Resource Allocation and Networking Games Lecture 6: Game Theoretic Approach to Scheduling Algorithm

EE E6970, Prof.Aurel A. Lazar
note taken by Raymond R.-F. Liao
liao@ctr.columbia.edu

April 1997

1 Introduction

This course note covers three consecutive classes delivered in March 1997 on using game theory to analyze the scheduling algorithms. The course material mainly follows the results from Scott Shenker in [1].

Performance problems arising from the evolving high-speed, wide-area networks such as Internet and ATM have been calling for distributed and robust control algorithms. Quite often the problems can be more realistically represented in a *noncooperative* paradigm due to various network delay and selfish user behavior. Game theory offers a refreshing perspective to understand these noncooperative behaviors and to systematic design effective and robust network control algorithms for noncooperative (i.e. most realistic) networks.

Game theory has been applied in all the aspects of network control areas, including flow control, routing, pricing and scheduling. However, most of the work are in the first three areas due to tractability. For scheduling, the focus shifts from users' utility function to the allocation rule which is the switch scheduling algorithm. Lack of formulation on the scheduling algorithms and of the underlying game structure makes the problem more difficult to solve. So far, few solid results have been reported except [1] which has to heavily rely on postulation.

Nevertheless, the design of scheduling algorithms for noncooperative users is a very useful issue because switch scheduling algorithm is the most effective control method that could be designed to improve network performance at a short time scale.

In the following, we present the notation and problem formulation in Section 2, introduce a specific scheduling algorithm in Section 3, present its good features in Section 4 and summarize in Section 5.

2 Notation and Game Formulation

The system considered here is a single switch shared by N users. Each user sends packets to the switch at a Poisson rate. The rate at which user i sends packets to the switch is r^i. The user controls this rate by using a flow control algorithm. The switch is serviced by an exponential server. The congestion experienced by a user, c^i, is measured by the average number of packets from user i waiting in the server's queue. The congestion is dependent on both the rates at which the various users send packets to the server and on the service discipline. The server has control on the buffer dimensioning and scheduling algorithm.

The notations used in the lecture are different from the ones in the reference paper [1] in some cases. This note follows mainly with the notations used in the paper (which are also used in the routing part of the lecture) with a few modifications shown below to be consistent with the lecture.

- the rate strategy profile of user i given the others are fixed is represented by (r^i, r^{-i}) in this note but $(\vec{r} \,|^i\, r_i)$ in the reference [1].
- the Nash equilibrium point in the strategy space is represented by r^* here but $\vec{r}^{Nash}$ in the reference, and the inequality condition for Nash equilibrium becomes : $U_i({r^i}^*, C^i(r^*)) \leq U_i(r^i, C^i(r^i, {r^{-i}}^*))$ instead of the $U_i({r_i}^{Nash}, C^i(\vec{r}^{Nash})) \leq U_i(\hat{r}_i, C^i(\vec{r}^{Nash} \,|^i\, \hat{r}_i))$ in the reference.

The game-theoretic analysis of this system is based on four principles.

1. User i's satisfaction, its utility function $U_i(r^i, c^i)$, models the amount and quality of service provided by the switch, where r^i is the Poisson arrival rate and c^i is the average queue length experienced by user i. This function relates the user's level of satisfaction with a given service level. In particular, the utility function allows us to distinguish between the different levels of satisfaction by comparing a user's utility under different allocations, i.e. comparing $U_i(r^i, C^i(r^i, r^{-i}))$ with $U_i(\hat{r}^i, C^i(\hat{r}^i, r^{-i}))$. In a noncooperative system, the user's utility function is private, and the user is only aware of his own r^i and c^i.

2. Users are selfish. Each user tries to maximize his utility by adjusting the rate at which he sends packets to the switch. The stable operating points of the system are the Nash equilibria.

   The noncooperative user assumption may initially appear to be a regrettable reality. However, assuming user to be selfish actually allows user to adopt simple hill-climbing optimization method without abstract knowledge of their preference; enables network to satisfy a wide variety of service requirements by placing the onus on user to optimize their own satisfaction; and avoids requiring any universal flow control algorithm which could impede the use of new network technologies. Therefore, the game-theoretical approach may in fact be the best way to insure good performance in the large, heterogeneous, and rapidly changing networks of the future.

3. The switch algorithm is under a centralized control. While users are independent entities, the switch is a shared resource. The focus of this work is on the switch service disciplines, which in economic term, is to control the allocation of cost (congestion).

4. The performance of the switch is measured by level of total user satisfaction achieved, which is the optimization criterion of the scheduling algorithm. Other criteria include *power*, utilization, or delay but they neglect the differences among users' preferences.

   The meaning of good performance of a scheduling algorithm in a noncooperative system has three main factors: the equilibria need to be efficient and fair; need to be unique and easy to obtain (i.e. quick to converge) ; and need to be robust, that is when not in a state of equilibrium, the system must perform at some minimal level of satisfaction.

Next we present the mathematical model for the design of scheduling algorithm under noncooperative users.

3 Mathematical Model

For tractability reason, the queueing model used here is M/M/1 . Each of the N users adds packets to the system at a rate $r^i > 0$. The average number of user i packets enqueued, c^i, is dependent on the service discipline used. In addition, the switch is shared by all the users.

The pair of quantities (r, c) shows how the switch is allocated among the users. This scheduler must satisfy the work-conserving condition. Because all the work-conserving schedulers have the same busy time distribution, the aggregated arrival processes all have the same average waiting time, no matter whether the individual flows have priority above one another.

This condition can be represented as $\sum_{i=1}^{N} c^i = f(r)$ where $f(r)$ is the queue length of the FIFO M/M/1: $f(r) = g(\sum_{i=1}^{N} r^i)$ and $g(x) = \frac{x}{1-x}$.

Furthermore, for any subset of users, because the best performance they can achieve is the performance without the interference from the other users, which equals to the FIFO M/M/1 queue length, therefore the aggregated queue length should be lower-bounded by the FIFO M/M/1 queue length for the same subset. $\sum_{i \in I} c^i \leq g(\sum_{i \in I} r^i)$, $I \subset \{1, 2, ..., N\}$.

Let AC denotes the set of acceptable allocation functions that satisfy the above conditions and the conditions that $C(r)$ is symmetric in r and continuous on first derivatives.

The allocation function for FIFO is simply the proportional allocation given by

$C^{i^P}(r) = \frac{r^i}{1-\sum_{j=1}^{N} r^j}$

Next we introduce the Fair Share scheduling which is constructed from the Fair Queueing intuition that the user should always maintain a fair share of the service, independent of the other users' actions. However, the definition of Fair Share is more synthetic towards the warrant of good game theoretical properties like the existence and convergence of Nash equilibrium.

The operation of Fair Share scheduling can be better viewed as a preemptive priority queueing system shown in Figure 1.

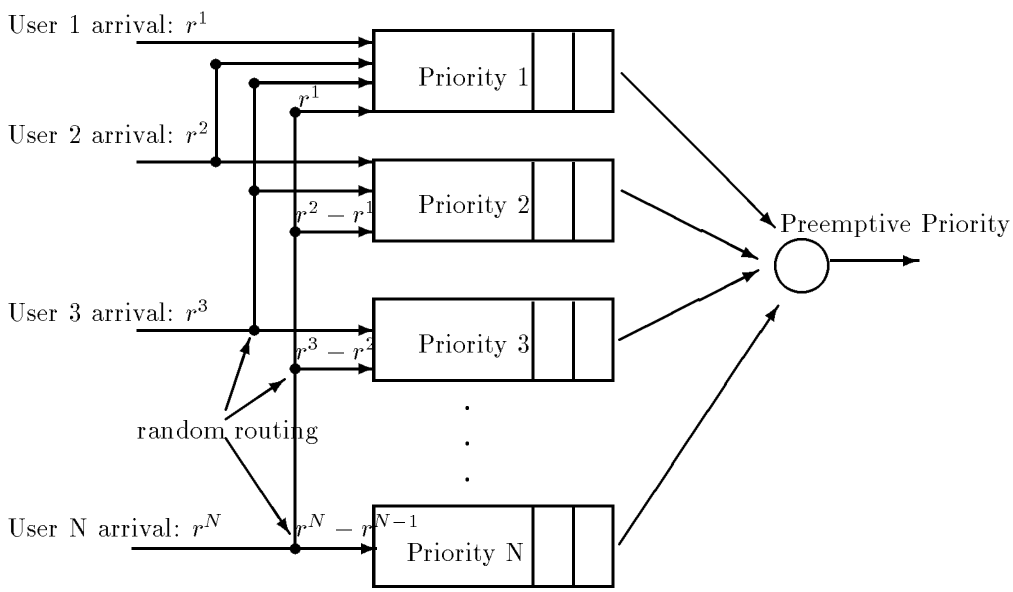

Figure 1: Illustration of Fair-Share Scheduling

All of user one's packets are in the highest priority queue, together with the same rate r^1 of packets from all the other users. Similarly, the rest users all have rate of $r^2 - r^1$ packets in the second highest queue. The procedure repeats until all the portions of all the rates are assigned with priorities.

The formal definition of Fair Share scheduling is : with the users labeled so that their r^i's are in increasing order, the kth user's allocation is defined as

$$C^{1^{FS}}(r) = \frac{g(nr^1)}{n}$$

and

$$C^{k^{FS}}(r) = \quad C^{k-1^{FS}}(r) + [g((n-k+1)r^k + r^{k-1} + \cdots + r^1) \\ -g((n-k+2)r^{k-1} + r^{k-2} + \cdots + r^1)]/(n-k+1)$$

The above definition is deduced from the M/M/1 preemptive priority queue shown in Figure 1. $g((n-k+2)r^{k-1} + r^{k-2} + \cdots + r^1)$ is the total queue length for the first $(k-1)$ priority queues, which is actually derived from the single priority M/M/1 queue with the same aggregated traffic intensity by using the work-conserving argument we discussed previously. Similarly, $g((n-k+1)r^k + r^{k-1} + \cdots + r^1)$ is the total queue length for the first k priority queues. Therefore, the second part of the right-hand-side of the equation is the mean queue size for each of the $(n-k+1)$ users sharing the kth priority queue. With the recursive definition, the above equation holds.

Since for $i \neq j$, we have $(\frac{\partial C^{i^{FS}}}{\partial r^j} > 0) \Longleftrightarrow (r^i < r^j)$, namely $C^{i^{FS}}$ depends only on those r^j which are less than r^i. Small variations in r^j will affect $C^{i^{FS}}$ if and only if $r^j \leq r^i$. This partial insularity is crucial for the good properties the Fair Queueing obtained. It allows the noncooperative users to compete for the resource in a limited way so that they can converge to equilibrium.

Intuitively, during the evolution of the Fair Share system toward equilibrium , the highest priority queue first reaches equilibrium, hence user one first reaches equilibrium, then followed by the second highest priority queue, and user two, etc., until the last priority queue and the user n reaches equilibrium.

The service disciplines in AC have various good properties. We would like to focus on the service disciplines in a subset of AC which obey certain monotonicity conditions so that the achievable properties lead to a unique scheduling algorithm, the Fair Share algorithm. In fact, the definition of this restricted set, called MAC, strongly postulates on the partial-insularity structure of the Fair Share algorithm.

Definition 1 *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. $\{\frac{\partial c^i}{\partial r^j} = 0 \text{ at } r^o\} \Longrightarrow \{\frac{\partial c^i}{\partial r^j} = 0 \; \forall r \text{ with } r^j \geq r^{j^o}, j \neq i, \text{and } r^i \leq r^{i^o}\}$

The first two conditions assume that no user benefits when other users consume more throughput. The third condition exactly specifies the partial-insularity. It states that if r^j has no impact on r^i at point r^o, then the impact is kept at 0 as r^j increases or r^i decreases.

Even though the set of MAC encompasses many scheduling algorithms like FIFO, LIFO, processor sharing, polling and HOL priority, these algorithms don't have the partial-insularity property. In fact, the condition (3) is void to them because for these algorithms, $\frac{\partial c^i}{\partial r^j} \neq 0$.

It is not surprising to see that the Fair Share allocation is the only allocation function in MAC such that $\frac{\partial C^i}{\partial r^j} = 0$ whenever $r^j = r^i, i \neq j$, which means when the rates of two users equal to each other, their mutual interference stops. Furthermore, from MAC property (3), for any allocation function in MAC, $\{\frac{\partial C^i}{\partial r^j} = 0 \text{ for all } i \neq j\} \Longrightarrow \{r^j = r^i \text{ for all i,j}\}$. Additionally, 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.

For the benefit of the existence of Nash equilibrium, the user's utility function $U_i(r^i, c^i)$ is assumed to be strictly monotonic in both variables, increasing in r^i and decreasing in c^i and is a function of only that user's service allocation. Furthermore, U_i is required to be concave[1], and continuous in the second derivatives. Denote by AU the set of acceptable utility functions that satisfy the above conditions.

The utility function is a representation of the user's preference orderings of the various allocations (r^i, c^i). Each user i independently maximizes his individual utility function by adjusting r^i while holding the other r^j constant. If an equilibrium is reached in such a noncooperative environment, then it is a Nash Equilibrium.

Definition 2 *A point r^* is a Nash equilibrium point if*
$U_i({r^i}^*, C^i(r^*)) \geq U_i(\hat{r}^i, C^i(\hat{r}^i, {r^{-i}}^*)$ *for all $\hat{r}_i$ and i.*

In a set of acceptable utility functions, under the fair share algorithm, every point that satisfies the Nash First Derivative Condition (FDC), $\frac{dU_i}{dr^i} = 0$ or $M_i(r^i, c^i) = \frac{\partial C^i}{\partial r^i}$, $M_i(r^i, c^i) \equiv \frac{\partial U_i}{\partial r^i} / \frac{\partial U_i}{\partial c^i}$ for all i, is a Nash equilibrium.

The Nash equilibrium is an optimal operating point of a noncooperative system.

It is not always possible for the users to optimize their performance. In such a situation an induced allocation function is used and must have the same properties as $c(r)$ but with some of the variables held constant.

If users are greedy, i.e. able to self-optimize their own performance, then the equilibrium operating point is the Nash equilibrium. To achieve good performance the Nash equilibria must have certain required characteristics. The standard criterion for efficiency, user satisfaction, is that of the Pareto optimality.

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

[1]Note: Typo, reference[1] assumes U_i to be convex, which is wrong because otherwise the Nash derivative condition $\frac{dU_i}{dr^i} = 0$ is not sufficient to guarantee that Fair Share allocation reaches Nash equilibrium.

A point (r, c) is Pareto optimal if and only if there exists a vector $\vec{W}$, where $W_i > 0$ [2] 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 $(\bar{r}, \bar{c})$. As a result, each Pareto allocation maximizes at least one weighted sum of the utilities.

The necessary condition that $\sum_{i=1}^{N} W_i U_i(r^i, c^i)$ has optimum with condition $F(r, c) = 0$ leads to the following equation with the help of Langrange multiplier:

$$\frac{\partial U_i}{\partial r^i} / \frac{\partial F}{\partial r^i} = \frac{\partial U_i}{\partial c^i} / \frac{\partial F}{\partial c^i} \Longrightarrow \frac{\partial U_i}{\partial r^i} / \frac{\partial U_i}{\partial c^i} = \frac{\partial F}{\partial r^i} / \frac{\partial F}{\partial c^i} \Longrightarrow M_i(r^i, c^i) = Z_i(r^i, c^i)$$

with the definition $M_i(r^i, c^i) \equiv \frac{\partial F}{\partial r^i} / \frac{\partial F}{\partial c^i} = -\frac{\partial f}{\partial r^i} = -(1 - \sum_{j=1}^{N} r^j)^{-2}$.

Unfortunately, an allocation function can not be chosen such that the Pareto optimal and the Nash equilibrium points are the same point for every set of utility functions. This leads to the next section on the performance of the scheduling algorithms.

4 Performance

4.1 Efficiency

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

Brief proof: This theorem is proved by contradiction. Assume, to the contrary, that $C(r)$ is an allocation function in MAC such that every Nash equilibrium is also Pareto optimal. For any point $r \in D$ we can find a vector of utility functions $\vec{U} \in AU^N$ that has this point as a Nash equilibrium (Lemma 5 in [1]). Combining the Pareto and Nash FDC conditions, the allocation function must satisfy the relation $\frac{\partial f}{\partial r^i} = \frac{\partial C^i}{\partial r^i}$ throughout D. We shall show that this condition is too strong to be satisfied by the allocation function for all the constraints. 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 functions satisfy the constraint yields the relation $(N-1)f(r) = \sum_{i=1}^{N} h_i(r)$. This is clearly not satisfied by our constraint function.

□

Furthermore, the combined Pareto and Nash FDC conditions are too strong to be met even when the users are allowed to signal the scheduler their preferences.

Corollary 1 *Consider allocation functions $C(r, \alpha)$ that are continuous upto the second derivative, where α is a vector of constants representing user specified*

[2] Note: Typo, reference[1] uses $W_i \geq 0$ which is not right for the sufficient condition because $W_i = 0$ doesn't lead to the inequalities for user i.

signalling parameters. There is no allocation function in MAC such that every Nash equilibrium is Pareto optimal.

However, if we relax the work-conserving condition on the scheduler, it is possible to find allocation functions whose Nash equilibria are all Pareto optimal.

Theorem 1 states that no allocation can guarantee that for **all** utility functions is AU, every Nash equilibrium is Pareto optimal. However, the next theorem shows that for a given set of utility functions, it is possible to have a service discipline whose Nash equilibria are Pareto optimal, and if all the users have the same utility function, then the Nash equilibria of the Fair Share allocation technique are always Pareto optimal.

Theorem 2 *Consider an allocation function in MAC, and a vector of utility functions in the set of acceptable utility functions.*

1. If the Nash equilibrium is Pareto optimal, the $r^i = r^j$ for all i and j.

2. Any completely symmetric r that gives rise to a Pareto optimal allocation is also a Nash equilibrium of the Fair Share allocation function.

Brief proof: 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 $\sum_{i=1}^{N} C^i = f(r)$. Taking the derivative of this 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 we have $r^i = r^j$ for all i, j.

Prove the second claim: At such a symmetric point, the delay values are completely determined by the constraint. Consequently, 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 to guarantee that this point is a Nash equilibrium. Thus, any completely symmetric r that gives rise to a Pareto optimal allocation is also a Nash equilibrium of the Fair Share allocation function.

□

The Fair Share allocation function achieves all points where Nash equilibria are Pareto optimal, as opposed to the proportional allocation functions that always have $\frac{\partial C^i}{\partial r^j} > 0$ and never have Pareto optimal Nash equilibria.

4.2 Fairness

In addition to efficiency, it is preferable to have fairness property. An allocation is deemed fair if it is envy-free. Under envy-free environment, user i will not be envious of user j if $U_i(r^i, c^i) < U_i(r^j, c^j)$. Envy doesn't involve comparison of two users' utility functions (in fact different users' utility functions are not comparable). It only involves the comparison of two user's allocations under the preference ordering of one of the users. The condition defined next states that no matter what else does, if a user maximizes her own utility by choosing r^i, she will envy no one. The allocation functions satisfy this unilaterally envy-free condition have envy-free Nash equilibria.

Definition 4 *An allocation function is unilaterally envy-free if*
$\{U_i(r^i, C^i(r)) \geq U_i(\hat{r}^i, C^i(\hat{r}^i, r^{-i}))$ *for all* $\hat{r}^i\} \Longrightarrow$
$\{U_i(r^i, C^i(r)) \geq U_i(r^j, C^j(r))$ *for all* $j\}$

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.

Proof on the first claim: Assume that user one has the utility function U in AU. User one will envy user two if the function $E(r^1, r^2) = U(r^2, C^{2^{FS}}) - U(r^1, C^{1^{FS}})$ is positive. For a given r^2, let r^1 assume the value that maximizes $U(r^1, C^{1^{FS}})$; call this value r^{um}, the unilateral maximum. At this point we have the FDC $\frac{d}{dr^1}U(r^1, C^{1^{FS}}) = 0$. 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 $E(r^{um}, r^2) \leq 0$. We can maximize this envy by fixing $r^1 = r^{um}$ and varying r^2 between r^{um} and 1. Varying r^2 does no affect $C^{1^{FS}}$ under the Fair Share allocation function, so that maximizing the envy is equivalent to maximizing the function $U(r^2, C^{2^{FS}})$ But, from the definition of r^{um}, we know that this function satisfies the FDC $\frac{d}{dr^2}U(r^2, C^2FS) = 0$ at $r^2 = r^{um}$ and also that $\frac{d^2}{dr^{2^2}}U(r^2, C^{2^{FS}}) \leq 0$ 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.

□

4.3 Convergence

There are a variety of desired characteristics required during the convergence of the system to an equilibrium such as the uniqueness of the equilibria, the rate of convergence, and the viability of the technique. Without uniqueness there is ambiguity, therefore there must be only one Nash equilibrium point. The Fair Share allocation function is the only *MAC* service discipline that ensures there will be one and only one Nash equilibrium.

Theorem 4 *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.

Another requirement for convergence is robustness. One technique used to find an equilibrium point is the hill-climbing technique. In an environment where all the users are performing optimization at the same time the hill climbing algorithm dynamics can be very complex. The sophisticated user becomes a leader, with the other less sophisticated users following. The simple hill climbing users can be exploited if a sophisticated user has information about other users' utility functions, and therefore knows how their flow control could react. Consequently, the hill climbing technique needs to be invulnerable to sophisticated strategies.

Definition 5 *Let user one be the leader, and let $\bar{r}(r^1)$ be a function such that*

1. *the first component of the vector is given by the argument, and*
2. *for all $i > 1$, $U_i(\bar{r}^i, C^i(\bar{r})) \geq U_i(\hat{r}^i, C^i(\hat{r}^i, \bar{r}^{-i}))$ for all $\hat{r}^i$.*

We then consider it a Stackelberg equilibrium with user one leading if $U_1(r^1, C^1(\bar{r}(r^1)) \geq U_1(\hat{r}^1, C^1(\bar{r}(\hat{r}^1)))$ for all $\hat{r}^1$.

The leader's utility in Stackelberg equilibrium is never less than its corresponding Nash equilibrium. As a result, one should implement an allocation function that gives rise to Nash equilibria that are also Stackelberg equilibria.

Moreover, the convergence to the Nash equilibrium should be robust enough so that the convergence is assured as long a user implements any reasonable form of self-optimization. We define a reasonable learning algorithm as an algorithm where users eliminate all the values of r^i which will produce worse performance than other values of r^i. The convergence will be robust if any combination of generalized hill climbing optimization algorithms converge to the unique Nash equilibrium within a finite amount of time.

Theorem 5 *1. 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.

If the users report their utility functions directly to the switch, we need an allocation mechanism that is a function of the reported utility functions. This will improve the efficiency of the protocol even more. There is a function B that maps the set of reported utility functions into a set of (r,c) allocations: $(r^i, c^i) = B^i(\hat{U})$. This function is called a revelation function. It encourages users to tell the truth about their utility functions and not to exploit their followers, which is what we mean in the Vickery auction game, the incentive compatibility.

4.4 Robustness

It's quite possible that the usual operating points of the system are out-of-equilibrium. Therefore the robustness of the allocation functions is important. The unilaterally envy-free condition guarantees that a self-optimizing user will envy no one, but this is not enough to guarantee the satisfaction of the user. Here we define a service discipline *protective* if the allocated congestion is upper bounded by $C^i(r) = r^i/(1 - Nr^i)$ where all the elements in r are equal.

The protective property gives a minimum guarantee to each user. The Fair Share allocation has this property because of its partial-insularity.

Theorem 6 *1. The Fair Share allocation function is protective in all subsystems.*

2. Fair Share is the only MAC allocation function that is protective in all the subsystems.

5 Summary

In a noncooperative network a state of equilibrium can be reached with the proper choice of service disciplines. The Fair Share allocation allows us to reach a fair Nash equilibrium by any set of reasonable optimization algorithms due to its partial-insularity property. These optimization techniques converge quickly, and during the convergence process all users are protected.

Last, before conclude the topic, we compare the difference between the auction game and the scheduling/flow control problem. In auctioning, the model we considered is deterministic model, the game has no repetition; in the scheduling problem, we use stochastic model, the preemptive server can allow a later higher bidder to stay in service, furthermore the game is repetitive as a tournament. Besides this, the essential difference between "Auctioning Game" and "Scheduling Game" is the **incentive compatibility**. Auctioning uses price to drive users' incentive compatibility. While for scheduling, even though Fair Share tries to achieve this, it is still constrained by the time scale of the scheduler unless pricing is applied only at the connection level.

References

[1] Scott J. Schenker, "Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines", *IEEE/ACM Trans. on Networking*, Vol. 3, No. 6, pp819-831, Dec. 1995.

[2] Sareh Greher, Course Note of Lecture 6, EE E6940, "Game Theory and Its Application to Network Flow Control in a Noncooperative Environment", 1995.