

Exploiting Anarchy in Networks: A Game-Theoretic Approach to Combining Fairness and Throughput

Sreenivas Gollapudi
SUNY Buffalo

D. Sivakumar
IBM Almaden

Aidong Zhang
SUNY Buffalo

Abstract—We propose a novel mechanism for routing and bandwidth allocation that exploits the selfish and rational behavior of flows in a network. Our mechanism leads to allocations that simultaneously optimize throughput and fairness criteria. We analyze the performance of our mechanism in terms of the induced *Nash equilibrium*. We compare the allocations at the Nash equilibrium with throughput-optimal allocations as well as with fairness-optimal allocations. Our mechanism offers a smooth trade-off between these criteria, and allows us to produce allocations that are approximately optimal with respect to both. Our mechanism is also fairly simple and admits an efficient distributed implementation.

I. INTRODUCTION

The success and stability of a large network such as the Internet depends crucially on its ability to achieve a high throughput while treating all users in a fair manner. Fairness and throughput maximization are well-known to be conflicting desiderata. There are several methods to achieve high throughput, but they all lead to poor fairness properties. Jaffe’s classical bottleneck flow control algorithm [1] achieves what is arguably the “most fair” allocation, but leads to poor throughput and utilization of the network bandwidth. Recently, there has been a surge of interest in routing and allocation policies that simultaneously attempt to achieve fairness without compromising too much in terms of throughput [2], [3], [4], [5].

Jaffe’s allocation mechanism and the more recent work of [2], [3], [4], [5] represent two extreme approaches to route traffic in large networks. In the former, each network user (aka. “flow”) submits a path to the allocation algorithm, which then determines how much bandwidth the flow receives along the chosen path. This is a natural approach, given that most routers in the network have a fairly accurate snapshot of the network topology. The drawback of Jaffe’s algorithm is that regardless of how carefully a flow chooses its path, the global bottlenecks in the traffic will be the limiting factor in determining its allocation, *irrespective* of how much the flow can contribute to the throughput. In the latter works, routing and bandwidth apportionment are done by a single algorithm, to which each flow merely submits the source and destination information. While this leads to provable guarantees on the simultaneous achievement of approximate fairness and throughput properties, this is clearly unrealistic in large networks. Moreover, the general problem here is NP-hard, and the known approximation algorithms carry substantial overhead.

In this paper, we approach the question of routing and bandwidth allocation from a different perspective. We design routing and allocation algorithms that are cognizant of the fact that flows are *selfish* and *rational*. By selfish, we mean that flows care essentially about their own allocations, without any consideration for global throughput or fairness issues. By rational, we mean that once the routing and allocation policies are instituted, each flow make decisions (e.g., choosing paths) that do not negatively affect its own allocation (e.g., in the presence of an alternative, a flow will never choose a more congested path). In fact, as we will see, our algorithm is not only aware of selfish and rational behavior, but will crucially exploit this behavior in achieving its goals.

The study of selfish and rational agents participating in a non-cooperative game is central to game theory. A fundamental notion in this context is that of a *Nash equilibrium*, which characterizes the eventual behavior of the agents from which they have no incentive to deviate. Designing *mechanisms*, or the ground rules for the games involving selfish and rational agents, so that the resulting Nash equilibria have desirable properties is an important topic of research in algorithmic game theory.

We initiate the study of routing and allocation mechanisms that induce Nash equilibria at which the resulting allocations that satisfy fairness and throughput criteria. Thus our approach is different from that of [2], [3], [4], [5], and is more similar to Jaffe’s approach in that flows choose paths and the mechanism will determine each flow’s allocation. While Jaffe makes fairness guarantees about individual runs of the algorithm, we analyze mechanisms from the viewpoint of their performance at the Nash equilibrium. Here we consider the Nash equilibria that result when users follow their natural greedy strategy, often called the “best response dynamics” (to be described in detail in Section II).

Our main contribution in this paper is a mechanism for allocating bandwidth when each flow submits a path in a network. The Nash equilibria resulting from our mechanism (when players follow best response dynamics) can be characterized by the parameters of the mechanism; by making appropriate choices of these parameters, it is possible to produce allocations that can achieve very high throughput or excellent fairness properties, or, more importantly, a smooth interpolation between these two endpoints. In particular, we show that the mechanism admits parameter settings for which the resulting allocations achieve close to optimal throughput

and are also close to being max-min fair.

In addition to achieving high throughput while being quite fair, our mechanism has several desirable properties. The first addresses the task of routing flows. In the holistic approach of [2], [3], [4], [5], routing is explicitly performed by the algorithm that determines the allocation. In contrast, we exploit the selfish and rational behavior of flows to perform their own routing. In particular, our mechanism automatically handles the dynamic case where new flows enter the network, and determine their paths greedily to maximize their allocation. The appealing feature of our mechanism is that in spite of (actually, because of) the anarchy that results from flows choosing their own paths, we are able to achieve good global properties that selfish flows do not care about).

Another desirable feature of our mechanism is that admits a very natural distributed and asynchronous implementation with very little overhead, both in terms of message complexity and local computational complexity.

We conducted extensive experiments to study the allocations that arise from selfish and rational behavior by flows in response to our mechanism. Our experiments lead to a fine understanding the roles of various parameters of the mechanism in determining the throughput and fairness at various congestion levels. They also show that the overall performance of the mechanism is consistent with the principles underlying the mechanism. Finally, our experiments show a dramatic difference in the Nash equilibria that emerge from Jaffe’s mechanism and our mechanism. Briefly, Jaffe’s mechanism leads to a Nash equilibrium where every flow is “dissatisfied” with its allocation along any path it chooses, and consequently maintains a large set of paths from which it chooses uniformly from time to time. In contrast, our mechanism leads to a Nash equilibrium where a good fraction of the flows are able to identify a single path along which they receive a substantial allocation (one that is much higher than under Jaffe’s Nash equilibrium), while many remaining flows maintain small sets of paths from which they choose. Flows that use a single path lead to an improved throughput (over Jaffe’s Nash equilibrium), while flows that choose from a large set of paths do so to guard themselves from unfair treatment from the mechanism (as they do under Jaffe’s Nash equilibrium).

A. Related work

There is an extensive literature where game-theoretic tools and analyses have been applied to problems in network routing, flow control, and bandwidth allocation. We will not attempt a complete survey here; rather, we will point out the salient differences between existing work and ours. The reader is referred to the survey article by Altman et al. [6] for an excellent overview of the various threads of research in this area.

Several papers (e.g., [7], [8]) consider total latency of all users as their objective function. There is a fundamental difference between latency and the bandwidth allocation as objective functions: the latency experienced by a flow along a path is the sum of delays encountered on each link in the path; by

contrast, the bandwidth allocation for a flow is the minimum of the allocations it receives from each link in the path. This difference between additive and non-additive objectives is crucial in our analysis, since our mechanism, similar to Jaffe’s allocation mechanism, is based on “bottleneck allocation,” where the allocation for a flow is determined by the minimum allocation it receives from the links on its path. See Section 2.5 of [9] for some comparison between additive and non-additive models.

There is a class of papers (e.g., [10], [11], [12], [13], [14], [15]) that deal with pricing as a method for flow control. In our setting, we do not employ pricing or taxation or any other form of Stackelberg strategy (where a centralized authority determines the paths/allocations for some users and the rest follow from selfish choices).

Service provisioning for quality of service applications has also received game-theoretic analysis [16], [17], [18], taking into account the existence of multiple service levels and varying abilities of users to pay for service.

Finally, there are papers (e.g., [19]) that characterize allocation games where the user strategies correspond to reserving various values of bandwidth. In our setting, the strategy space of users corresponds to the set of paths between their sources and destinations.

In summary, the most natural point of comparison for our work is the allocation mechanism of Jaffe. To the best of our knowledge, there is no prior work in analyzing Jaffe’s mechanism from a game-theoretic perspective, especially taking into account the selfish behavior of flows in networks.

II. MODELS AND PRELIMINARIES

In this section, we describe the network model, define the routing and allocation problems, and describe the relevant game-theoretic framework we use in this paper.

The network consists of an undirected graph $G = (V, E)$, where V denotes the set of nodes, E denotes the set of links, and where each link $l \in E$ has a real capacity $c(l) > 0$. The genre of routing and allocation mechanisms proposed in this paper are especially suited to inter-domain routing among the border nodes in a large network. The inter-domain links carry the bulk of the network traffic, and are particularly susceptible to congestion, greedy/malicious traffic, and the overall network performance critically depends on the policies and protocols employed on these links. The inter-domain network tends to be sparse and well-connected. The mechanisms proposed in this paper do not assume any specific structural characteristics of the graph.

The traffic on the network consists of n flows, where each flow is a source–destination pair of the form (s, d) , where $s, d \in V$. By a *path* from s to d we mean a sequence $P = (u_0, u_1), (u_1, u_2) \dots, (u_{t-1}, u_t)$ in G , where $u_0 = s$, $u_t = d$, and for $1 \leq i < t$, $(u_i, u_{i+1}) \in E$. Note that a flow can be routed along any path from s to d in G . Where no confusion arises, we will often refer to P as a set of edges rather than an ordered sequence.

The process of routing and bandwidth allocation is continuous and iterative. Each flow (s, d) selects a path from s to d , and submits it to an allocation mechanism. Given the set of paths, the allocation mechanism computes a bandwidth allocation for each flow, subject to the link capacity constraints. The network then routes traffic from the flows along their chosen paths according to their allocations. At any point of time, a flow might choose a new path from its source to its destination, and obtain a new allocation along this path.

The basic process outlined above has been widely employed in network routing and allocation; for example, the classical bottleneck flow control algorithm of Jaffe [1] uses this structure. We next define this process formally as a non-cooperative game, together with the related game-theoretic notions.

Definition 1 (Allocation mechanism $M^{G,F,P}$): Let $G = (V, E)$ be a graph with capacity function $c : E \rightarrow \mathbf{R}$. Let F denote a set of n flows in G , and for $f \in F$, let $P(f) \subseteq E$ denote a path for flow f in G . An *allocation mechanism* M takes G, F and P as inputs, and determines an allocation $M^{G,F,P} : F \rightarrow \mathbf{R}$ such that for every $l \in E$, the capacity constraint $\sum_{f \in F: l \in P(f)} M^{G,F,P}(f) \leq c(l)$ is satisfied.

The allocation–routing game. Fix the graph $G = (V, E)$ that corresponds to the communication network. Let F denote a set of n flows that require bandwidth. For $f \in F$, let $s_f \in V$ denote the source vertex for flow f and $d_f \in V$ denote the destination vertex for flow f . An allocation mechanism M naturally defines a non-cooperative game where:

- each flow in F is a participant;
- the set of strategies available to flow f is the set of paths in G from s_f to d_f ;
- the payoff to participant $f \in F$ is the allocation determined by M when each flow chooses a strategy, namely a path from its source its destination.

The basic premise of our work is that flows in a network are *selfish*; that is, each flow would like to maximize the bandwidth allocated to it. The standard game-theoretic model of selfish behavior is described as follows:

- each flow $f = (s_f, d_f)$ employs a probability distribution μ_f over the set of all s_f – d_f paths in G , from which it chooses a path P .
- each flow knows the probability distribution employed by all the other flows in the network, and the mechanism that will determine the allocations. Thus, the distributions μ_f are chosen strategically in a full-information setting so as to maximize the expected allocation of flow f .

Nash [20] showed that every game of this form is guaranteed to have an “equilibrium” where each participant employs a “mixture of strategies,” that is, a probability distribution over “pure strategies,” which, in our case, are the paths. We define this formally next.

Notation. Let μ be a probability distribution on a set Z ; by $X \in_{\mu} Z$, we mean that X is a random variable chosen from Z according to the distribution μ , that is, for $z \in Z$, $\Pr[X = z] = \mu(z)$. Let $\mathbb{E}[Z]$ denote the expectation of the random variable Z .

Definition 2 (Expected allocation under μ with respect to M): Let G be a graph and let F be a set of flows in G . For $f = (s, d) \in F$, let $\mathcal{P}(f)$ denote a collection of s – d paths in G , and let μ_f denote a probability distribution on the paths in $\mathcal{P}(f)$. Let M be an allocation mechanism. The *expected allocation of M under μ* is a mapping $M^{G,F,\mu} : F \rightarrow \mathbf{R}$, defined by $M^{G,F,\mu}(f) = \mathbb{E}[M^{G,F,P}(f)]$, where the expectation is taken over choices of paths $P_g \in_{\mu_g} \mathcal{P}(g)$, for each flow $g \in F$.

Definition 3 (Nash equilibrium with respect to M): Let G be a graph and let F be a set of flows in G . For $f = (s, d) \in F$, let $\mathcal{P}(f)$ denote a collection of s – d paths in G , and let μ_f denote a probability distribution on the paths in $\mathcal{P}(f)$. Let M be an allocation mechanism. The collection of distributions $\{\mu_f \mid f \in F\}$ are said to be in *Nash equilibrium with respect to M* if for all $f \in F$ and all distributions ν_f on $\mathcal{P}(f)$, $M^{G,F,\mu}(f) \geq \mathbb{E}[M^{G,F,P}(f)]$, where the expectation in the RHS is taken when $P_f \in_{\nu_f} \mathcal{P}(f)$ and for $g \neq f$, $P_g \in_{\mu_g} \mathcal{P}(g)$.

In other words, a collection of distributions $\{\mu_g \mid g \in F\}$ is said to be at equilibrium if no flow $f \in F$ has any incentive to unilaterally deviate from μ_f and use a different distribution ν_f when the other flows $g \neq f$ adopt μ_g . Equivalently, if a flow $f \in F$ unilaterally deviates from μ_f while the flows $g \neq f$ follow μ_g , its expected allocation cannot increase.

A collection of distributions $\{\mu_f \mid f \in F\}$ is said to be a *pure strategy Nash equilibrium* if for each $f \in F$ there is exactly one path $P_f \in \mathcal{P}(f)$ such that $\mu_f(P_f) = 1$, that is, each flow employs a strategy set consisting of exactly one path. Note that, in general, pure strategy Nash equilibria are not guaranteed to exist for all games. Note also that the game may have multiple Nash equilibria. In fact, it is an active area of research to characterize games that are guaranteed to contain pure strategy equilibria and/or a unique equilibrium. For example, a classical result of Rosenthal [7] asserts that every game in a class of games called “congestion games” has a pure strategy Nash equilibrium. Another class of games with pure strategy Nash equilibrium is the so-called “potential games” of Monderer and Shapley MS96. See the articles by Altman et al. [6] and Lazar, Orda, and Pendarakis [19] for additional pointers.

A. Fairness and Throughput

Our goal is to design an allocation mechanism M such that resulting Nash equilibria have desirable properties from the viewpoint of network performance. Namely, the resulting allocations should be equitable to the participants, result in high throughput, and maximize utilization of the network capacity. Below we precisely define the measures of performance that we are interested in. The two conflicting measures of primary importance in networks are throughput and fairness. From a game-theoretic viewpoint, throughput is a measure of total welfare (traffic supported) of the participants (flows) in the network, while fairness relates to the welfare (allocation) of individual participants. While throughput has a fairly straightforward quantitative definition, defining fairness

is more subtle. The most widely adopted notion of fairness is max-min fairness.

Definition 4 (Throughput): For a set F of flows in a graph G , the throughput of an allocation $a : F \rightarrow \mathbf{R}$ is defined as $\sum_{f \in F} a(f)$.

Definition 5 (Max-min fair allocation): Let F be a set of flows on a graph G . An allocation $a : F \rightarrow \mathbf{R}$ is said to be *max-min fair* if, for any allocation $b : F \rightarrow \mathbf{R}$ and $f \in F$, $b(f) > a(f)$ implies that there exists $g \neq f$ such that $b(g) < a(g)$ and $a(g) \leq a(f)$.

Jaffe’s classical algorithm [1] results in an allocation that is max-min fair. For future reference, we outline this allocation algorithm.

Mechanism $J(G = ((V, E), c), F, P)$

- (1) For $l \in E$, let $F_l \leftarrow \{f \mid l \in P(f)\}$.
- (2) While $F \neq \emptyset$ do:
 - (2.1) For $l \in E$ let $share(l) \leftarrow c(l)/|F_l|$.
 - (2.2) Let $l^* = \arg \min_l \{share(l)\}$.
 - (2.3) For $f \in F_{l^*}$:
 - $alloc(f) \leftarrow share(l^*);$
 - for $k \in P(f)$, $c(k) \leftarrow c(k) - share(l^*);$
 - $F \leftarrow F - \{f\}$
 - (2.4) $E \leftarrow E - \{l\}$.
- (3) Return $alloc$.

B. Comparing allocations

As outlined in the introduction, the primary goal of our work is to understand the quality of mechanisms by studying their induced Nash equilibria. Namely, let M and N be two allocation mechanisms, and let μ and ν , respectively, denote the probability distributions that arise in the Nash equilibria with respect to M and N . We will compare the expected allocations $M^{G,F,\mu}$ and $N^{G,F,\nu}$ from the viewpoint of throughput and fairness. For the latter, we present three ways to compare two allocation functions for the same set of flows, specifically tailored to study fairness properties of an allocation mechanism. We also define the ratio of throughputs as a measure to compare allocations from the viewpoint of throughput maximization. These will be used later when we compare the Nash equilibrium induced by our mechanism with the equilibrium induced by Jaffe’s max-min fair allocation.

Given two allocations a and b , the first definition compares their “bottomline” allocations, that is, the smallest allocations made by a and b . The second definition is a measure of how competitive a is with respect to the smallest allocation made by b . While the first two measures compare the allocations made by a with respect to the smallest allocation made by b , the third definition compares the allocations made by a and b to individual flows, and provides an aggregate.

Definition 6 (Indices for comparing allocations): Let F be a set of flows on a graph G . Let $a : F \rightarrow \mathbf{R}$ and $b : F \rightarrow \mathbf{R}$ denote two allocation functions for the set F of flows.

(1) The *minimum allocation index* $\omega_{a,b}$ of a with respect to b is defined by $\omega_{a,b} = (\min_{f \in F} a(f)) / (\min_{f \in F} b(f))$.

(2) The *competitive index* $\kappa_{a,b}$ of a with respect to b is defined by $\kappa_{a,b} = 1 - (|\{f \mid a(f) < b_{\min}\}|) / (|F|)$, where $b_{\min} = \min_{f \in F} b(f)$.

(3) For $0 \leq c \leq 1$, let $\gamma_c^{a,b} = (|\{f \mid a(f) \geq cb(f)\}|) / (|F|)$. The *global approximation index* $\gamma_{a,b}$ of a with respect to b is defined by $\gamma_{a,b} = \int_0^1 c \gamma_c^{a,b} dc$.

(4) The *throughput index* $\tau(a,b)$ of a with respect to b is defined by $\tau(a,b) = (\sum_{f \in F} a(f)) / (\sum_{f \in F} b(f))$.

C. Critique of our model

In this section, we briefly outline some of the features and limitations of our work.

Unlike much of the literature in the game-theoretic analysis of flow control, we do not consider the exact form of the users’ utility functions. The only assumption we make is that the utility functions, namely the benefit that each user derives for various values of bandwidth allocation, is non-decreasing.

Secondly, our mechanism does not require the users to report any private information, e.g., utility functions. As a consequence, there is no concern about truthfulness of users, namely users mis-reporting their utilities to gain unfair advantage. The only information supplied by a user to the mechanism is a path from their source to their destination.

In analyzing Nash equilibria of games with respect to desirable properties such as fairness and throughput, an important question is: which Nash equilibrium does one study? There are several possible answers to this question. If the game has a unique Nash equilibrium, obviously this question is irrelevant. Unfortunately, for many games of interest (such as ours) it is either not known whether there is a unique equilibrium, or worse yet, it is known that multiple Nash equilibria exist. A pessimistic approach is to study the properties of the worst-case equilibrium; this has been called the “price of anarchy” [21], [8]. An optimistic approach is to study the properties of the best-case equilibrium; this has been called the “price of stability” [22], [23]. There are two reasons why this is not unrealistic to study: the best-case Nash equilibrium, if it can be computed, serves as a proposal that a centralized agent can enforce, with the guarantee that no user has any incentive to deviate from it; secondly, in many interesting games, the equilibrium attained when users follow a natural and greedy strategy can be shown to be a stable equilibrium. We follow this approach. In our setting, we study the Nash equilibrium that emerges in the following iterative process. After flows submit their paths to the mechanism and an allocation is determined, each flow revises its mixed strategy based on the expected allocation (from previous iterations) along every link in the network, and the process repeats. This is called the “best response dynamics” in game theory literature. An important question is whether this process converges to an equilibrium. Our experiments show that this process converges, in fact, fairly quickly, to an equilibrium. If it can be shown that this process converges always to a Nash equilibrium, it would be a very interesting result, since computing Nash equilibrium in general is a hard computational problem [24]. We leave this as an interesting open question.

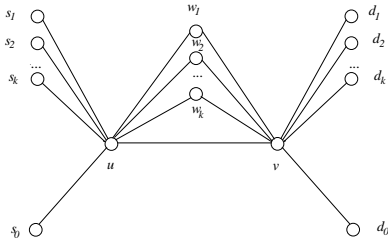


Fig. 1. The “spider” topology

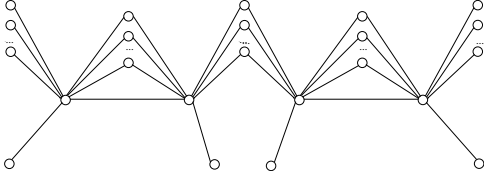


Fig. 2. A series of spiders

III. SELFISH ROUTING AND QOS

The fundamental premise of our work is that flows in a network are *selfish and rational*, that is, agents try to maximize their bandwidth allocation by rationally picking suitable paths in the network. With the advances in router hardware and software, routers can employ sophisticated algorithms to identify paths of low congestion in the network, rather than simply route along the shortest paths. This opens up the possibility for long-lived flows to dynamically choose different paths on the network depending on the congestion. Our premise also gives us a convenient model for adaptive routing schemes that can be employed in packet-switched and active network architectures.

The main consequence of our assumption is the need to evaluate allocation mechanisms not simply in the static sense, but in terms of the Nash equilibrium resulting from the selfish behavior outlined above.

A. The Nash equilibrium of Jaffe

Our first result is that the bottleneck flow control mechanism of Jaffe [1] fares rather poorly when we study the throughput under the (expected) allocations at Nash equilibrium. While it is known that, in general, max-min fair allocations lead to poor throughput, our experiments demonstrate that the expected allocations at Nash equilibrium with respect to Jaffe’s mechanism is significantly worse than the non-equilibrium (static) case.

Proposition 1: For any $\delta > 0$, there is a network and a collection of flows for which the ratio between the throughput at Nash equilibrium with respect to Jaffe’s mechanism and the optimal throughput is at most δ .

Proof: We prove the proposition by presenting a simple network topology and capacity constraints for which the Nash equilibrium under Jaffe’s allocation mechanism achieves a throughput that is arbitrarily small compared to the optimal throughput. In Figure 1, let us define the link capacities as follows, where k will be a suitably large integer (to be chosen).

$$\begin{aligned} c(u, v) &= 1 \\ c(s_0, u) &= c(v, d_0) = 1 \\ (\forall i) \quad c(s_i, u) &= c(v, d_i) = 1/(k+1) \\ (\forall i) \quad c(u, w_i) &= c(w_i, v) = 1/(k+1) - \epsilon \end{aligned}$$

Suppose there are $k+1$ flows (s_i, d_i) , for $0 \leq i \leq k$. If, for each i , flow i initially presents the path $(s_i, u), (u, v), (v, d_i)$, then the allocation made by Jaffe’s mechanism is $1/(k+1)$ for each flow. The throughput achieved by this allocation is 1. Since (s_0, d_0) does not have an alternate path, it repeatedly presents the same path. For $1 \leq i \leq k$, the flow (s_i, d_i) has an alternate path $(s_i, u), (u, w_i), (w_i, v), (v, d_i)$; however, routing along this path results in an allocation of $1/(k+1) - \epsilon$. Therefore, these flows still route along the $u-v$ link; this is the Nash equilibrium for these flows. On the other hand, if all the (s_i, d_i) flows for $i > 0$ route along their alternate path via w_i , the total throughput approaches 2 (for small enough ϵ). Thus the Nash equilibrium induced by Jaffe’s mechanism achieves only half the throughput of the optimal allocation.

By connecting a series of t networks in the spider topology, for a suitably large t , it is easy to see that the ratio between the throughput under Nash equilibrium with respect to Jaffe’s mechanism and the optimal throughput is asymptotically $1/k$. The case of two networks is shown in Figure 2. By choosing $k \gg 1/\delta$, the result follows. ■

B. Shortcomings of throughput-optimal allocations

Given a collection of paths, the problem of allocating bandwidth to maximize throughput can be written as a simple (packing) linear program [25], [26]. Even though packing linear programs admit faster algorithms than are known for general linear programs (see [27], for example), there is a number of limitations to using these in the context where flows dynamically choose paths depending on the history of allocations. Computing a throughput-optimal allocation for a given set of paths needs to be done in a centralized, non-local manner at designated nodes of the network. When each flow employs a “working set” of paths from its source to its destination and routes portions of traffic along these paths (according to their favorite probability distribution on the working set), recomputing throughput-optimal allocations for every collection of path requests is unrealistic. Finally, as one may naturally suspect, and as our experiments demonstrate, the fairness properties of the Nash equilibrium with respect to throughput-optimal allocations are quite poor.

A different approach to balancing fairness vs. throughput is to design routing algorithms that take a collection of source–destination pairs and compute a path for each pair, together with an appropriate allocation. See [2], [4], [3] for some examples. This approach is orthogonal to ours, since flows are not allowed to choose their own paths; these considerations lead to heavily centralized and expensive algorithms. Nahrstedt and Chen [5] propose a method for routing that addresses fairness and throughput considerations simultaneously; however, their analysis is not carried out for selfish flows.

C. Exploiting selfishness

Our main design philosophy is to exploit the fact that flows are selfish, and, when given a poor allocation along a path, tend to choose alternate paths that can increase their allocation. This leads flows to identify underutilized links and attempt to route their traffic along them. The resulting behavior improves utilization of the network, which also leads to improved throughput. Somewhat counter-intuitively, selfish behavior also leads to solutions that are reasonably fair: when every flow attempts to maintain an acceptably large allocation by choosing from a collection of paths, the bottomline allocation gets better.

This design philosophy suggests the following natural mechanism for allocation. Given the set of paths, one for each flow, compute the optimal throughput T that can be routed along the given paths. Then, for each flow f , compute the optimal throughput T_{-f} that can be achieved, assuming that f is absent. Finally, flow f receives an allocation proportional to T/T_{-f} (assuming the denominator is non-zero). The idea is that a flow that is crucial to achieving optimal throughput has a better claim for bandwidth than a flow that can be safely excluded from the viewpoint of throughput-optimal allocation. The latter flows are thus forced to choose paths that use underutilized links, which leads to better overall throughput.

While experiments indicate that this mechanism achieves very good throughput (especially at Nash equilibrium), it is prohibitively expensive, and shares all the drawbacks of throughput-optimal allocation based on linear programming. Note that this mechanism does achieve optimal throughput at Nash equilibrium on the spider networks of Proposition 1.

D. Mechanism M

In order to exploit the selfish behavior of flows while avoiding the deficiencies of the schemes described above, our goal is to design a mechanism that has the following desirable properties:

- (1) It should be based on local, rather than global, views of the flows and paths. This will facilitate an distributed implementation.
- (2) It should admit an efficient implementation, especially in a distributed setting.
- (3) It should allow a continuum of trade-offs between fairness and throughput optimization.

The starting point for our mechanism is the following observation about Jaffe’s mechanism. When the “tightest” link l^* is found in step (2.2) of Mechanism J , no distinction is made among the flows that use link l^* . This leads to underutilization, since one of the flows could potentially have a much higher “offer” of bandwidth from all the other links along its path, but is now constrained to obey the bottleneck on l^* . We would like to differentiate among the flows that use l^* in a principled manner, taking into account their ability to contribute to an increased throughput. We implement this in the following iterative manner. Each link offers to each flow whose path uses this link a uniform slice of its capacity. Each flow then reports to the links on its path the *second*

minimum offer it received along its path. The links then recompute a new apportionment that is proportional to the second minimum offers reported, and make this offer to the flows. This process continues for a number of rounds to be prescribed as a parameter. Finally, each flow is allocated the minimum offer it received in the last round. A more formal description follows.

In the following description, G denotes the graph, F denotes the set of flows (source–destination pairs), for each $f \in F$, $P(f)$ denotes a path that f chooses, and $r \geq 2$ is an integer parameter that denotes the number of rounds. We require $r \geq 2$ because $r = 1$ leads to an extremely poor utilization of the network, and is easily seen to be even worse than Jaffe’s allocation in terms of throughput.

Notation. In the following, for an ordered set S , $\text{2nd-min}S$ denotes the second smallest element of S .

Mechanism $M(G = ((V, E), c), F, P, r)$

- (1) For $l \in E$, let $F_l \leftarrow \{f \mid l \in P(f)\}$.
- (2) For $f \in F$, let $\beta_f \leftarrow 1$.
- (3) Repeat r times:
 - (3.1) For $l \in E$ and $f \in F_l$, let $\text{offer}(l, f) \leftarrow (\beta_f) / (\sum_{g \in F_l} \beta_g) c(l)$.
 - (3.2) For $f \in F$, let $\text{alloc}(f) \leftarrow \min_{l \in P_f} \{\text{offer}(l, f)\}$.
 - (3.3) For $f \in F$, let $\beta(f) \leftarrow \text{2nd-min}_{l \in P_f} \{\text{offer}(l, f)\}$.
- (4) Return alloc .

Remarks.

(1) *Interleaving data and control.* The allocation policies codified in mechanism M may be thought of as “control plane” operations, as opposed to the “data plane” operation where actual traffic is routed. The two operations may be seamlessly interleaved in the following way, which yields an asynchronous implementation of mechanism M . Each link (that is, the router controlling the link) maintains the β values for the flows that use that link, and uses this information in queue management (and hence the allocation). Periodically, each link executes a “renegotiation” step, where it recomputes the proportional allocation for the flows it supports, and the flows update their β values accordingly. In this mode of deployment, the role of the parameter r may be realized by having each link reset to uniform allocation after every r renegotiation steps.

(2) *Routing new flows.* An important consideration in choosing a mechanism for routing and allocation is how easily it can handle flows entering and leaving the network at various points of time. Jaffe’s mechanism has implementations [1], [28], [29] that allow for seamless handling of changes in traffic patterns. From this viewpoint, mechanism M is also quite powerful. In particular, the basic mechanism needs very little change for this context. This follows from the fact that in no round of the mechanism, do we assume that the set of flows are the same as it was in the previous round. Thus a new flow may enter the network, choose a path, and join the allocation mechanism at the “next round.” Each link along its path will assign it a

default β value equal to the average of all flows using the link. This value will then quickly converge to the correct β value for this flow. Depending on the bandwidth requirements, the flow will then explore alternate paths, and the Nash equilibrium of the entire system, including the new flow, emerges naturally.

(3) *Comparison to throughput-optimal allocation.* Mechanism M does not lead to a Nash equilibrium that achieves the optimal throughput for all networks; in fact, by suitable modification of the spider networks of Proposition 1, we can show that the ratio between the achieved throughput and the optimal throughput could be arbitrarily low. Unlike the Jaffe mechanism, experiments indicate that on realistic networks, this ratio remains quite large for a wide variety of parameter settings and various levels of congestion.

E. Complexity of M

We state the following proposition concerning the complexity of M ; the proof is omitted.

Proposition 2: Let G denote an m -link graph, F a set of n flows in G , P the set of paths for flows in F , and r an integer ≥ 2 . Let $Z = \{(f, l) \mid l \in P(f)\}$. Mechanism $M(G, F, P, r)$ takes $O(r|Z|)$ time, and admits a distributed implementation with r rounds with n messages of size $O(1)$ in each round (one per flow per round).

The running time $O(r|Z|)$ is quite attractive from a practical viewpoint. In fact, we will soon present an enhancement of M that will essentially allow us to treat $r = O(1)$ and hence achieve an $O(|Z|)$ time implementation. By contrast, note that Jaffe’s mechanism might take $\Omega(nm)$ time in the worst case (and $\Omega(nm^2/|Z|)$ on the average). Thus mechanism M , besides being extremely simple, achieves significant improvement in efficiency over Jaffe’s mechanism, and certainly over LP-based throughput-optimal mechanisms.

An acceleration method for M . Suppose a flow f that uses a link l has the highest value of $2\text{nd-min}_{k \in P(f)}\{\text{offer}(k, f)\}$ among all flows that use link l . Then f will receive the highest offer from l in the next round, and in all likelihood, it will also receive the highest offer from other links as well. This gradually reinforces the dominance of f over the other flows that use l . We note that a similar effect can be achieved by modifying step (3.1) in mechanism M as follows. Let $e > 0$ be a real number.

$$(3.1)' \text{ For } l \in E \text{ and } f \in F_l, \text{ let} \\ \text{offer}(l, f) \leftarrow (\beta_f^{1+e}) / (\sum_{g \in F_l} \beta_g^{1+e}) c(l).$$

A naive estimate of the speedup due to this modification (under suitable assumptions of correlation of the dominance of flows) is that the effect of r rounds may now be achieved in $O(\log_{1+e} r)$ rounds. Our experiments show that this is indeed the case.

IV. EXPERIMENTS

In this section, we present a variety of experimental results that substantiate the throughput and fairness properties of the Nash equilibrium induced by mechanism M . We first discuss the performance measures computed and control variables

used, then present some implementation notes, including network topology, computation of Nash equilibrium, etc., and finally analyze the experimental results. To avoid inundating the reader with several combinations of parameters, we only present a selection of the most important trends, chosen from nearly 5000 runs of the mechanism.

A. Performance Indices and Control Parameters

The performance indices we focus on are based on the indices for comparing allocations described in Section II. Specifically, while evaluating the fairness properties of mechanism M in terms of the indices ω, κ, γ , we compare the expected allocation at the Nash equilibrium induced by M against the expected allocation at the Nash equilibrium induced by Jaffe allocation; similarly, while evaluating the throughput properties of M , we report the throughput index τ of the expected throughput at the Nash equilibrium induced by M with respect to the expected throughput at the Nash equilibrium induced by throughput-optimal allocation (via linear programming). All indices are expressed as percentages.

The control parameters in our experiments are the number r of rounds used in mechanism M , the exponent e used in (the accelerated version of) M , and the congestion in the network, measured by the number of flows in the network. Where the network and the set of flows are understood, we simply write $M(e, r)$ to denote mechanism M with exponent e and number of rounds r .

In addition, we study various parameters of natural import in our setting, including the sizes and distributions on the working set of paths for flows at Nash equilibrium, and the correlation between the size of the working set and the allocation at equilibrium.

B. Implementation Notes

Topology. In our experiments, we considered networks generated according to an evolving preferential attachment model (see, e.g., [30]). These and the related evolving copying models (e.g., [31], [32]) are popular in generating network topologies that satisfy various observed properties of the Internet and other large networks. For example, they satisfy the degree power law, which states that the probability that a uniformly picked node has degree d is proportional to d^{-c} for some constant $c > 1$. The Internet graph reportedly has the constant $c \approx 2.18$ [33].

To generate an N -node graph according to the preferential attachment model, we start with a small random “core” of 10 nodes generated according to the $G_{n,p}$ model with $n = 10$ and $p = 1/2$. Then we add each of the $n - 10$ nodes in turn. When we add a new node, it connects to $D = 5$ existing nodes, chosen in the following way. Each neighbor is chosen uniformly at random with probability $\alpha \approx 0.4$ and with probability $1 - \alpha$, is chosen to be a node with probability proportional to its current degree. These settings ensure that the resulting graph has a power law exponent close to 2.18.

Note that network constructed is quite sparse (degree = 5), but has excellent connectivity properties, similar to the inter-domain network. Rather than vary the degree parameter in

our experiments, we achieve essentially the same effect by choosing various levels of congestion (between 500 and 2000 flows); we validated this latter equivalence by independent experiments, whose results we do not report here.

In all our experiments, the graph consisted of 200 nodes and degree 5.

Computing Nash equilibria. In general, computing a Nash equilibrium of a multi-agent game is computationally very hard, with no known polynomial-time algorithm (see [24] for a summary of the complexity of Nash). As described earlier in Section II-C, we employ the best response dynamics of the game, a form of local search based approximate computation, described below. The basic idea is that a flow maintains a working set of up to 10 paths and a probability distribution over this set, and periodically explores the network to find alternate paths to add to its working set. The probability distribution on the set is periodically updated to reflect the allocations received along the paths. Note that the computed equilibrium is indeed an approximate equilibrium (and not just a local optimum encountered by local search), for the following reason. For a flow f with expected allocation $a(f)$ under the computed distribution, the convergence of the search implies that if all the other flows g employ their distribution on their working set of paths, then f can achieve an expected allocation of $a(f)$; furthermore, since f has explored alternatives and decided that choosing any other path does not lead to a better expected allocation than $a(f)$. We confirmed by our experiments that the probability distribution over the working set of paths did indeed lead to an expected allocation that converged fairly rapidly.

In all our experiments, we set the number of iterations of the local search to be 100. (The one exception is the experiment where we study the rate of convergence to Nash equilibrium.)

Computing throughput-optimal allocations. We employed a linear programming solver (GNU LP Kit, see <http://www.gnu.org/software/glpk/glpk.html>) to compute throughput-optimal allocations. The method was based on the simplex method applied to the natural linear program derived from the capacity constraints.

C. Results and Analysis

Jaffe vs. Jaffe at Nash equilibrium. We briefly note that the throughput performance of Jaffe’s mechanism is quite poor, especially if we consider the expected throughput under the Nash equilibrium induced by Jaffe’s mechanism. We computed the Nash equilibria for Jaffe’s allocation mechanism as well as for the throughput-optimal allocation mechanism. Without Nash equilibrium, when given shortest paths between source–destination pairs, Jaffe’s allocation yields between 60 and 65% of the optimal throughput; under Nash equilibrium, this degrades further to between 45 and 50% of the optimal throughput. In other words, Jaffe’s mechanism, combined with selfish behavior of flows, leads to an equilibrium where more than half the usable capacity of the network is wasted!

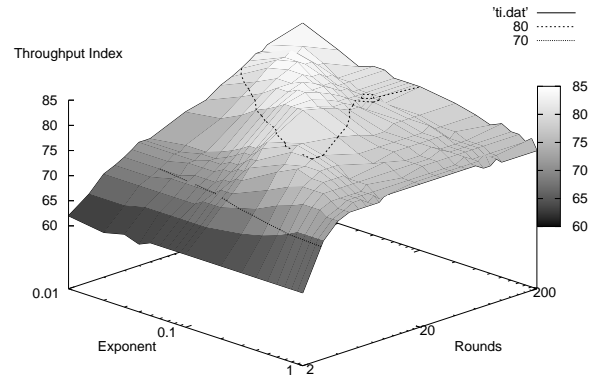


Fig. 3. Throughput index τ vs. (e, r)

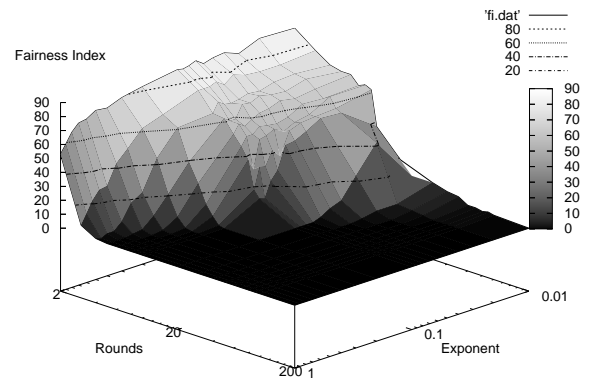


Fig. 4. Fairness index ω vs. (e, r)

Overall performance of M . To study the overall performance characteristics of mechanism M , we present five 3-dimensional intensity plots, each with the following semantics. Along the x -axis, we plot the exponent e , along the y -axis, we plot the number of rounds r , and along the z -axis, we plot the index of interest. In all cases, the exponent and round are plotted on a logarithmic scale. In all of these plots, the number of flows was set to 1500.

Figures 3 through 6 plot the four indices described earlier (see Section II and IV-A). The salient observations from these figures include the following.

(1) The best value of throughput index τ (Figure 3) is achieved with low exponents and a large number of iterations (e.g., $e = 0.01$ and $r = 100$). Increasing an exponent is an inexpensive way to reduce the number of rounds (for example, $e = 0.1$ and $r = 10$ is essentially as good as $e = 0.01$ and $r = 100$). This bears out our rationale that led to the accelerated version of M .

(2) Figure 4 plots the “fairness index” of mechanism M , defined as the minimum allocation index ω of the expected allocation of M at Nash equilibrium with respect to the

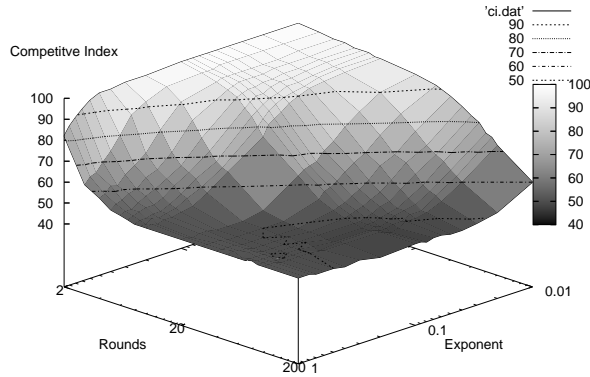


Fig. 5. Competitive index κ vs. (e, r)

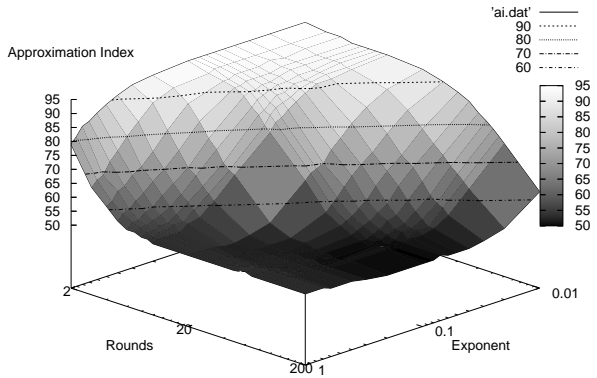


Fig. 6. Global approx. index γ vs. (e, r)

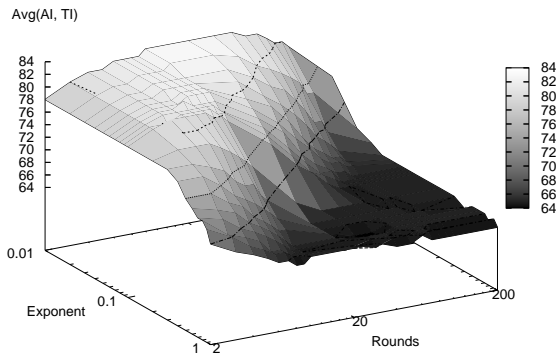


Fig. 7. Average of γ and τ vs. (e, r)

expected allocation of Jaffe at Nash equilibrium. Here we see that a small exponent and a small number of rounds achieves a very high fairness index (close to 90), and the index remains fairly large even for moderately large exponent and small number of rounds (e.g., 70 at $e \approx 1$ for small r). This is good news, since the latter setting also achieves high values of the throughput index.

(3) On the other hand, for large values of either e or r , at least one flow suffers a significant degradation in its allocation, which drastically affects the minimum allocation index. Figure 5 plots the competitive index κ , which measures how many flows receive at least as much as they did under Jaffe's Nash equilibrium. Here again, the range of small e and small r achieves excellent performance (close to 100), while a large range of e and r still lead to competitive allocations where at least half the flows receive at least the minimum of Jaffe allocation.

(4) Figure 6 plots the global approximation index γ of M with respect to Jaffe (both under Nash equilibria); this gives a broader picture of the fairness properties of M . Recall that to achieve a high value of γ , it is important to have a large fraction of flows being allocated a large fraction of their allocation under Jaffe's Nash equilibrium. For example, to achieve an $\gamma = 80$, either 100% of the flows have to receive at least 80% of their allocation under Jaffe's Nash equilibrium, or 50% of the flows receive 100% of their allocation and the other 50% receive at least 60% of their allocation under Jaffe's Nash equilibrium. The performance of M with respect to this measure is excellent (80 or higher), especially at the low end of the (e, r) plane.

(5) Figure 7 allows us to simultaneously evaluate the throughput and fairness properties of M . Namely, if we care equally about both measures, a natural performance measure to employ is the arithmetic average of τ and γ , and this is precisely what this figure illustrates. Notice that the lowest reported value for this measure is 64, which is saying that mechanism M simultaneously approximates, to a factor of at least $2/3$ (and much higher for large ranges), the best of both worlds — fairness as compared to Jaffe's Nash equilibrium and throughput as compared to the Nash equilibrium induced by throughput-optimal allocations!

Effect of exponent e . In Figures 8 through 10, we study the role of the parameter e , for various congestion values (corresponding, respectively, to the number of flows set to 500, 1000, 1500, and 2000). In this experiment, parameter r is set to 5 rounds.

(6) Figure 8 shows that for low values of e , mechanism M handles congestion very well, while for higher values of e , γ drops fast as congestion increases. Note that even in the latter range, γ is at least 50 (e.g., at least 70% of the flows receive at least 70% of their Jaffe allocation under Nash equilibrium).

(7) Figure 9 shows that for all levels of congestion, larger exponents uniformly lead to significantly better throughput indices, but fall off after a peak. The drop is larger for increased congestion. This shows that merely increasing the

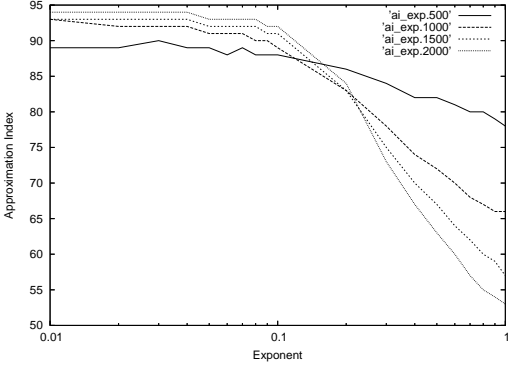


Fig. 8. γ vs. e ; $r = 5$

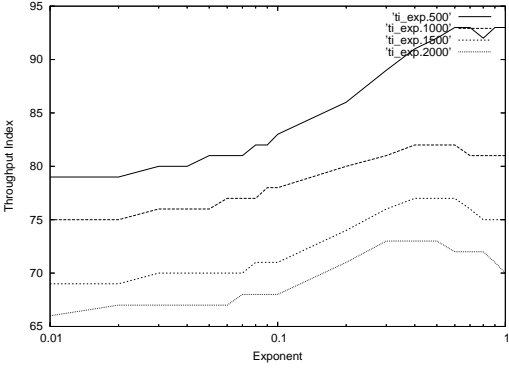


Fig. 9. τ vs. e ; $r = 5$

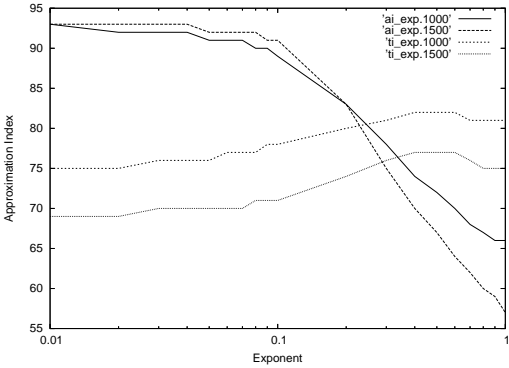


Fig. 10. γ and τ vs. e ; $r = 5$

exponent is a greedy strategy that makes the proportional allocations more polarized, and this, in turn, affects utilization, and ultimately, throughput.

(8) Figure 10 shows a subset of the plots from Figures 8 and 9 (corresponding to 1000 and 1500 flows). This plot allows us to simultaneously study the index γ of fairness and the throughput index τ for mechanism M , as functions of e . Notice that there is a clear trade-off between fairness and throughput. What is perhaps surprising is that it is possible to set the exponent value so as to achieve simultaneously good values for these indices (e.g., 75 at $e \approx 0.2$). The points of cross-over of the fairness and throughput curves present an interesting choice for e , since these curves are clearly concave, and hence their sum is concave as well. Therefore, if

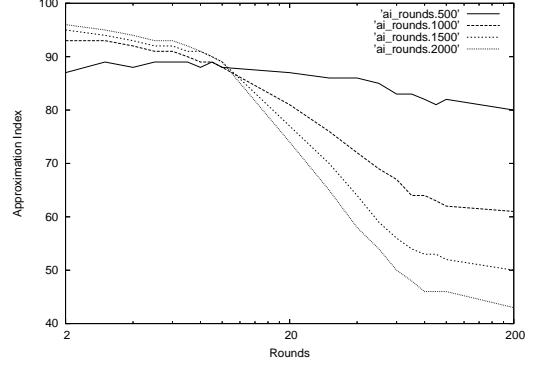


Fig. 11. γ vs. r ; $e = 0.05$

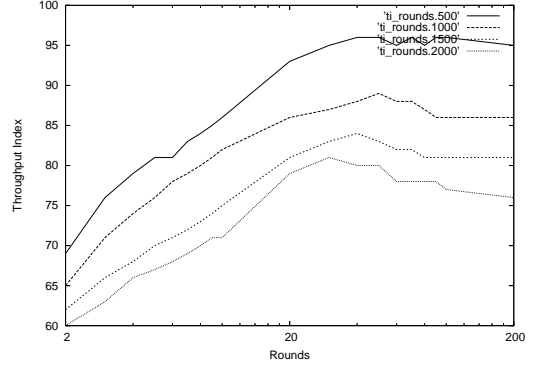


Fig. 12. τ vs. r ; $e = 0.05$

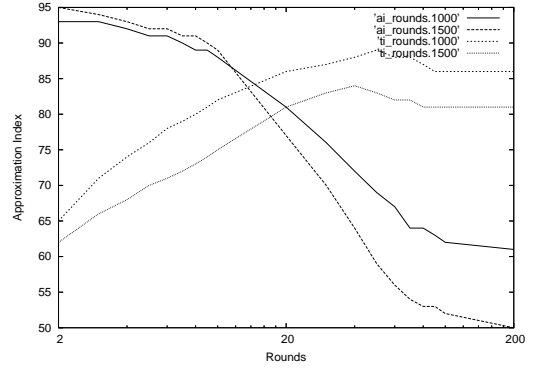


Fig. 13. γ and τ vs. r ; $e = 0.05$

we care about maximizing any weighted sum of fairness and throughput, this choice of e is natural.

Effect of r . In Figures 11 through 13, we study the role of the parameter r , for various congestion values. In this experiment, parameter e is set to 0.05.

(9) The main observation on the role of r from Figures 11 through 12 is that similar to e , a larger r leads to a higher τ , and a smaller r leads to a higher γ . However, we observe that the ascend (in the case of τ) and descend (in the case of γ) are quite sharp (unlike the case of e). Figure 13 shows that for $r \approx 10$, quite large values (e.g., 80) of both indices are achievable. Beyond the crossover point, while τ does not improve significantly, γ falls quite rapidly. Therefore, it is crucial to stay close to the crossover point.

Approximation Factor vs. congestion. The global approximation index γ analyzed earlier gives a summary statistic of how individual flows’ allocations under M ’s Nash equilibrium compare to their allocations under Jaffe’s Nash equilibrium. We now present a finer analysis of the approximation factors achieved. Figure 14 shows the approximation factors (with respect to the Jaffe allocation under Nash equilibrium) for various values of congestion. In this figure, M was employed with the fixed settings $e = 0.05$ and $r = 5$.

(10) As Figure 14 shows, the performance of M in terms of fairness is uniformly good, considering that for all congestion values, $\geq 65\%$ of the flows receive $\geq 90\%$ of their Jaffe–Nash allocations, and $\geq 90\%$ of the flows receive $\geq 70\%$ of their allocations. In fact, as congestion increases, M guarantees a large fraction of the flows (80% for the case of the most congested setting) over 90% of their Jaffe–Nash allocation!

Jaffe vs. Mechanism M

(11) Figure 15 shows that as congestion increases, the global approximation index γ of M (here used with $e = 0.05$ and $r = 5$) gets better and approaches 100, while the throughput index τ of M consistently outperforms that of Jaffe by a wide margin (at least 15%). This is very concrete evidence that it is possible to simultaneously achieve very good levels of fairness and throughput, especially at Nash equilibrium, by carefully picking the allocation mechanism.

Nash equilibria.

(12) Figure 16 presents a statistical summary of the sizes of working sets employed by the flows under equilibrium. If a flow f uses a set of t paths with probabilities μ_1, \dots, μ_t , then the *entropy* of the distribution μ is defined by $H(\mu) = -\sum_i \mu_i \log_2(1/\mu_i)$; an entropy value of η may be interpreted roughly as saying that the flow has 2^η paths in its working set with uniform distribution among them. We define $2^{H(\mu)}$ as the “average number of paths” employed by f . In this figure we plot the fraction of flows with various values of average number of paths. The curves correspond to varying numbers of iterations in the approximate computation of Nash equilibria. Recall that we used 100 iterations of a local search algorithm for computing Nash equilibria. (Carrying out more than 1000 iterations did not qualitatively change the convergence properties illustrated in Figure 16.) Notice that over 50% of the flows have just one path in their working set; on the other hand, nearly 30% of the flows have a nearly uniform distribution on 10 paths. (The “jumps” in the curve correspond to an integral number of paths — at Nash equilibrium, the expected allocation along every path in the working set must be identical, therefore, the only distributions that arise are uniform on the working set of paths; for a nearly uniform distribution on t paths, the entropy is $\approx \log_2 t$ and hence the average number of paths is $\approx t$.)

(13) A striking fact that we observed while computing similar statistics for the Nash equilibrium of Jaffe’s mechanism is that *every* flow maintained a working set of ten paths with uniform distribution on the set, and the equilibrium converged quite rapidly to this scenario. Together with the fact that M

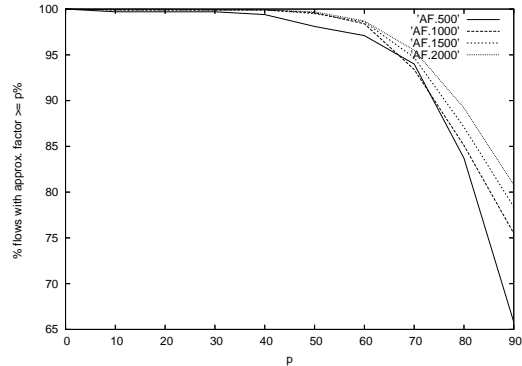


Fig. 14. Approximation Factor at $e = 0.05$, $r = 5$

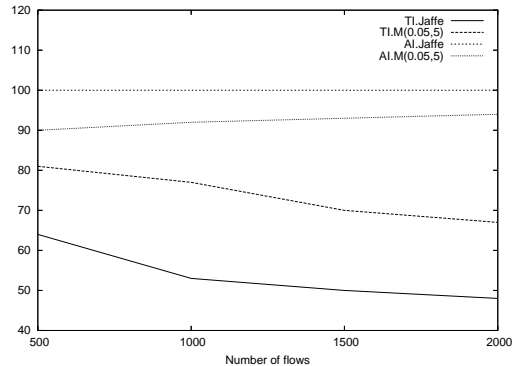


Fig. 15. Jaffe vs. $M(0.05, 5)$

leads to a Nash equilibrium most flows use exactly one path, the following natural question arises: why does this happen, and what, if any, is the correlation between the working set size and the allocation that a flow receives under Nash equilibrium? Figure 17 presents an interesting answer to this question. The flows that use a small number of paths (especially the ones that use just one path) are *precisely* the flows whose allocation under the Nash equilibrium of M (using those small sets of paths) are better than their allocation under the Nash equilibrium of Jaffe. Conversely, the flows that have a working set of ten paths are precisely the ones whose allocations are worse than those under Jaffe. The fact that every flow has uniform distribution on ten paths under Jaffe’s Nash equilibrium suggests that Jaffe’s mechanism limits the allocation of a flow along every possible path it attempts, so that there is no path for any flow from which it has no incentive to deviate. On the other hand, it is precisely by allowing flows to find paths from which they have no cause to deviate that mechanism M leads to improved allocations for such flows, and consequently, to better throughput.

V. CONCLUSIONS AND FUTURE WORK

We have presented a mechanism that simultaneously approximates optimal throughput and fairness measures, and admits deployment to optimize for any convex combination of these two criteria. Our experiments provide a detailed validation for the performance of our mechanism, and also contrast it to Jaffe’s max-min fair allocation mechanism.

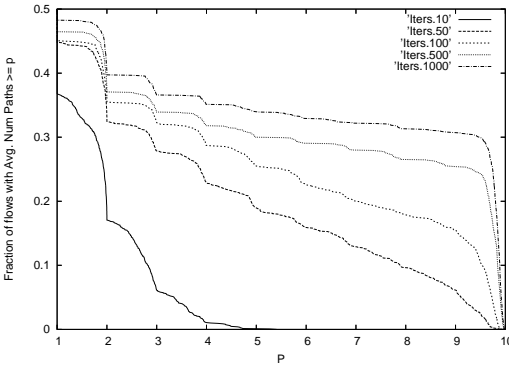


Fig. 16. Convergence of Nash eq. for $M(0.05, 5)$

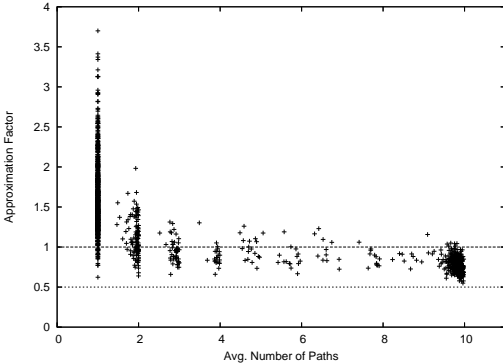


Fig. 17. Avg. number of paths in Nash equilibrium for $M(0.05, 5)$ vs. approximation factor. For Jaffe, every flow has nearly uniform distribution on ten paths.

We believe that exploiting the selfish and rational behavior of flows to design good routing algorithms is a novel idea, and deserves further study. Some specific questions to study include understanding the behavior of the allocations that arise from mechanism M when it is deployed in an asynchronous distributed setting, where the various instances of the mechanisms operate with different parameter settings. Another question is to identify parameters that allow faster convergence to Nash equilibria, especially when flows enter and leave the system frequently.

REFERENCES

- [1] J. Jaffe, "Bottleneck flow control," *IEEE Trans. Communications*, vol. 26, pp. 954–962, 1981.
- [2] J. Kleinberg, Y. Rabani, and E. Tardos, "Fairness in routing and load balancing," in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, 1999, pp. 568–578.
- [3] A. Goel, A. Meyerson, and S. Plotkin, "Combining fairness with throughput: online routing with multiple objectives," in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, 2000, pp. 670–679.
- [4] A. Kumar and J. Kleinberg, "Fairness measures for resource allocation," in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000, pp. 75–85.
- [5] S. Chen and K. Nahrstedt, "Maxmin Fair Routing in Connection-Oriented Networks," in *Proc. of Euro-Parallel and Distributed Systems Conference*, Vienna, Austria, July 1998, pp. 163–168.
- [6] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter, "A survey on networking games in telecommunications," to appear in *Computers and Operations Research*, 2004.
- [7] R. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973.
- [8] T. Roughgarden and E. Tardos, "How bad is selfish routing?" *Journal of the ACM*, vol. 49, no. 2, pp. 236–259, 2002.
- [9] E. Altman and L. Wynter, "Equilibrium, games, and pricing in transportation and telecommunication networks," *Networks and Spatial Economics*, vol. 4, pp. 7–21, 2004.
- [10] A. Orda and N. Shimkin, "Incentive pricing in multi-class communication networks," in *IEEE INFOCOM*, 1997.
- [11] H. Yaiche, R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing of elastic connections in broadband networks: Theory and algorithms," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, 2000.
- [12] S. Low and D. Lapsley, "Optimization flow control—I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.
- [13] S. Low, "Equilibrium bandwidth and buffer allocations for elastic traffics," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 373–383, 2000.
- [14] P. Marbach and R. Berry, "Downlink resource allocation and pricing for wireless networks," in *IEEE INFOCOM*, 2002.
- [15] P. Marbach, "Priority service and max-min fairness," in *Proceedings of INFOCOM*, 2002.
- [16] K. Park, M. Sitharam, and S. Chen, "Quality of service provision in non-cooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness," in *Proceedings of the 1st International Conference on Information and Computation Economics*, 1998.
- [17] A. Lomonosov, M. Sitharam, and K. Park, "Stability vs. optimality trade-off in game theoretic mechanisms for QoS provision," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, 2003, pp. 28–32.
- [18] S. Gollapudi and D. Sivakumar, "A mechanism for equitable bandwidth allocation under QoS and budget constraints," in *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS 2004)*, 2004.
- [19] A. Lazar, A. Orda, and D. Perdarakis, "Virtual path bandwidth allocation in multiuser networks," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 861–871, 1997.
- [20] J. F. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, pp. 289–295, 1951.
- [21] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Proc. of 16th Annual Symposium on Theoretical Aspects of Computer Science*, 1999, pp. 404–413.
- [22] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler, "Near-optimal network design with selfish agents," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, 2003, pp. 511–520.
- [23] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, "Price of stability for network design with fair allocation of costs," in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, 2004, to appear.
- [24] C. Papadimitriou, "Algorithms, Games, and the Internet," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, Crete, Greece, 2001, pp. 749–753.
- [25] V. Chvatal, *Linear Programming*. Freeman, 1983.
- [26] H. Karloff, *Linear Programming*. Birkhauser, 1991.
- [27] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998, pp. 300–309.
- [28] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987.
- [29] Y. Afek, Y. Mansour, and Z. Ostfeld, "Phantom: a simple and effective flow control scheme," *Computer Networks*, vol. 32, no. 3, pp. 277–305, 2000.
- [30] S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the Internet's large-scale topology," *Proceedings of the National Academy of Sciences*, vol. 99, no. 21, pp. 13 382–13 386, October 2002.
- [31] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "Stochastic models for the web graph," in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000, pp. 57–65.
- [32] C. Cooper and A. Frieze, "A general model of web graphs," *Random Structures and Algorithms*, vol. 22, no. 3, pp. 311–335, 2003.
- [33] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM*, 1999, pp. 251–262.