Fair Share allocation function.

Theorem 6 *(from [23]) The allocation mechanism $\vec{B}^{FS}$ is a revelation mechanism.*

When we apply stringent differentiability assumptions, we have shown in [34] that the Fair Share allocation function is the only one which has this property. Satterthwaite and Sonnenschein [31] were the first to note the crucial role of acyclicity in such revelation mechanisms.

4.2.3 Rapid Convergence

In the previous section we wanted convergence to be robust, and wanted simple hill-climbing optimization techniques to be invulnerable to more sophisticated strategies. We now impose the criterion that incremental hill-climbing techniques converge rapidly, at least in the linear regime. We assume that each user can, either through slight variations in r_i or by direct communication with the switch, determine the derivatives $\frac{\partial C_i}{\partial r_i}$ and $\frac{\partial^2 C_i}{\partial {r_i}^2}$. The quantity $E_i = M_i(r_i, C_i(\vec{r})) + \frac{\partial C_i}{\partial r_i}$ is the measure of how far from the Nash condition a user is. In the absence of any other knowledge about other users and their intentions, the simplest hill-climbing technique is to use Newton's method. Each user increments her throughput rate by an amount Δ_i, $r_i(t+1) \mapsto r_i(t) + \Delta_i$, where $\Delta_i = -\frac{E_i}{\frac{dE_i}{dr_i}}$. Assuming that the users update their r_i synchronously, the time evolution of the E_i can be described in a linear approximation by the relaxation matrix $A_{i,j}$: $E_i(t+1) = \sum_{i=1}^{N} A_{i,j} E_j(t)$, where

$$A_{i,j} = \delta(i,j) - \frac{\frac{\partial E_i}{\partial r_j}}{\frac{\partial E_j}{\partial r_j}}$$

We desire an allocation function that renders this optimization strategy stable and fast. In this linear approximation, fast (i.e., superlinear) convergence occurs when the eigenvalues of the relaxation matrix are all zero. This is equivalent to $A_{i,j}$ being nilpotent; ${A_{i,j}}^N = 0$.

Theorem 7 *:*

1. *The Fair Share discipline always has a nilpotent relaxation matrix in all subsystems.*
2. *Fair Share is the only MAC allocation function that always has a nilpotent relaxation matrix.*

From Theorem 7 we know that the proportional allocation is not always nilpotent. However, the proportional allocation may even be linearly unstable, which occurs when the stability matrix $A_{i,j}$ has eigenvalues of magnitude greater than 1. For example, in a system with N identical users each with the linear utility function $U(r,c) = r - \gamma c$, the proportional allocation gives rise to a relaxation matrix whose leading eigenvalue is $1 - N$. Recall that stability requires that all eigenvalues have magnitudes bounded by unity, so eigenvalues of the form $1 - N$, for $N > 2$, indicate instability.

4.3 Out-of-Equilibrium

The unilaterally envy-free condition guarantees that a self-optimizing user will envy no one. This, however, does not guarantee a satisfactory level of service to a self-optimizing user. If, for instance, all of the other users are self-destructive, then not envying them is little consolation. Thus, we need to address the ability of a service discipline to protect users from the actions of others, i.e. to guarantee certain levels of user satisfaction. The utilities U_i are functions of r_i and c_i; while each user has complete freedom in choosing r_i, the congestion c_i depends on the entire vector $\vec{r}$. The protective properties of a service discipline can be quantified by the upper bound on the congestion c_i as a function of r_i; $Max(C_i(\vec{r}|^i r_i))$ where the maximum is taken over all $\vec{r}$. Symmetry demands that this bound be no smaller than $C_i(\vec{e}r_i) = \frac{r_i}{1-Nr_i}$, where $\vec{e} = (1,1,1,1,1.....,1)$. We call a service discipline *protective* if it can match this bound.

Definition 7 *An allocation function is protective*[13] *if* $C_i(\vec{r}) \leq C_i(\vec{e}r_i)$ *for all* $\vec{r}$.

This protection guarantee is the best we can offer and is, in essence, the converse of the Golden Rule (a user has done unto her no worse than she does onto others). This condition prevents naive sources from being unduly harmed by other users, even if these other users are being malicious. Non-optimizing users would fare as well under a protective switch as they would under any symmetric switch competing with N other identical users. The Nash equilibria that result from protective allocation functions give, to each user i, a level of satisfaction that is at least as great as that achieved at the symmetric Pareto optimal allocation in a system with all N users having the utility function U_i.

Theorem 8 *:*

1. *The Fair Share mechanism is protective in all subsystems.*
2. *Fair Share is the only MAC allocation function that is protective in all subsystems.*

5 Discussion

5.1 Summary of Technical Results

We have considered an independent and selfish user population. One might have expected this to lead, in the typical case, to poor network performance. With service disciplines that give rise to the proportional allocation function, such as FIFO, this is indeed the case. Nash equilibria resulting from the proportional allocation function are never Pareto optimal and are not guaranteed to be fair. Users are not protected from each other, and convergence to equilibrium is not guaranteed.

However, one can avoid many of the negative effects of selfishness through the choice of service discipline. While it is impossible to guarantee that the Nash equilibrium will be Pareto optimal, one can guarantee certain other desirable properties. The Fair Share allocation function always has a unique Nash equilibrium, and this equilibrium is always fair. Moreover, this equilibrium can be reached by any set of reasonable optimization algorithms, and naive hill-climbing optimization strategies are as effective as more sophisticated ones.[14] Furthermore, these optimization techniques converge rapidly. Even out of equilibrium the Fair Share allocation function provides protection to all users in all subsystems. Perhaps most importantly, the Fair Share allocation function is the only MAC allocation function that has any one of these properties. In addition, whenever a Nash equilibrium is Pareto optimal for any allocation function in MAC, the same Nash equilibrium can be achieved with the Fair Share allocation function. On the basis of these results, it is clear that the Fair Share allocation function is the best choice, at least for allocation functions in MAC.

[13] This is closely related to the "unanimity bound" as discussed in [21, 22, 23, 24, 25].

[14] In this paper we have not addressed the issue of what happens when there is a joint effort to manipulate the system; that is, when a coalition of users acts in concert to gain an advantage. We have shown, in [23] (see page 1025), that all Fair Share Nash equilibria are resilient against such coalitional manipulations.

5.2 Applications to Real Networks

Fair Queueing is a switch service discipline for real networks (as opposed to Poisson sources and exponential servers) which is similar to the Fair Share allocation function [3] (see also [9, 15, 26]).[15] The service discipline essentially approximates a Head-of-Line Processor Sharing algorithm without using time-slicing. Simulations have indicated that even with today's rather unsophisticated protocols, and with only primitive notions of utility functions (users doing FTP's care only about throughput, and users doing Telnet's care only about delay), Fair Queueing still provides important advantages over the usual FIFO service discipline. In particular, Fair Queueing provides fair allocation of throughput, lower delay for sources using less than their full share of throughput, and protection from ill-behaved sources [3]. Thus, the results from the game-theoretic analysis presented here are consistent with what we have observed in real networks (or at least in simulations of real networks).

5.3 Related Work

There is a rather large and rapidly growing literature of game-theoretic approaches to network allocation. One segment of this literature, as exemplified by [4, 5, 6, 29], analyzes the properties of, and algorithms to compute, Pareto optimal allocations for a given vector of utility functions. Achieving efficiency in this approach relies on cooperative algorithms, and so the results are not relevant to the selfish user population considered here.

Another segment of this literature, represented by [1, 4, 10, 17, 18], discusses the nature of Nash equilibria in networks using the FIFO service discipline. While these papers considers the incentive issues we are concerned with, their approach is essentially descriptive in that they consider the traditional network architecture and then describe the equilibria that would result from selfish user behavior.

The approach we have taken here, which is also taken in [7, 16, 30, 33] combines the efficiency considerations of the first segment of the literature with the incentive considerations of the second segment of the literature. This approach not only considers what is socially desirable (efficient, fair, etc.), but also how one can achieve these goals given that users are selfish. This notion of designing service disciplines and other more general mechanisms which give socially desirable Nash equilibria is borrowed directly from the economics and game-theory literatures (see [19] for an overview).

Ferguson *et al.* [7] developed network resource allocation mechanisms whose Nash equilibria are always Pareto efficient. In addition, Ferguson *et al.* address the iterative nature of the Nash equilibration process and presents simulation results on stability. However, their network model differs from ours in three important ways. First, all gateways are assumed to have FIFO service, so that the set of feasible allocations is quite reduced. Thus, while the allocations determined by the mechanisms in [7] are Pareto optimal with respect to the restricted set of FIFO-feasible allocations, they are not, in general, Pareto optimal with respect to the complete feasible set. Second, the delay d_i used in the utility functions in [7] is not the true average delay, which depends on the entire vector

[15] The similarity between Fair Share and Fair Queueing is based only on their being derived from the same intuition of partial insularity; we make no claims about the two algorithms being mathematically related.

$\vec{r}$, but is the "worst-case" delay which is a function of r_i only. This greatly reduces the coupling between users. Third, the utility functions are rather different than the ones considered here; in [7] users care only about throughput up to a threshold, after which they care only about delay. Such utility functions, which essentially decouple throughput and delay (no user has marginal utility for both at the same time), make the incentive issues much simpler.

In the present paper we only considered allocations which were achievable by nonstalling service disciplines. In [33] we construct more general mechanisms for which all Nash equilibria are Pareto optimal. These mechanisms require that the service discipline be stalling, or non-work-conserving. Interestingly, it is the introduction of this inefficiency (the stalling) that allows the Nash equilibrium to be efficient.

Sanders [29, 30] and Keshav [16] have introduced incentive compatible reward mechanisms for switches, in which users optimize their utility when they cooperate so that the distinction between selfish and cooperative behavior disappears. Electronic market algorithms have also been proposed [20]. However, each of these algorithms require sidepayments of transferable utility, i.e. money, which is outside the scope of our treatment.

The problem we pose in this paper can be thought of as a cost sharing problem, and recently there have been several economics papers which have discussed the strategic and normative properties of various cost sharing schemes [8, 23, 24, 25, 34, 35]. These references contain several other characterizations of the Fair Share allocation function[16]. While the game theory literature typically concentrates on the properties of the Nash equilibria itself, here we introduce some out-of-equilibrium considerations as well.

5.4 Network of Switches

One area that remains unexplored is the properties of a network of such switches. There are two key problems to face. First, while we have assumed in our model that the sources are all Poisson, the output process from nontrivial service disciplines are not. Thus, one would need to characterize such output processes and the queueing behavior resulting at the next switch. This is a daunting challenge. Second, even if we use the approximation of modeling the output processes as Poisson with the same rate as the input rate, the game theoretic treatment must be generalized. The theory should reflect the fact that users care only about the total congestion; in terms of congestion in the individual switches, this can be expressed as $c_i = \sum_\alpha {c_i}^\alpha$ where α is an index that identifies the switches in the networks and i identifies the users. Straightforward generalizations of most of the single-switch results remain true for networks. However, the fairness result is rendered irrelevant; a different definition of fairness is needed to meaningfully compare allocations to users using different routes.

[16] The Fair Share allocation function is referred to there as the serial cost sharing method.

5.5 Generalizations

We have applied game-theoretic ideas to the problem of network congestion control. Much of the formalism is directly applicable to any resource sharing problem where users are concerned about both the amount and quality of service, and where the central resource produces lower quality output as the total amount of service required increases. The server is described by a constraint function which represents the feasible tradeoffs between quality and amount of service. Different constraint functions give rise to different conclusions; in particular there are some constraint functions that allow the guarantee of Pareto optimal Nash equilibria. Aside from the specific formalism, the basic ideas presented here apply equally well to a much more general class of distributed resource allocation problems, such as file location and load sharing. Often these problems are addressed through the design of efficient algorithms to compute optimal or near-optimal allocations. These algorithms suffer from the same problems that we discussed for cooperative flow control. In particular, the cooperative approach assumes both that all users are willing to cooperate and that they are able to cooperate, in that they all know which distributed algorithm to run. As systems get larger, and as computer systems become more heterogeneous, the second assumption, if not the first, becomes more questionable. The noncooperative approach outlined here offers a different vision of coordinating a mass of independent users which requires neither of these assumptions.

6 Acknowledgements

I would like to thank the following for helpful discussions during the early stages of this work: Srinivasan Keshav, Peter Linhart, Thomas Marschak, Paul Milgrom, Roy Radner, Stefan Reichelstein, Stanley Reiter, Richard Steinberg, Abel Weinrib, and Seungjin Whang. I would especially like to thank Hervé Moulin for his insightful comments. I would also like to acknowledge helpful comments from three anonymous reviewers.

References

[1] A. Bovopoulos and A. Lazar, "Decentralized Algorithms for Flow Control", Proceedings of the 25th Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, October 1987.

[2] E. Coffman Jr. and I. Mitrani, "A Characterization of Waiting Time Performance Realizable by Single Server Queues", *Operations Research*, Vol. 28, No. 3 part II , May-June 1980.

[3] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", Internetworking: Research and Experience, Vol. 1, pp. 3-26, 1990.

[4] C. Douligeris and R. Mazumdar, "A Game Theoretic Approach to Flow Control in an Integrated Environment", *Journal of the Franklin Institute*, Vol. 329, No. 3, pp. 383-402, March 1992.

[5] C. Douligeris and R. Mazumdar, "On Pareto Optimal Flow Control in a Multiclass Environment", Proceedings of the 25th Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, October 1987.

[6] C. Douligeris and R. Mazumdar, "More on Pareto Optimal Flow Control", Proceedings of the 25th Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, October 1988.

[7] D. Ferguson, C. Nikolaou, and Y. Yemini, "An Economy for Flow Control in Computer Networks", IEEE Infocom, 1989.

[8] E. Friedman and S. Shenker, "Learning by Distributed Automata", preprint, 1993. (available from parcftp.parc.xerox.com as pub/net-research/learning.ps)

[9] E. Hahne, "Round-Robin Scheduling for Max-Min Fairness in Data Networks", *IEEE JSAC*, Vol. 9, pp. 1024-1039, 1991.

[10] M.-T. Hsiao and A. Lazar, "A Game Theoretic Approach to Decentralized Flow Control of Markovian Queueing Networks", in Performance '87, P.-J. Courtois and G. Latouche eds., North Holland, 1988.

[11] V. Jacobson, "Congestion Avoidance and Control", ACM Sigcomm Proceedings, pp. 314-329, 1988.

[12] J. Jaffe, "Bottleneck Flow Control". *IEEE Transactions on Communications*, Vol. 29, No. 7, pp. 954-962, July 1981.

[13] R. Jain and K. K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals, and Alternatives", Proceedings of the Computer Networking Symposium, pp. 134-143, 1988.

[14] R. Jain, "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks", *Computer Communications Review*, Vol. 19, No.5, pp. 56-71, 1989.

[15] M. Katevenis, "Fast Switching and Fair Control of Congested Flow in Broadband Networks", *IEEE JSAC*, Vol. 5, pp. 1315-1326, 1987.

[16] S. Keshav, "A Mechanism for Congestion Control in Computer Networks", preprint, 1989. (available from research.att.com as dist/qos/mechanism.ps)

[17] Y. Korilis and A. Lazar, "Why is Flow Control Hard: Optimality, Fairness, Partial and Delayed Information", preprint, 1992. (available from ftp.ctr.columbia.edu as CTR-Research/comet/public/papers/92/KOR92.ps.z)

[18] Y. Korilis and A. Lazar, "On the Existence of Equilibria in Noncooperative Optimal Flow Control", Proceedings of ITC Workshop (Bangalore, India), pp. 243-251, 1993. (also available from ftp.ctr.columbia.edu as CTR-Research/comet/public/papers/93/KOR93.ps.gz)

[19] E. Maskin, "The Theory of Implementation in Nash Equilibrium", in Social Goals and Organization: Essays in Memory of Elisha Pazner, Hurwicz, Schmeidler, and Sonnenschein eds., Cambridge University Press, 1985.

[20] M. Miller and K. Drexler, "Markets and Computation: Agoric Open Systems", in The Ecology of Computation, B. Huberman editor., North Holland, 1988.

[21] H. Moulin, "Joint Ownership of a Convex Technology: Comparison of Three Solutions", *Review of Economic Studies*, Vol. 57, pp. 439-452, 1990.

[22] H. Moulin, "Uniform Externalities: Two Axioms for Uniform Allocation", *Journal of Public Economics*, Vol. 43, pp. 305-326, 1990.

[23] H. Moulin and S. Shenker, "Serial Cost Sharing", *Econometrica*, Vol. 60, pp. 1009-1037.

[24] H. Moulin and S. Shenker, "Average Cost Pricing versus Serial Cost Sharing: An Axiomatic Comparison", *Journal of Economic Theory*, Vol. 64, No 1, pp. 178-201, 1994.

[25] H. Moulin, "Cost Sharing under Increasing Returns", *Games and Economic Behavior*, forthcoming.

[26] J. Nagle, "On Packet Switches with Infinite Storage", *IEEE Transactions on Communications*, Vol. 35, pp. 435-438, 1987.

[27] K. K. Ramakrishnan and R. Jain, "An Binary Feedback Scheme for Congestion Avoidance in Computer Networks", *ACM Transactions on Computer Systems*, Vol. 8, pp. 158-181, 1990.

[28] J. Regnier, "Priority Assignment in Integrated Services Networks", Report LIDS-TH-1565, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts, December, 1986.

[29] B. Sanders, "An Asynchronous Distributed Flow Control Algorithm for Rate Allocation in Computer Networks", *IEEE Transactions on Computers*, Vol. 37, No. 7, pp. 779-787, July 1988.

[30] B. Sanders, "An Incentive Compatible Flow Control Algorithm for Rate Allocation in Computer Networks", *IEEE Transactions on Computers*, Vol. 37, No. 9, pp. 1067-1072, September 1988.

[31] M. Satterthwaite and H. Sonnenschein, "Strategy-Proof Allocation Mechanisms at Differentiable Points", *Review of Economic Studies*, Vol. 48 , pp. 587-597, 1981.

[32] S. Shenker, "A Theoretical Analysis of Feedback Flow Control", Sigcomm '90, 1990.

[33] S. Shenker, "Efficient Network Allocations with Selfish Users", Performance '90, 1990.

[34] S. Shenker, "On the Strategy-Proof and Smooth Allocation of Private Goods in a Production Economy", preprint, 1993. (available from parcftp.parc.xerox.com as pub/net-research/otss.ps)

[35] S. Shenker, "Some Technical Results on Continuity, Strategy-Proofness, and Related Strategic Concepts", preprint, 1993. (available from parcftp.parc.xerox.com as pub/net-research/str.ps)

[36] W. Thomson and H. Varian, "Theories of Justice Based on Symmetry", in Social Goals and Organization: Essays in Memory of Elisha Pazner, Hurwicz, Schmeidler, and Sonnenschein eds., Cambridge University Press, 1985.

A Appendix

Before proceeding, we present several Lemmas.

Lemma 1 *The Fair Share allocation is the only allocation function in* MAC *such that* $\frac{\partial C_i}{\partial r_j} = 0$ *whenever* $r_j = r_i$, $i \neq j$.

Proof of Lemma 1: Let $\vec{C}(\vec{r})$ be an allocation function in MAC such that $\frac{\partial C_i}{\partial r_j} = 0$ whenever $r_j = r_i$, $i \neq j$. Consider some $\vec{r}$ (labeled so that the r_i are in increasing order). At the point $(r_1, r_1, r_1, r_1, ...)$ we know that $\frac{\partial C_i}{\partial r_j} = 0$ for all $i \neq j$. The properties of MAC then imply that we can increase the other components r_i, $i \geq 2$, without changing C_1: $C_1(\vec{r}) = C_1(r_1, r_1, r_1, r_1, ...) = {C^{FS}}_1(\vec{r})$. Now, consider the point $(r_1, r_2, r_2, r_2, ...)$. As before, $\frac{\partial C_i}{\partial r_j} = 0$ for all $i \neq j$, $i, j \geq 2$, so we can increase the components r_i, $i \geq 3$, without changing C_2: $C_2(\vec{r}) = C_2(r_1, r_2, r_2, r_2, ...) = {C^{FS}}_2(\vec{r})$. We can continue this procedure for all the components. Thus, we must have $\vec{C}(\vec{r}) = \vec{C}^{FS}(\vec{r})$ for all $\vec{r}$. □

Lemma 2 *For any allocation function in* MAC, $\left\{\frac{\partial C_i}{\partial r_j} = 0 \;\; for\ all\ i \neq j\right\} \Rightarrow \{r_j = r_i \ for\ all\ i, j\}$.

Proof of Lemma 2: Consider an allocation function $\vec{C}$ in MAC. Assume that there exists a point $\vec{\hat{r}}$ such that $\frac{\partial C_i}{\partial r_j} = 0$ for all $i \neq j$ and that $\hat{r}_2 > \hat{r}_1$. In what follows, restrict r_1 and r_2 to be in the range $[\hat{r}_1, \hat{r}_2]$, and consider all other components fixed: $r_j = \hat{r}_j$ for all $j > 2$. From condition (3) in the definition of MAC, we know that the C_i's for $i \neq 1$ remain invariant as we vary r_1 in this range. The constraint equation gives us an expression for C_1 as a function of r_1: $C_1(\vec{\hat{r}}|^1 r_1) = C_1(\vec{\hat{r}}) + f(\vec{\hat{r}}|^1 r_1) - f(\vec{\hat{r}})$. Furthermore, when we decrease r_2 below $\hat{r}_2$, none of the C_j can increase. This leads to the inequalities $C_1(\vec{\hat{r}}|^1 r_1) \geq C_1(\vec{\hat{r}}|^{1,2} r_1, r_2)$ and $f(\vec{\hat{r}}|^{1,2} r_1, r_2) \leq f(\vec{\hat{r}}|^1 r_1) + C_1(\vec{\hat{r}}|^{1,2} r_1, r_2) + C_2(\vec{\hat{r}}|^{1,2} r_1, r_2) - C_1(\vec{\hat{r}}|^1 r_1) - C_2(\vec{\hat{r}}|^1 r_1)$. Setting $r_1 = r_2$ and combining these expressions, we find that $2f(\vec{\hat{r}}|r_1) - f(\vec{\hat{r}}|^{1,2} r_1, r_1) - f(\vec{\hat{r}}) - C_2(\vec{\hat{r}}) + C_1(\vec{\hat{r}}) \geq 0$. Denoting the left-hand side of this inequality by $g(r_1)$, we can see that (1) $g(\hat{r}_2) = 0$, (2) $g'(\hat{r}_2) = 0$, and (3) $g''(\hat{r}_2) < 0$. Thus, the inequality is necessarily violated for $r_2 = \hat{r}_2 - \epsilon$ with sufficiently small ϵ, thereby reaching a contradiction. □

For the next Lemma, we need the following definition. For any matrix $X_{i,j}$, a k-cycle is a nonrepeating set of indices i_1 , i_2 , ... , i_k (setting $i_{k+1} = i_1$) such that all elements $X_{i_j, i_{j+1}} \neq 0$. A matrix is acyclic if there are no k-cycles for $k \geq 2$.

Lemma 3 *The Fair Share allocation is the only allocation function in* MAC *such that the matrix* $\frac{\partial C_i}{\partial r_j}$ *is always acyclic.*

Proof of Lemma 3: From symmetry, $\{r_i = r_j\} \Rightarrow \left\{\frac{\partial C_j}{\partial r_i} = \frac{\partial C_i}{\partial r_j}\right\}$. Thus, for the matrix $\frac{\partial C_i}{\partial r_j}$ to be always acyclic, we must have $\frac{\partial C_i}{\partial r_j} = 0$ whenever $r_j = r_i$ with $i \neq j$. From Lemma 1 we know that this uniquely characterizes the Fair Share allocation function in MAC. □

Lemma 4 *Consider a set of utility functions in* AU. *Under the Fair Share mechanism, every point satisfying the Nash FDC is a Nash equilibrium.*

Proof of Lemma 4: The convexity of U_i and the fact that $\frac{\partial^2 {C^{FS}}_i}{\partial {r_i}^2} > 0$ imply that, for any $\vec{\hat{r}}$, the function $g_i(r_i) \equiv U_i(r_i, \vec{C}(\vec{\hat{r}}|^i r_i))$ is concave. Thus, the Nash FDC is sufficient to guarantee the satisfaction of the Nash condition. □

Lemma 5 *Consider an allocation function* $\vec{C}(\vec{r})$ *in* MAC. *For any point* $\vec{r} \in D$, *there is a set of utility functions in* AU *such that this point* $\vec{r}$ *is a Nash equilibrium.*

Proof of Lemma 5: Recall that a point $\vec{\bar{r}}$ is a Nash equilibrium iff the functions $g_i(r_i) \equiv U_i(r_i, \vec{C}(\vec{\bar{r}}|^i r_i))$ all have their global maxima at $r_i = \bar{r}_i$. Consider some point $\vec{\bar{r}} \in D$ and define $\vec{\bar{c}} = \vec{C}(\vec{\bar{r}})$ and the set of utility functions $U_i(r,c) = {U^r}_i(r) + {U^c}_i(c)$ with ${U^r}_i(r) = -\frac{{\alpha_i}^2}{\beta_i} e^{\frac{-\beta_i}{\alpha_i}(r-\bar{r}_i)}$ and ${U^c}_i(c) = -\frac{{\gamma_i}^2}{\nu_i} e^{\frac{\nu_i}{\gamma_i}(c-\bar{c}_i)}$, where α, β, γ, and ν are all positive constants . These utility functions are in AC. Set the parameters α_i and γ_i such that $\frac{\alpha_i}{\gamma_i} = \frac{\partial C_i}{\partial r_i}$, so that the Nash FDC is satisfied. Note that when $-(\beta_i {\gamma_i}^2 + \nu_i {\alpha_i}^2){\gamma_i}^{-3} < \frac{\partial^2 C_i}{\partial {r_i}^2}$, the values $\bar{r}_i$ are all local maxima of the functions $g_i(r_i)$. Given the allocation function, we can then find sufficiently large values of β_i and ν_i such that the values $\bar{r}_i$ are all global maxima of the functions $g_i(r_i)$, making $\vec{\bar{r}}$ a Nash equilibrium. To see this, consider the limit of very large β_i and ν_i. There are two categories of points: (1) $r_i > \bar{r}_i$ and so $C_i(\vec{\bar{r}}|r_i) > C_i(\vec{\bar{r}})$, and (2) $r_i < \bar{r}_i$ and so $C_i(\vec{\bar{r}}|r_i) < C_i(\vec{\bar{r}})$. For points in category (1), ${U^r}_i(r_i) \approx {U^r}_i(\bar{r}_i)$ and ${U^c}_i(C_i(\vec{\bar{r}}|r_i)) \ll {U^c}_i(C_i(\vec{\bar{r}}))$. For points in category (2), ${U^r}_i(r_i) \ll {U^r}_i(\bar{r}_i)$ and ${U^c}_i(C_i(\vec{\bar{r}}|r_i)) \approx {U^c}_i(C_i(\vec{\bar{r}}))$. Thus, for all points in categories (1) and (2), $g_i(r_i) < g_i(\bar{r}_i)$. □

Proof of Theorem 1: The reasoning here follows closely, until the final step, that in [23], page 1023. Assume, to the contrary, that $\vec{C}(\vec{r})$ is an allocation function in MAC such that every Nash equilibrium is also Pareto optimal. From Lemma 5, for any point $\vec{r} \in D$ we know that we can find a vector of utility functions $\vec{U} \in AU^N$ that has this point as a Nash equilibrium. Combining the Pareto and Nash FDC conditions, the allocation function must satisfy the relation $Z_i(r_i, c_i) = -\frac{\partial C_i}{\partial r_i}$ throughout D. Note that this condition no longer involves the utility functions, but is merely a property of the constraint and the allocation function. For our constraint function, $Z_i(r_i, d_i) = -\frac{\partial f}{\partial r_i}$ so the necessary FDC condition becomes $\frac{\partial C_i}{\partial r_i} = \frac{\partial f}{\partial r_i}$. Upon integrating we can express the allocation function in terms of a set of functions h_i: $C_i = f - h_i$, where $\frac{\partial h_i}{\partial r_i} = 0$. Demanding that the allocation function satisfy the constraint yields the relation $(N-1)f(\vec{r}) = \sum_{i=1}^{N} h_i(\vec{r})$. This is clearly not satisfied by our constraint function (consider, for instance, applying the partial derivative $\frac{\partial^n}{\partial r_1 \partial r_2 \ldots \partial r_n}$ to both sides of the equation). □

Proof of Corollary 1: When we add the parameters $\vec{\alpha}$ to the allocation function, the Pareto FDC's are unchanged (since they are properties of the allocations, not the mechanism) and the Nash FDC's are merely augmented by the additional condition $\frac{\partial C_i}{\partial \alpha_i} = 0$. As long as there is an open neighborhood of points that are all Nash equilibria of some utility profile (which there must be if the set of Nash equilibria has nonempty interior), then the proof of Theorem 1 applies to that neighborhood as well. The existence of an allocation mechanism for which every Nash equilibrium is Pareto optimal would imply the existence of functions $h_i(\vec{r}, \vec{\alpha})$ such that, throughout this neighborhood, $C_i = f - h_i$ where $\frac{\partial h_i}{\partial \alpha_i} = \frac{\partial h_i}{\partial r_i} = 0$. Again, the constraint condition yields $(N-1)f(\vec{r}) = \sum_{i=1}^{N} h_i(\vec{r}, \vec{\alpha})$ which cannot hold for our particular constraint function $f(\vec{r})$. □

Proof of Corollary 2:

1. Let $C_i(\vec{r}) = f(\vec{r}) - h_i(\vec{r})$. The condition that $f(\vec{r}) - h_i(\vec{r}) \geq 0$ assures that this $\vec{C}(\vec{r})$ maps into the set of nonnegative allocations. At every Nash equilibrium the Nash FDC must hold. With this choice of $\vec{C}(\vec{r})$, the Nash FDC implies the Pareto FDC. The Pareto FDC, in turn, implies Pareto optimality because of the convexity condition on the feasible set.

2. This result follows from the proof of Theorem 1.

□

Proof of Theorem 2:

1. From the proof of Theorem 1, we know that at a Nash/Pareto point we have the FDC $\frac{\partial C_i}{\partial r_i} = \frac{\partial f}{\partial r_i}$. The feasibility condition on allocation functions requires that $\sum_{i=1}^{N} C_i = f(\vec{r})$. Taking the derivative of this expression with respect to r_j and then combining it with the Nash/Pareto FDC, we find that $\sum_{i \neq j} \frac{\partial C_i}{\partial r_j} = 0$. Since allocation functions in MAC have $\frac{\partial C_i}{\partial r_j} \geq 0$, we must have $\frac{\partial C_i}{\partial r_j} = 0$ for all $j \neq i$. Thus, following Lemma 2, we have $r_i = r_j$ for all i, j.

2. At such a symmetric point, the delay values are completely determined by the constraint so the Fair Share allocation function realizes the same Pareto optimal allocation. The question is whether or not this point is a Nash equilibrium for the Fair Share allocation function. The Fair Share mechanism satisfies the Nash FDC conditions, since $\frac{\partial C_i}{\partial r_j} = 0$ for all $j \neq i$ at this point, which is sufficient, from Lemma 4, to guarantee that this point is a Nash equilibrium. Thus, any completely symmetric $\vec{r}$ that gives rise to a Pareto optimal allocation is also a Nash equilibrium of the Fair Share allocation function.

□

Proof of Theorem 3:

1. Assume that user 1 has the utility function U in AU. User 1 will envy user 2 if the function $E(r_1, r_2) = U(r_2, {C^{FS}}_2) - U(r_1, {C^{FS}}_1)$ is positive. For a given r_2, let r_1 assume the value that maximizes $U(r_1, {C^{FS}}_1)$; call this value r_{um}, the unilateral maximum. At this point we have the FDC $\frac{d}{dr_1} U(r_1, {C^{FS}}_1) = 0$. There are two cases. If $r_{um} \geq r_2$ then there is no envy because $E(r_{um}, r_2) \leq E(r_2, r_2) = 0$. We now need to show that if $r_{um} < r_2$, then $E(r_{um}, r_2) \leq 0$. Let us maximize this envy by fixing $r_1 = r_{um}$ and varying r_2 between r_{um} and 1. Note that varying r_2 does not affect ${C^{FS}}_1$ under the Fair Share allocation function, so that maximizing the envy is equivalent to maximizing the function $U(r_2, {C^{FS}}_2)$. But, from the definition of r_{um}, we know that this function satisfies the FDC $\frac{d}{dr_2} U(r_2, {C^{FS}}_2) = 0$ at $r_2 = r_{um}$ and also that $\frac{d^2}{d{r_2}^2} U(r_2, {C^{FS}}_2) \leq 0$ for r_2 between r_{um} and 1. Thus, when $r_{um} < r_2$, we have no envy since $E(r_{um}, r_2) \leq E(r_{um}, r_{um}) = 0$. This proof also applies to all subsystems.

2. Consider a unilaterally envy-free allocation function in MAC. As in the proof of part (1), we fix the variables r_k for $k \geq 3$ and will vary r_1 and r_2 in an attempt to make user 1 envious of user 2. Consider the two user subsystem with both users having this same utility function $U(r, c)$. Lemma 5 implies that, for any symmetric point ($r_1 = r_2$) in this two user subsystem, we can choose U so that the point is a Nash equilibrium. Consider any symmetric point $r_1 = r_2 = r_{Nash}$, define $c_{Nash} = C_1(\vec{r}|^{1,2} r_{Nash}, r_{Nash})$, and choose, as in Lemma 5, $U(r, c) = -\frac{\alpha^2}{\beta} e^{\frac{-\beta}{\alpha}(r - r_{Nash})} - \frac{\gamma^2}{\nu} e^{\frac{\nu}{\gamma}(c - c_{Nash})}$ with the parameters α, β, γ, and ν chosen appropriately ($\frac{\alpha}{\gamma} = \frac{\partial C_1}{\partial r_1}$, and β,ν sufficiently large). Define the function $\bar{r}_1(r_2)$ as the value of r_1 that, for a given r_2, maximizes $U(r_1, C_1(r_2, r_1))$. This function is smooth in the neighborhood of the point $r_2 = r_{Nash}$. We know that $\bar{r}_1(r_{Nash}) = r_{Nash}$. Note that $\frac{d}{dr_1} U(r_1, C_1(r_2, r_1)) = 0$ when evaluated at $r_1 = \bar{r}_1(r_2)$. The amount of envy is described by the function

$$E(r_2) \equiv U(r_2, C_2(\bar{r}_1(r_2), r_2)) - U(\bar{r}_1(r_2), C_1(\bar{r}_1(r_2), r_2))$$

Clearly, $E(r_{Nash}) = 0$. If we can find a value of r_2 such that $E(r_2) > 0$ then the allocation function is not unilaterally envy-free. If the system is unilaterally envy-free then we must have $\frac{d}{dr_2}E(r_2) = 0$ at the Nash point $r_2 = r_{Nash}$. This requires that (with all quantities evaluated at $r_1 = r_2 = r_{Nash}$)

$$0 = \frac{\partial U}{\partial r}\frac{\partial C_2}{\partial r_1}(\frac{d\bar{r}_1}{dr_2} - 1)$$

However, $\frac{\partial U}{\partial r} < 0$, so either $\frac{\partial C_2}{\partial r_1} = 0$ or $\frac{d\bar{r}_1}{dr_2} = 1$.
The condition $\frac{d\bar{r}_1}{dr_2} = 1$ at $r_2 = r_{Nash}$ gives rise to the condition

$$\beta + \nu\frac{\partial C_1}{\partial r_1}\left\{\frac{\partial C_1}{\partial r_1} + \frac{\partial C_1}{\partial r_2}\right\} + \gamma\left\{\frac{\partial^2 C_1}{\partial r_2 \partial r_1} + \frac{\partial^2 C_1}{\partial {r_1}^2}\right\} = 0$$

Since we are free to vary the values of β and ν (as long as they obey the inequality condition), it is impossible for an allocation function to guarantee that $\frac{d\bar{r}_1}{dr_2} = 1$. Thus, we must have $\frac{\partial C_2}{\partial r_1} = 0$ (with all quantities evaluated at $r_1 = r_2 = r_{Nash}$). Therefore, the allocation function must satisfy $\frac{\partial C_i}{\partial r_j} = 0$ whenever $r_j = r_i$, $i \neq j$. This uniquely specifies the Fair Share allocation function in MAC.

□

Proof of Theorem 5:

1. This follows from Theorem 8 in [8].

2. Consider a two-user subsystem and some profile (U_1, U_2) and Nash equilibrium (r_1, r_2). Assume that it is also a Stackelberg equilibrium of the two-person subsystem and assume that $\frac{\partial C_2}{\partial r_1} \neq 0$. By using the utilities introduced in the proof of Lemma 5, we can choose utilities such that the best-reply function $\vec{\bar{r}}(r_1)$ is differentiable with $\frac{d\bar{r}_2}{dr_1} \neq 0$ around the Nash equilibrium. The necessary but not sufficient first derivative condition is

$$\frac{dU_1(r_1, C_1(\vec{\bar{r}}(r_1)))}{dr_1} = 0$$

This condition can be reexpressed as:

$$M_1(r_1, c_1) = -\frac{\partial C_1}{\partial r_1} - \frac{\partial C_1}{\partial r_2}\frac{d\bar{r}_2}{dr_1}$$

Since this is also a Nash equilibrium, we must also have

$$M_1(r_1, c_1) = -\frac{\partial C_1}{\partial r_1}$$

Combining these expressions, we have $\frac{\partial C_1}{\partial r_2}\frac{d\bar{r}_2}{dr_1} = 0$. However, since we know that this second term is nonzero, the first term must vanish. Thus, $\frac{\partial C_2}{\partial r_1} \neq 0 \Rightarrow \frac{\partial C_1}{\partial r_2} = 0$. Therefore, when $r_1 = r_2$ we must have $\frac{\partial C_2}{\partial r_1} = \frac{\partial C_1}{\partial r_2} = 0$. From Lemma 1, this implies that we must have the Fair Share allocation function.

□

Proof of Theorem 7:

1. Recall the definitions $E_i = M_i(r_i, C_i(\vec{r})) + \frac{\partial C_i}{\partial r_i}$ and $A_{i,j} = \delta(i,j) - \frac{\partial E_i}{\partial r_j}\left(\frac{\partial E_j}{\partial r_j}\right)^{-1}$. Since the matrices $\frac{\partial C^{FS}{}_i}{\partial r_j}$ and $\frac{\partial^2 C^{FS}{}_i}{\partial {r_j}^2}$ are lower triangular (when we arrange the r_i's in increasing order), $A_{i,j}$ is also lower triangular with zeros on the diagonal. This is clearly nilpotent. As before, the proof carries over to all subsystems.

2. Clearly $A_{i,i} = 0$ in general. The condition of nilpotence requires that $A^N = 0$. Consider, for $M \geq N$,

$$0 = [A^M]_{i,j} = \sum_{s_1=1 s_2=1 \ \dots \ s_{M-1}=1}^{N} A_{i,s_1} A_{s_1,s_2} A_{s_2,s_3} \dots A_{s_{M-1},j}$$

This sum can be partitioned into a sum over combinations (C's) of M integers in $[1, N]$ and a sum over permutations (P's) of that combination.

$$0 = [A^M]_{i,j} = \sum_{C's} \sum_{P's} A_{i,s_1} A_{s_1,s_2} A_{s_2,s_3} \dots A_{s_{M-1},i}$$

By choosing the U_i's appropriately, we can freely adjust the terms in E_i depending on derivatives of M_i. Thus, each sum over permutations of a particular combination must separately vanish. Assume that the matrix $A_{i,j}$ has a cycle. Let K_{min} be the length of the shortest cycle and assume, without loss of generality, that this cycle is made up of elements $1, 2, ..., K_{min}$. Since $A_{i,i} = 0$, $K_{min} > 1$. There can be no other cycle of length K_{min} with the same combination of integers, since one could then combine these two cycles to form a shorter cycle. Consider the term $[A^{NK_{min}}]_{1,1}$. Each term in the sum over permutations of the combination of elements of N copies of $[1, .., K_{min}]$ consists of (1) an N-fold product of the original cycle (which does not vanish), (2) an ordering of indices that contains a shorter cycle (and therefore vanishes), or (3) a permutation of our shortest cycle (which must also vanish). However, the nilpotent condition requires that the total sum vanish. Thus, by contradiction, there can be no cycles. The vanishing of a term $A_{i,j}$ for $i \neq j$ for all M_i requires both $\frac{\partial C_i}{\partial r_j} = 0$ and $\frac{\partial^2 C_i}{\partial r_i \partial r_j} = 0$. Thus, acyclicity of the dynamical stability matrix is equivalent to acyclicity of the dependency matrix. The only MAC allocation function having an acyclic dependency matrix is the Fair Share allocation function (from Lemma 3).

□

Proof of Theorem 8:

1. The Fair Share allocation $\vec{C}^{FS}(\vec{r})$ satisfies the equation

$$F((r_1\ ,\ r_2\ ,\ r_3\ ,\ ...r_k - 1\ ,\ r_k\ ,\ r_k\ ,\ r_k...)\ ,\ (C_1\ ,\ C_2\ ,\ C_3\ ,\ ...C_k - 1\ ,\ C_k\ ,\ C_k\ ,\ C_k...)) = 0$$

for all k, when the users are labeled in increasing order of r_i's. Clearly $\frac{\partial C_i}{\partial r_j} = 0$ whenever $r_j \geq r_i$, and $\frac{\partial C_i}{\partial r_j} > 0$ otherwise. Therefore, ${C_i}^{FS}(\vec{r}) \leq {C_i}^{FS}(\vec{e} r_i)$, proving protectiveness.

2. Consider a protective MAC allocation function $\vec{C}$ and a subsystem with only two users. When $r_1 = r_2$ symmetry requires that $C_1 = C_2$ and this value is determined by the constraint function. If we continuously increase r_2 holding r_1 constant, then C_1 must also remain constant. This is because the monotonicity conditions in MAC require that C_1 not decrease when increasing r_2 and the protectiveness condition of the hypothesis requires that C_1 never be more than its value when all users in the subsystem request the same amount. Since this applies to every two user subsystem of larger systems, we must have $\frac{\partial C_i}{\partial r_j} = 0$ whenever $r_i \leq r_j$. This condition, as shown in Lemma 1, uniquely specifies the Fair Share allocation function in MAC.

□