Fast Convergence of Selfish Rerouting *

Eyal Even-Dar † Yishay Mansour ‡

October 11, 2004

Abstract

We consider n anonymous selfish users that route their communication through m parallel links. The users are allowed to reroute, concurrently, from overloaded links to underloaded links. The different rerouting decisions are concurrent, randomized and independent. The rerouting process terminates when the system reaches a Nash equilibrium, in which no user can improve its state.

We study the convergence rate of several migration policies. The first is a very natural policy, which balances the expected load on the links, for the case that all users are identical and apply it, we show that the rerouting terminates in expected $O(\log\log n + \log m)$ stages. Later, we consider the Nash rerouting policies class, in which *every* rerouting stage is a Nash equilibrium and the users are greedy with respect to the next load they observe. We show a similar termination bounds for this class. We study the structural properties of the Nash rerouting policies, and derive both existence result and an efficient algorithm for the case that the number of links is small. We also show that if the users have different weights then there exists a set of weights such that every Nash rerouting terminates in $\Omega(\sqrt{n})$ stages with high probability.

1 Introduction

Routing is one of the most basic tasks in networking. Traditionally it has been view as a large multi-valued optimization problem, which the network provider has to address. Clearly, the selection of the load on the various routes greatly impacts the overall network performance. As part of the optimization process, the goals of different individual users are contrasted with the desire to achieve a high utilization of the network resources.

A different perspective of the routing problem, is to allow individual users to specify the path on which their traffic would be routed. In such setting each individual user optimizes her own utility function, and naturally tries to route her traffic on the least loaded path. An appropriate solution concept for such system is that of Nash equilibrium [17], where no user can benefit from rerouting her traffic. Although the distributed view of network routing has many interesting features, it carries the difficulty that the selfish users decisions might negatively impact the overall performance of the system.

Because of the highly attractive nature of decentralize distributed routing, networks researchers have looked for over a decade into issues involved in the design and implementation of such mechanism. One important line of research has been to try and "determine" the resulting equilibrium, with the motivation of allowing the network administrator to predict and control the resulting equilibrium. Therefore, cases in which the equilibrium is unique have been highly desired [13, 18].

The main focus in theoretical computer science, in this research direction, has been the study of the quality of the resulting Nash equilibrium. The work of Koutsoupias and Papadimitriou [12] defined the *coordination ratio*, which is the ratio of the cost of an optimal global objective function to the cost of the worst Nash equilibrium. Many following works have studied the coordination ratio in a wide range of routing and job scheduling scenarios [3, 5, 6, 11, 12, 19].

We consider a different aspect of Nash equilibrium and are interested in "how fast Nash equilibrium is reached" rather than "the quality of the Nash equilibrium". Traditionally, the convergence to Nash equilibrium was studied using the Elementary Step System, where only one user can reroute at each stage [8, 9, 13, 18]. We consider algorithms that have no centralized unit and decisions are made locally and concurrently. The decisions are no longer deterministic but are stochastic and a user action is a distribution over the links. Therefore, if an equilibrium point is not ex-

*This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, by a grant from the Israel Science Foundation and an IBM faculty award. This publication only reflects the authors' views.

†School of Computer Science, Tel-Aviv university, email evend@post.tau.ac.il

‡School of Computer Science, Tel-Aviv university, email mansour@post.tau.ac.il.

post Nash[1], several users will try to migrate in the next time step.

Although the Elementary Step System has attractive theoretical properties, it has also very clear drawbacks. First, since only one user is allowed to reroute in each stage, the convergence time is at least $\Omega(n)$. Second, the implementation of a centralized controller is unattractive and fairly difficult in a truly distributed system. Third, by avoiding the issue of concurrency, many realistic issues that are involved in the decision to reroute "disappear", such as the fear of a user that after the rerouting she would be worse off.

We consider a simple model of m parallel links (with possibly different speeds). The links are shared by n identical selfish users, at each stage each user selects one of the parallel links. The cost of each user is the number of users that share the link with her, and therefore users would like to reroute from overloaded links to underloaded links. In our dynamic model, at each stage, each user decides stochastically whether to stay or reroute her traffic to a different link. We assume that each user has access only to the load on the links and that the users are anonymous, thus all identical users that share the same link select the same stochastic rerouting policy.

For identical links, we first consider a simple policy, `BALANCE`, in which every link has the same expected load during each rerouting stage. Specifically, for an overloaded link (a link with load $L > n/m$), each user stays with probability $n/(Lm)$ and reroutes otherwise. Given that a user decided to reroute, the probability of rerouting to an underloaded link (a link with load $L' < n/m$), is proportional to $n/m - L'$. We show that the expected number of stages until `BALANCE` converges to Nash equilibrium is $O(\log\log n + \log m)$. We extend the result to links with different speeds and derive a similar convergence bound.

One weakness of `BALANCE` is that a selfish user might not have an incentive to follow it. For example consider two identical links, one with 200 users and one with 400 users. In the `BALANCE` policy each user from the overloaded link reroutes with probability 1/4. Consider a selfish user using the overloaded link. When this user considers its best response it computes the expected load on the links, ignoring its own action. For such a user the expected load on the underloaded link is $300 - 1/4 = 299.75$ and on the overloaded link is $300 - 3/4 = 299.25$. Being greedy, the only best response of the user is not to reroute, rather than to reroute with probability 1/4. This motivates our study of *Nash rerouting policies* that are a Nash equilibrium in every stage. In Nash rerouting policies, by definition, no greedy user has any incentive to deviate from them. In the above example, the Nash rerouting policy reroutes with probability 99.5/399, and a user from the overloaded link observes an expected load of 299.5 on both links, ignoring its influence.

We study a very natural class of Nash rerouting policies that has the following properties: (1) all identical users that share a link have the same policy, (2) no user from an underloaded link reroutes, and (3) no user from an overloaded link reroutes to a different overloaded link. We show that if at every stage Nash rerouting policy is used, then the expected number of stages until convergence is $O(\log\log n + \log m)$ for identical users but can be $\Omega(\sqrt{n})$ even for two links for weighted users. In addition we study the structure of this class, and show that there always exists such a Nash rerouting policy. In contrast we show that there are examples in which there is no Nash rerouting policy of the following form: (1) In each overloaded link, all users have the same probability to stay or reroute. (2) All users that decided to reroute use the same probability distribution to select an underloaded link. (Note that the `BALANCE` policy has this property.) Finally, we show that a Nash rerouting policy can be computed in time $O(2^{m^2})$.

Related work Research in the communication networks community studied various aspects of selfish rerouting for over a decade. In [18] it was shown that for general topology, under certain conditions, a unique equilibrium exists. They also show that in an Elementary Step System (ESS), where only one user reroutes in each time step, the best response dynamics converges to Nash equilibrium for two parallel links. In [13] it was shown that for parallel links and a specific cost function there exists a unique Nash equilibrium. They also show that in the ESS model the best response dynamics converges. In [14] selfish users can select their priority and pay accordingly, and the result is a unique Nash equilibrium which is also Max-Min Fairness.

The work of [8] studied the convergence to Nash equilibrium under the ESS model, for several settings: identical, related and unrelated machines and identical/weighted jobs. For each setting different lower and upper bounds on the convergence rate were shown. For instance, in the case of identical machines and weighted jobs, the convergence might take exponential time, under one natural selection policy, and linear time using a different selection policy.

The work of [9] studied a randomized model in which each user can select a random delay over continuous time. The continuous time implies that only

[1] An ex-post Nash remain an equilibrium even *after*, every user observe all the random choices of all other users.

one user tries to reroute at each specific time, and thus this model is very similar to the ESS model. In that model rerouting succeeds only if the user lowers its load. The work shows a simple randomized algorithm in which the expected number of rerouting attempts, until convergence to a Nash equilibrium, is polynomial in the number of links m and users n.

The work of [16] studied a model in which only one user is allowed to move in each time step, but the main interest was not in the equilibrium point but on the social value after a short best response path and therefore they were interested in the convergence time to an approximate solution and not to exact solution as we are.

When contrasting our work with the ESS dynamics, one should first note that in the ESS dynamics, by definition, the worst case number of steps is at least linear in the number of users. Another important difference is that in the ESS model, at each time step the system "improves" while in our setting the system might also "deteriorate", which is a major conceptual difference between ESS and concurrent rerouting.

In [10] several types of dispersion (anti coordination) games were defined. Simple concurrent algorithms that are similar to `BALANCE` were tested empirically and demonstrated logarithmic convergence rate but only a linear upper bound on the convergence rate was shown. We derive for those simple policies a logarithmic upper bound.

A somewhat related topic are games of *throwing balls in to bins* (see [15] for a survey). The game there is a *single shot*, once a ball is assigned to a bin it does not change its bin, and the goal is to minimize the maximum load. In this work we are interested in the dynamics that converge to a state where all the loads are almost identical, and our main parameter is the time to convergence, and the game theoretic motivation of the selfish users to follow the prescribed policy.

2 Model Description

Our model is composed from a set of n identical users $x_1, \dots, x_n$ [2], and a set of m parallel links $M_1, \dots, M_m$ with speeds $s_1, \dots s_m$, respectively. (When we will refer to identical links we mean that $s_i = 1$ for $i \in [1, m]$.) The time would progress in stages from $t = 1$. At each stage t each user x_i is on one of the links M_k, and there is a function g_t such that $g_t(x_i) = k$ if and only if user x_i is on link M_k at time t. We define G to be the set of all configurations, i.e., $g_t \in G$. The number of users that use link M_k at time t is $n_t(k) = |\{x_i | g_t(x_i) = k\}|$ and the load on M_k is $L_t(M_k) = n_t(k)/s_k$.

[2] Only in Subsection 3.3 we consider users with weights

The average system load is $\bar{L} = n / \sum_{i=1}^{m} s_i$ (for identical links it is simply $\bar{L} = n/m$). For simplicity of both the algorithms and analysis we assume that $\bar{L}$ is an integer. A link M_k is *balanced* at time t if its load is $\bar{L}$, i.e., $L_t(M_k) = \bar{L}$, it is *overloaded* if its load is above $\bar{L}$, i.e., $L_t(M_k) > \bar{L}$, and it is *underloaded* otherwise, i.e., $L_t(M_k) < \bar{L}$. A strategy P^t of the users $x_1, \dots, x_n$ is a product distribution $P_1^t \cdots P_n^t$, where $P_i^t = \langle p_i^t(1) \dots p_i^t(m) \rangle$ is a distribution on $M_1, \dots, M_m$. Let P_{-i}^t be the joint strategy of all users at time t, except user x_i. We assume that the users are selfish and rational, thus a user would like to minimize the load on the link that she uses and can use rerouting (moving to a different link) to achieve this goal. Through the paper we consider policies in which only users from overloaded links reroute to underloaded links.

In this paper we deal with two types of equilibria. One is for the final state of the system, when all links are balanced, and one is for each rerouting stage. We start by defining the final state equilibrium.

DEFINITION 2.1. *A system is in* Nash-equilibrium *at time t if for every links M_i and M_j we have $L_t(M_i) \leq L_t(M_j) + \frac{1}{s_j}$.*

Our second type of equilibria is an equilibrium during each rerouting stage. It considers the joint strategy of all users in the system, such that no user would like to change its rerouting policy.

DEFINITION 2.2. *A joint strategy P^t is an ϵ-Nash* rerouting policy *if no user can gain more than ϵ (in expected load) by deviating from P^t. Formally, for every user x_i and link M_j such that $p_i^t(j) > 0$ then $E_{P_{-i}^t}[L_{t+1}(M_j)] \leq \epsilon + \min_k E_{P_{-i}^t}[L_{t+1}(M_k)]$. A Nash* rerouting policy *is a* 0-Nash rerouting policy.

The convergence proofs use extensively large deviation bounds, specifically the relative Chernoff bound [2], as stated in the following lemma.

LEMMA 2.1. *[2] Let $0 < p < 1$, let $Z_1, \dots, Z_n$ be independent binary random variables and $Z = \sum_{i=1}^{n} Z_i$, where $\sum_{i=1}^{n} \frac{E[Z_i]}{n} = p$, and let $\hat{p} = \frac{\sum Z_k}{n}$. Then,*

$$\mathbf{P}\left(p \leq \hat{p} + \sqrt{\frac{2p\ln(1/\delta)}{n}}\right) \geq 1 - \delta \quad 0 \leq p \leq 1$$

$$\mathbf{P}\left(p \geq \hat{p} - \sqrt{\frac{3p\ln(1/\delta)}{n}}\right) \geq 1 - \delta \quad \frac{\ln(1/\delta)}{3n} \leq p \leq 1$$

$$\mathbf{P}\left(p \geq \hat{p} - \frac{2\ln(1/\delta)}{n}\right) \geq 1 - \delta \quad 0 \leq p \leq \frac{\ln(1/\delta)}{3n}$$

Another bound that is used through the paper is a bound on the probability of hitting the mean.

LEMMA 2.2. *Let $Z_1, \dots, Z_n$ be i.i.d $\{0,1\}$-random variables with $\mathbf{P}(Z_i = 1) = p$ and $Z = \sum_{i=1}^{n} Z_i$. Then*

Input : Links: M_1, M_2, Users: $X = \{x_1, \ldots, x_n\}$;
Let $t = 1$; $d_1 = |L_t(M_1) - L_t(M_2)|$;
$\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$;
while $d_t \geq 2$ **do**
  for x_i *s.t* $g_t(x_i) = M_{\texttt{over}}$ **do in parallel**
    Move with probability $d_t/(2L_t(M_{\texttt{over}}))$
  end
  $t = t + 1; d_t = |L_t(M_1) - L_t(M_2)|$;
  $\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$;
end

Algorithm 1: Distributed algorithm, `BALANCE`, for two identical links.

$\mathbf{P}(Z = \lceil pn \rceil) \geq \frac{1}{c\sqrt{\lceil pn \rceil}}$ *for some constant* $c > 0$ [3]. *If* $pn = q$ *is an integer then,* $\mathbf{P}(Z = q) \geq \frac{1}{\sqrt{2\pi q}}$

Notation: The notation $g = \tilde{O}(f)$ implies that there are constants c_1 and c_2 such that $g \leq c_1 f \ln^{c_2}(f)$.

3 Two Links

3.1 Two Identical Links

In this subsection we analyze the convergence rate of a distributed algorithm for two identical links. We first give a technical lemma.

LEMMA 3.1. *Given that the difference at time t is d_t, then by time $t+\ell+1$ algorithm* `BALANCE` *terminates with probability at least* $(\frac{1}{2})^\ell \sqrt{1/(2\pi(2.039 d_t)^{(\frac{1}{2})^\ell})}$.

Sketch of Proof: By applying ℓ times the relative Chernoff bound (Lemma 2.1), with $\delta = 1/2$ each time, we obtain that the difference at time $t+\ell$ is bounded by $(2.039 d_t)^{(\frac{1}{2})^\ell}$ with probability $(\frac{1}{2})^\ell$. Using Lemma 2.2, we obtain that the probability of terminating in the last stage is at least $\sqrt{1/(2\pi(2.039 d_t)^{(\frac{1}{2})^\ell})}$.

THEOREM 3.1. *With probability at least $1-\delta$ algorithm* `BALANCE`*, terminates within $O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$ stages.*

Proof. Let ℓ be an upper bound on the number of stages of algorithm `BALANCE` until reaching Nash equilibrium. (We latter fix ℓ and show that it is bounded with high probability.) We divide the execution to two phases, the first is while $d_t > 3\ln(1/\delta')$ and the second starts when $d_t \leq 3\ln(1/\delta')$, where $\delta' = \delta/(2\ell)$. We first show that the first phase ends with probability $1 - \delta/2$ within $O(\ln\ln(n))$ stages. By Lemma 2.1 with probability at least $1 - \delta'$, $d_{t+1} \leq \sqrt{3 d_t \ln(2/\delta')}$ for every stage $t \leq \ell$ in the first phase. Since d_0 is bounded by n, then $d_{t+1} \leq 3n^{1/2^{t+1}} \ln(1/\delta')$ with high probability. Therefore, after $O(\ln\ln(n))$ stages d_t is bounded by $3\ln(1/\delta')$ and once $d_t \leq 3\ln(1/\delta')$ it remains so with probability $1 - \delta/2$ until the end of stage ℓ by Lemma 2.1. For the second phase, we apply Lemma 3.1 with $\ell = O(\ln\ln\ln(1/\delta'))$, and obtain that algorithm `BALANCE` terminates with probability $O(\frac{1}{\ln\ln(1/\delta')})$ in $O(\ln\ln\ln(1/\delta'))$ stages. Therefore, if we make additional $O(\ln\ln(1/\delta')\ln\ln\ln(1/\delta')\ln(2/\delta))$ stages, we will terminate with probability at least $1 - \delta/2$. Solving the following equation

$$\ell = c_1 \ln\ln(n) + c_2 \ln(\ln(\ell/\delta)) \ln\ln(\ln(\ell/\delta)) \ln(2/\delta) + c_3,$$

we arrive at $\ell = O(\ln(1/\delta)\ln(\ln(1/\delta)\ln\ln(\ln(1/\delta)) + \ln\ln(n)) = O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$. □

A direct and simple corollary from the theorem is regarding the expected time of convergence.

COROLLARY 3.1. *Algorithm* `BALANCE` *terminates within expected $O(\ln\ln(n))$ stages.*

The following lemma shows that although algorithm `BALANCE` is not a Nash rerouting policy, the deviation is bounded by 1.

LEMMA 3.2. *At every stage t algorithm* `BALANCE` *is a* 1-*Nash rerouting policy.*

Proof. Consider any stage before reaching Nash equilibrium, where there are $n + d$ users on link 1 and $n - d$ on the link 2, i.e., $d_t = 2d$. User on the overloaded link reroutes with probability $\frac{d}{n+d}$. We compare the expected load on the links, excluding one user on the overloaded link.

$$\begin{aligned} L_1 &= (n + d - 1)(1 - \frac{d}{n+d}) = n - 1 + \frac{d}{n+d} \\ L_2 &= n - d + (n + d - 1)(\frac{d}{n+d}) = n - \frac{d}{n+d}. \end{aligned}$$

Therefore the difference is bounded by 1 and can be as large as $1 - \frac{2}{n+1}$. It is easy to see that the best response of users on the underloaded link is to remain on their current link. □

Next we present a modification of algorithm `BALANCE`, algorithm `NashTwo` which is a Nash rerouting policy in every stage, the algorithm appears at the top of the next page.

LEMMA 3.3. *At every stage Algorithm* `NashTwo` *is a Nash rerouting policy.*

Proof. As in Lemma 3.2 we compare the load on two links, one with load $n+d$ and the other with load $n-d$,

[3]This holds only for $p \geq \frac{1}{c' n}$ but we do not use smaller probabilities.

Input : Links: M_1, M_2, Users: $X = \{x_1, \dots, x_n\}$;
Let $t = 1$; $d_1 = |L_1(M_1) - L_1(M_2)|$;
$\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$;
while $d_t \geq 2$ **do**
  for x_i $s.t$ $g_t(x_i) = M_{\texttt{over}}$ **do in parallel**
    Move with probability
    $(d_t - 1)/(2L(M_{\texttt{over}}) - 2)$
  end
  $t = t + 1$; $d_t = |L_t(M_1) - L_t(M_2)|$;
  $\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$;
end

Algorithm 2: Distributed algorithm `NashTwo` for two identical links.

when excluding one user on the overloaded link. Since $d_t = 2d$, we have

$$L_1 = (n+d-1)(1 - \frac{2d-1}{2(n+d)-2}) = n - 1/2$$
$$L_2 = n - d + (n+d-1)(\frac{2d-1}{2(n+d)-2}) = n - 1/2.$$

Therefore, a greedy user on the overloaded link has no incentive to deviate from the joint strategy. As before, users from the underloaded link can only lose by trying to reroute to the overloaded link. □

Our next aim is to show that the convergence rate of `NashTwo` is similar to that of `BALANCE`. (The proof is similar to the proof of Theorem 3.1 and omitted.)

THEOREM 3.2. *Algorithm* `NashTwo` *is a Nash rerouting policy and with probability at least* $1 - \delta$*, reaches a Nash Equilibrium, within* $O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$ *stages. Also Algorithm* `NashTwo` *terminates within expected* $O(\ln\ln(n))$ *stages.*

3.2 Different Speed Links In this subsection we consider links with different speeds, specifically, $s_1 = 1$ and $s_2 = \alpha \geq 1$. The resulting Nash equilibrium is not necessarily unique. More precisely, any $n_t(1)$ is Nash equilibrium if and only if $n_t(1) \in [(n-\alpha)/(1+\alpha), (n+1)/(1+\alpha)]$. Recall that $\bar{L} = n/(1+\alpha)$.

We first derive the convergence rate of algorithm `BalanceSpeeds`(The proof is similar to the proof of Theorem 3.1 and omitted).

THEOREM 3.3. *Algorithm* `BalanceSpeed` *is a* 1*-Nash rerouting policy and with probability at least* $1 - \delta$ *it terminates within* $O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$ *stages.*

We first derive the convergence rate of algorithm `NashTwoSpeeds` (The proof is similar to the proof of Theorem 3.1 and omitted.).

Input : Links: M_1, M_2 with speeds $s_1 = 1$ and $s_2 = \alpha > 1$, Users: $X = \{x_1, \dots, x_n\}$;
Let $t = 1$; $\texttt{over} = \arg\max\{L_1(M_1), L_1(M_2)\}$;
$\texttt{under} = \arg\min\{L_1(M_1), L_1(M_2)\}$;
while $|L_t(M_1) - L_t(M_2)| > 1/s_{\texttt{under}}$ **do**
  for x_i $s.t$ $g_t(x_i) = M_{\texttt{over}}$ **do in parallel**
    Move with probability $p(n_t(\texttt{over}), s_{\texttt{over}})$
  end
  $t = t + 1$; $\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$,
  $\texttt{under} = \arg\min\{L_t(M_1), L_t(M_2)\}$;
end

Algorithm 3: Distributed algorithm for two links with speeds. `BalanceSpeeds` is derived by taking $p(n_t(\texttt{over}), s_{\texttt{over}}) = \frac{n_t(\texttt{over}) - s_{\texttt{over}}\bar{L}}{n_t(\texttt{over})}$ and `NashTwoSpeeds` is derived by taking $p(n_t(\texttt{over}), s_{\texttt{over}}) = \frac{d_t - 1/s_{\texttt{over}}}{(1+1/\alpha)(n_t(\texttt{over})-1)}$

THEOREM 3.4. *Algorithm* `NashTwoSpeeds` *is Nash rerouting policy and with probability at least* $1 - \delta$ *it terminates within* $O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$ *stages.*

3.3 Multiple Weights In this subsection we consider a slightly different model and assume that each user has a weight and as a consequence in a Nash rerouting policy only users with same weight on a link must share a policy. We show that there exists a set of weights and an initial assignment such that any Nash rerouting policy converges within $\Omega(\sqrt{n})$ even for two links. [4] We consider n users, where n is a square of an even number. Each user belongs to one of the $\sqrt{n}$ weights classes, where the weight of user in the k class is n^{2k} and the size of each class is $\sqrt{n}$. We say that class i is larger than class j if $i > j$, i.e. the weight of a user in class i is larger than the weight of those in class j. A Nash rerouting policy is fully described by $p_1, \dots, p_{\sqrt{n}}$, where p_i is the probability of user from class i to migrate from the overloaded link to the underloaded link.

It is straightforward to see that the only Nash equilibrium of this game is that each class is partitioned equally on the links. We say that a class i is balanced only if it is partitioned equally and every class j such that $j > i$ is balanced. We also note that after a class is balanced, its users will not move under any Nash rerouting policy.

The next lemma shows that if a user from the largest unbalanced class is on the underloaded link, then only the other members of her class move stochastically in the system.

[4] This result can be extended to a boarder class of policies.

LEMMA 3.4. *Let ℓ be the largest unbalanced class, then if there exists a class $k < \ell$ such that $p_k \in (0,1)$ then all users of class ℓ are on the overloaded link.*

Proof. Suppose by contradiction that there exists a configuration g in which both users from class ℓ and class k move stochastically and that there is at least one user from class ℓ on the underloaded link. Let $L_u(L_o)$ be the expected load after the users move on the underloaded(overloaded) link. Since the policy is Nash rerouting, the following must be satisfied

$$\begin{aligned} L_u - p_\ell w_\ell &= L_o - (1-p_\ell)w_\ell \\ L_u - p_k w_k &= L_o - (1-p_k)w_k. \end{aligned}$$

Simplifying these equations we obtain that $(1-2p_\ell)w_\ell = (1-2p_k)w_k$. Together with the fact that in a Nash rerouting policy users in the largest unbalanced class move with probability at most $1/2$, we obtain that users in every other unbalanced class that move stochastically move with probability at most $1/2$. Writing $p_i = 1/2 - \epsilon_i$ for some $\epsilon_i > 0$, we have $\epsilon_\ell w_\ell = \epsilon_k w_k$. Since $w_k/w_\ell \le 1/n^2$ we have that $\epsilon_\ell/\epsilon_k < 1/n^2$, which implies $p_\ell > 1/2 - 1/n^2$. By our assumption all user with weight larger than w_ℓ are already balanced. If X is the current number of users from class ℓ on the overloaded link, then the expected load after move due to class ℓ on the overloaded link is at most $X(1/2+1/n^2)w_\ell$ while the expected load on the underloaded link is at least $(X(1/2-1/n^2)+1)w_\ell$ (using the contradiction assumption). Since the expected load of all users from classes $1,\ldots,\ell-1$ is bounded w_ℓ/n, the difference between L_u and L_o is at least $w_\ell/2$ and users from class k will move deterministically to the overloaded link, a contradiction to the lemma assumption. □

Next, we complete the characterization of the Nash rerouting policy.

LEMMA 3.5. *Let ℓ be the largest unbalanced class, if there are users from class ℓ on the underloaded link, then all users from smaller classes move to the underloaded link deterministically.*

Proof. We first note that by Lemma 3.4 only users of class ℓ move stochastically. We denote $k = \sqrt{n}/2$, and assume that on the overloaded link that are $k+d$ users from class ℓ and on the underloaded link there are $k-d$ users, where $1 \le d \le k-1$. For any policy we let w_u and w_o be the expected weight of classes $1,\ldots,\ell-1$ on the underloaded and overloaded links. Hence, $p_\ell = \frac{d-1/2}{k+d-1} + \frac{w_o - w_u}{(k+d-1)n^{2\ell}}$, together with the fact that the load of all users from classes $1,\ldots,\ell-1$ is bounded by $n^{2\ell}/n$, we obtain that

$$\frac{1}{3\sqrt{n}} < \frac{1}{2k} - \frac{1}{n} \le p_\ell \le \frac{k-1.5}{2k-2} + \frac{1}{n} < \frac{1}{2} - \frac{1}{3\sqrt{n}}.$$

This assures that the expected load of the overloaded link in the next stage is at least larger by $O(n^{2\ell-1/2})$ from the expected load on the underloaded link, thus all users of classes $1,\ldots,\ell-1$ will move deterministically to the underloaded link. □

In the following proof we use the previous lemma to show that the classes are balanced sequentially.

THEOREM 3.5. *There exists a set of weights and an initial assignment such that every Nash rerouting policy converges in $\Omega(\sqrt{n})$ steps with probability at least $1 - O(1/n^{1/4})$.*

Proof. The initial configuration is that all users are on one link. We let s_i be the configuration in which all users from the unbalanced classes $1,\ldots,i$ are on one link and classes $i+1,\ldots,\sqrt{n}$ are balanced and let $S = \{s_1,\ldots,s_{\sqrt{n}}\}$ be the set of these configurations. Although most of the time the configuration is not expected to be inside the set S, we show that under any Nash rerouting policy the configuration is in S for $\Omega(\sqrt{n})$ times with high probability. We note that the in state s_i in any Nash rerouting policy, all users from classes $1,\ldots,i$ move with probability $1/2$ and users from classes $i+1,\ldots,\sqrt{n}$ stay deterministically.

Each time that the system is in $s_i \in S$,$(i \ge 3)$, the probability that the largest three classes are balanced at once is at most $O\left((\frac{1}{n^{1/4}})^3\right)$. The probability that we do not balance the three largest classes at once is at least $1 - O(1/n^{1/4})$ in first $\sqrt{n}$ times we are in S. Therefore, by Lemma 3.5 we reach s_i for $i \in [\ell-4, \ell-1]$. Thus, we reach at least $\Omega(\sqrt{n})$ configurations in the set S with probability at least $1 - O(1/n^{1/4})$. □

4 Multiple Links

4.1 Multiple identical links, $n = m$ In this subsection we consider the case where the number of users is equal to the number of links and all links are identical. The algorithm that we analyze this section is different from other algorithms in this paper, in the sense that a user can reroute from an overloaded link to a different overloaded link. Algorithm `RandomJump` works as follows: in each stage a user from an overloaded link assigns equal probability to reroute to any unbalanced link (overloaded or underloaded). This algorithm was first given in [1], later it was demonstrated to converge empirically in logarithmic time [10] but only a linear upper bound was given there. We show a logarithmic convergence bound.

THEOREM 4.1. *For $n = m$ algorithm* `RandomJump` *terminates within $O(\ln(n) + \ln(1/\delta))$ with probability at least $1 - \delta$.*

Proof. We note that for this special case an underloaded link is an empty link and a stable link is a link with one user on it. Therefore, every user on an overloaded link with load $d+1$ reroutes with probability $\frac{d}{d+1}$ and the target link is chosen randomly. We first show that the probability that an underloaded link becomes balanced in one step is $\frac{1}{e}$. Suppose that in the kth iteration there are τ underloaded links, then there $\tau+\ell$ users in ℓ overloaded links. We let E be the event that an underloaded link becomes stable, i.e. only one user jumps to it. This probability can be lower bounded as follows,

$$\begin{aligned}\mathbf{P}(E) &= \binom{\ell+\tau}{1}(1-\frac{1}{\ell+\tau})^{\ell+\tau-1}\frac{1}{\ell+\tau} \\ &= (1-\frac{1}{\ell+\tau})^{\ell+\tau-1} \geq \frac{1}{e}\end{aligned}$$

Our second argument is that each link becomes either stable or underloaded with probability $\frac{1}{2}$ in the next iteration. This is true due to the fact that for each link we have a binomial distribution, where its mean is the stable point. Therefore, the probability that a link becomes stable in two iterations is at least $\frac{1}{2e}$. Using the union bound we obtain that after $2e(\ln(m)+\ln(1/\delta))$ all links are stable with probability at least $1-\delta$. □

4.2 Multiple Links In this subsection we consider multiple links of different speeds. We assume that the speeds are sorted, i.e., $s_1 = 1 \leq s_2 \leq \cdots \leq s_m$. We derive a variant of the `Balance` algorithm for the case of multiple links with different speeds. For simplicity we prove the theorems only for identical links, however the theorem holds for links with different speeds as well.

We first note that algorithm `BalanceMultipleSpeed` is a 1-Nash rerouting policy. (proof omitted.)

LEMMA 4.1. *Algorithm* `BalanceMultipleSpeed` *is a* 1-*Nash rerouting policy.*

Next we characterize a class of policies, which converge fast.

DEFINITION 4.1. *A policy is* `fast convergent` *if, for some constant $c > 0$, it has the following properties:*

1. *For any configuration, g, link M, time t and $\delta < 1$, if $d_t = |L(M) - \bar{L}| \geq c\ln(1/\delta)$, then $d_{t+1} \leq \sqrt{cd_t \ln(1/\delta')}$ with probability at least $1-\delta$.*
2. *For any configuration, g, link M, time t and $\delta < 1$, if $d_t \leq c\ln(1/\delta)$, then $d_{t+1} \leq c\ln(1/\delta)$ with probability at least $1-\delta$.*
3. *For any configuration g, and link M, if $d = |L(M) - \bar{L}|$, then M is balanced in on stage with probability at least $O(1/\sqrt{d})$.*

Input : Links: $M_1, \ldots, M_m$, users $X = \{x_1, \ldots, x_n\}$
$t = 1$; For every $i \in \{1, \ldots, m\}$ let
$d_t(i) = n_t(i) - s_i\bar{L}$;
$\mathtt{OVER}_t = \{i | d_t(i) > 0\}$; $\mathtt{UNDER}_t = \{i | d_t(i) < 0\}$;
$d(t) = \sum_{i \in \mathtt{OVER}_t} d_t(i)$;
while $\exists i, j : L_t(M_i) - L_t(M_j) > 1/s_j$ **do**
  for $g_t(x_i) \in \mathtt{OVER}_t$ **do in parallel**
    Move with probability $d_t(g_t(x_i))/n_t(g_t(x_i))$
    if x_i *moves* **then**
      Move to link $j \in \mathtt{UNDER}_t$ with probability $\frac{|d_t(j)|}{d(t)}$
    end
  end
  $t = t + 1$; For every $i \in \{1, \ldots, m\}$ let
  $d_t(i) = L_t(M_i) - s_i\bar{L}$;
  $\mathtt{OVER}_t = \{i | d_t(i) > 0\}$; $\mathtt{UNDER}_t = \{i | d_t(i) < 0\}$
  and $d(t) = \sum_{i \in \mathtt{OVER}_t} d_t(i)$;
end

Algorithm 4: Distributed algorithm `BalanceMultipleSpeed` for multiple links

Next we show that given a link under any fast convergent policy, the link is balanced with high probability in $\tilde{O}(\ln\ln(n) + \ln(1/\delta))$ stages. (The proof is similar to the proof of Theorem 3.1 and omitted.)

LEMMA 4.2. *Given a link M, and any initial configuration, M is balanced after $O(\ln\ln(n)) + \tilde{O}(\ln(1/\delta))$ stages with probability at least $1-\delta$ under any fast convergent policy.*

The next proposition shows that a fast convergent policy indeed converges fast to Nash equilibrium, later we show that both `BalanceMultipleSpeed` and Nash rerouting policy are fast convergent policies and enjoy rapid convergence rate.

PROPOSITION 4.1. *If in every stage a fast convergent policy is used, the all links are balanced after $O(\ln\ln(n)) + \tilde{O}(\ln(m/\delta))$ stages with probability at least $1-\delta$.*

Proof. By Lemma 4.2, each link is balanced with probability at least $1-\frac{\delta}{m}$ in $O(\ln\ln(n)) + \tilde{O}(\ln(m/\delta))$ stages. Taking the union bound we obtain that all links are balanced in $\tilde{O}(\ln\ln(n) + \ln(m/\delta))$ stages with probability at least $1-\delta$. □

The next lemma shows that Algorithm `BalanceMultipleSpeed` produces a fast convergent policy.

LEMMA 4.3. *The policy generated by Algorithm* `BalanceMultipleSpeed` *is a fast convergent policy.*

Proof. Given a link M at time t, for any configuration g we let $d_t = L_t(M) - L$. We show that the policy generated by Algorithm `BalanceMultipleSpeed` satisfies all three properties of the fast convergent policy class. We begin by showing that if $|d_t| \geq 3\ln(1/\delta)$, then $|d_{t+1}| \leq \sqrt{3|d_t|\ln(1/\delta)}$. When $d_t \geq 3\ln(1/\delta)$ we apply Lemma 2.1 with $n/m + d_t$ i.i.d random variables, $Z_1, \ldots, Z_{n/m+d_t}$, where $\sum_{i=1}^{n/m+d_t} E[Z_i] = \frac{d_t}{n/m+d_t}$, thus $d_{t+1} \leq \sqrt{3d_t \ln(1/\delta)}$ with probability at least $1-\delta$. When $d_t \leq -3\ln(1/\delta)$ we apply the relative Chernoff bound with K independent random variables, $Z_1, \ldots, Z_K$, where $n/m + d_t < K < n - n/m + d_t$ and $\sum_{i=1}^{K} E[Z_i] = d_t$, to obtain that $|d_{t+1}| \leq \sqrt{3|d_t|\ln(1/\delta)}$. Showing that if $|d_t| \leq 3\ln(1/\delta)$, then $|d_{t+1}| \leq 3\ln(1/\delta)$, is done similarly using the relative Chernoff bound. By Lemma 2.2 the probability of terminating in one stage (for both underloaded and overloaded links) is at least $O(\sqrt{1/|d_t|})$. □

By Lemma 4.3 and Proposition 4.1 we derive the following theorem.

THEOREM 4.2. *With probability at least $1-\delta$ algorithm* `BalanceMultipleSpeed` *terminates within $O(\ln\ln(n)) + \tilde{O}(\ln(m/\delta))$ stages, and it terminates in expected $O(\ln\ln(n) + \ln(m))$ stages.*

4.3 Nash Rerouting policies for Multiple Links

We define a very natural class of policies, where users reroute only from overloaded links and only to underloaded links. In addition we require that the deviation from a Nash rerouting policy would not be too large. This set clearly includes the various `BALANCE` policies. We will show that this set also includes a Nash rerouting policy. In addition, any rerouting, which at every stage uses some policy from this set, converges fast.

DEFINITION 4.2. *For any $g \in G$ let* `OVER`(`UNDER`) *be the set of overloaded (underloaded) links in g. Let $\Pi(g)$ be the set of user policies satisfying the following properties for $P \in \Pi(g)$:*

1. *Every user on a link in* `OVER` *does not move to a different link in* `OVER`*. I.e., for every i such that $g(i), j \in$ `OVER` and $g(i) \neq j$ we have $p_i(j) = 0$.*

2. *Every user on a link in* `UNDER` *does not move to any other link. I.e., for every $i \in$ `UNDER` we have $p_i(g(i)) = 1$.*

3. *The expected load of every link M_i, with respect to P, is in the interval $(\bar{L} - 1/s_i, \bar{L} + 1/s_i)$*

4. *All the users on a link share the same policy. I.e., for every M_i for all users j and k such that $g(j) = i = g(k)$ we have $P_j = P_k$.*

The lemma next relates the class $\Pi(g)$ to the fast convergent policies class. (The proof is similar to the proof of Lemma 4.3 and omitted.)

LEMMA 4.4. *If for every configuration, g, a policy in $\Pi(g)$ is used then the resulting policy is a fast convergent policy.*

The following theorem relates $\Pi(g)$ to the convergence rate and its proof follows immediately from Lemma 4.4 and Proposition 4.1.

THEOREM 4.3. *If during every stage of the rerouting some policy in $P^t \in \Pi(g_t)$ is used, then the system converges to Nash equilibrium, with probability at least $1-\delta$, within $O(\ln\ln(n)) + \tilde{O}(\ln(m/\delta))$ stages.*

Next we show that for any configuration $g \in G$, the set $\Pi(g)$ contains a Nash rerouting policy for g. (The proof, in Appendix A, is similar to the proof that a symmetric game has a symmetric Nash equilibrium, and is based on Brouwer fixed point theorem.)

THEOREM 4.4. *For every configuration $g \in G$, there exists a Nash rerouting policy in $\Pi(g)$.*

Theorem 4.3 and Theorem 4.4 imply that there exists a Nash rerouting policy that converges in expected $O(\ln\ln n + \ln m)$ stages to Nash equilibrium.

The next observation is regarding the Nash rerouting policies structure.A two phases rerouting policy is a policy that behaves as follows: every user from an overloaded link k leaves link k with probability p_k. Any user that leaves an overloaded link, moves to an underloaded link j with probability q_j. (Note that the various `Balance` algorithms are a two phase rerouting policy.) We show that for some configurations no two phase rerouting policy is a Nash rerouting policy, (proof is in appendix A).

LEMMA 4.5. *There exists a configuration $g \in G$, in which no two phases rerouting policy is a Nash rerouting policy.*

Next we present an algorithm, which computes Nash rerouting policy. The algorithm is based on the fact that given the support of each user, we can compute the Nash rerouting policy by solving set of linear equations. Thus by enumerating all possible supports of all users, we can compute a Nash rerouting policy.

THEOREM 4.5. *Computing a Nash rerouting policy can be done in $O(2^{m^2})$.*

References

[1] S. Alpern and D. Reyniers, "Spatial Dispersion as a Dynamic Coordination Problem", *Theory & Decision* , 53:1, pp. 29-59, 2002.

[2] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155-193, 1979.

[3] B. Awerbuh, Y. Azar, and Y. Richter, "Analysis of worse case Nash equilibria for restricted assignment", WOWA 2003.

[4] L. E. J. Brouwer. Uber abbildung von mannigfaltigkeiten. *Mathematische Annalen*, 71:97-115, 1912.

[5] A. Czumaj, P. Krysta and B. Vocking, "Selfish traffic allocation for server farms," *in Proceedings of the 34th Symposium on Theory of Computing*,pp. 287-296, 2002.

[6] A. Czumaj and B. Vocking, "Tight bounds on worse case equilibria", *In Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 413-420, 2002.

[7] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence, *Random Structures and Algorithms*, 13:2, pp. 99-124,1998.

[8] E. Even-Dar, A. Kesselman and Y. Mansour, "Convergence Time To Nash Equilibria," *In Proceedings of the 30th ICALP*, pp. 502-513, 2003.

[9] Paul Goldberg, "Bounds for the convergence rate of randomized local search in multiplayer games, uniform resource sharing game", *In Proceedings of the Twenty-Third PODC*, pp. 131-140, 2004.

[10] T. Grenager, R. Powers and Y. Shoham, "Dispersion games: general definitions and some specific learning results", *In Eighteenth national conference on Artificial intelligence*, pp 398-403, 2002.

[11] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis, "The Structure and Complexity of Nash Equilibria for a Selfish Routing Game," *In Proceedings of the 29th ICALP*, Malaga, pp 123-134, 2002.

[12] E. Koutsoupias, C. H. Papadimitriou, "Worst-case equilibria,"*in Proceedings of 16th STACS*, pp. 404–413, 1999.

[13] L. Libman and A. Orda, "Atomic Resource Sharing in Noncooperative Networks ", *Telecommunication Systems*, 17:4, pp 385-409, 2001.

[14] P. Marbach "Priority service and max-min fairness", *IEEE/ACM Trans. Netw.*, 11:5, pp 733-746, 2003.

[15] M. Mitzenmacher, A. Richa and R. Sitaraman, "The power of two random choices: A survey of the techniques and results". In *Handbook of Randomzied Computing*, Rajasekaran et al., Kluuwer Academic Press, 2000.

[16] V. Mirrokni and A. Vetta, "Convergence issues in competitive games", In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization*, 2004.

[17] J. F. Nash, "Non-cooperative games," *Annals of Mathematics*, Vol. 54, pp. 286-295, 1951.

[18] A. Orda, R. Rom and N. Shimkin, "Competitive routing in multi-user communication networks," *IEEE/ACM Transaction on Networking*, Vol 1, pp. 614-627, 1993.

[19] T. Roughgarden and E, Tardos, "How Bad is Selfish Routing?," *Journal of the ACM*, 49(2):236–259, 2002.

A Proofs from Subsection 4.3

We first give characterization of every Nash equilibria which will be useful in the proof of Theorem 4.4.

Lemma A.1. *In every Nash equilibrium, in which all users on the same link share the same policy, if a user from an overloaded link M_i reroutes with positive probability to an underloaded link M_j at time t, then $E[L_{t+1}(M_i)] \geq E[L_t(M_j)]$*

Proof. Since the system is in Nash equilibrium, then the expected load on the links excluding one user on M_i is identical. Suppose by contradiction that $E[L_{t+1}(M_i)] < E[L_t(M_j)]$, thus $p_i^t(j) > p_i^t(i)$ but this implies that load on M_j is larger than the load on M_i excluding the last user, which is contradiction. □

Proof of Theorem 4.4 : We first note that any Nash equilibrium must satisfy the third condition. By restricting users actions in the game we can obtain the first and second condition. Therefore, it remains to prove that there exists an equilibrium in which all the users on a link share the same policy. We prove it by using Brouwer Fixed Point Theorem [4]. (The proof is almost identical to the Nash equilibrium existence proof.) Let Δ be the set of all possible policies, i.e. if $P \in \Delta$ then $P_{i,j}$ is the probability of the user i to move from its link $g(i)$ to link j and $\sum_j P_{i,j} = 1$. For every policy, P we define the policy $\bar{P}_{i,j} = \frac{\sum_{i':g(i)=g(i')} P_{i',j}}{\sum_{i':g(i)=g(i')} \mathbf{1}_{P_{i',j}}}$, where $\mathbf{1}_{P_{i',j}}$ is 1 if $P_{i',j} > 0$ and 0 otherwise. Intuitively, $\bar{P}$ averages the policies of users that share the same link, it remains to show that $\bar{P}$ can be an equilibrium. We define the utility function of the distribution $\bar{P}$ as,

$$u_i(\bar{P}) = E_{j \backsim \bar{P}_{i,j}} E_{\bar{P}_{-i}}[L(M_j)] + 1/s_j$$

Next we define the utility of switching deterministically to link j.

$$Z_{i,j} = \max\{u_i(j, \bar{P}_{-i}) - u_i(\bar{P}), 0\}$$

After defining these rewards define the mapping $y : \Delta \longmapsto \Delta$, with i, j^{th} component

$$y_{i,j} = \frac{p_{i,j} + Z_{i,j}}{1 + \sum_j Z_{i,j}}$$

Note that this mapping is continuous and Δ is a compact and convex set. Therefore, there exists a fixed point for this mapping, we would like to show that there exists at least one fixed point which is an equilibrium and all the users on the same link share the same policy. We show that if P is an equilibrium then $\bar{P}$ is an equilibrium. If P is an equilibrium then for any link j such that $P_{i,j} > 0$ the move from $g(i)$ to j is best response with respect to $\bar{P}$. Thus every distribution on these link is a best response including $\bar{P}$. It remains to show that an equilibrium in the restricted game is an equilibrium in the original game as well. By Lemma A.1, if a user on an overloaded link would like to reroute to a different overloaded link, then there is also an underloaded link with less or equal load that he would like to move to, thus the system was not in equilibrium in the restricted game. If a user on an underloaded link would like to reroute then there exists a user on an overloaded link that would like to reroute to that link as well, which is contradiction to the the fact that it is an equilibrium in the restricted game.

Proof of Lemma 4.5 : We consider a case where there are four links; Two overloaded links M_1, M_2 with load $n + d_1, n + d_2$ and two underloaded links M_3, M_4 with load $n - d_3, n - d_4$. We let $p_1(p_2)$ be the probability leaving $M_1(M_2)$ and $q_3(q_4)$ be the probabilities of reaching $M_3(M_4)$ if a user moves. Clearly all probabilities are strictly positive. A Nash equilibrium must satisfy the following equations:

$$\begin{aligned}
(n+d_1-1)(1-p_1) &= n-d_3+p_1(n+d_1-1)q_3 \\
&\quad +p_2(n+d_2)q_3 \\
(n+d_1-1)(1-p_1) &= n-d_4+p_1(n+d_1-1)q_4 \\
&\quad +p_2(n+d_2)q_4 \\
(n+d_2-1)(1-p_2) &= n-d_3+p_1(n+d_1)q_3 \\
&\quad +p_2(n+d_2-1)q_3 \\
(n+d_2-1)(1-p_2) &= n-d_3+p_2(n+d_1)q_4 \\
&\quad +p_2(n+d_2-1)q_4
\end{aligned}$$

By subtracting Eq.1 form Eq.2 and Eq.3 from Eq.4 we obtain,

$$d_4 - d_3 = p_1(n+d_1-1)(q_4-q_3) + p_2(n+d_2)(q_4-q_3)$$

$$d_4 - d_3 = p_1(n+d_1)(q_4-q_3) + p_2(n+d_2-1)(q_4-q_3).$$

Therefore, $p_1(q_4 - q_3) = p_2(q_4 - q_3)$ which implies that either $p_1 = p_2$ or $q_3 = q_4$. By taking $d_1 = n, d_2 = 1, d_3 = 1, d_4 = n$, we show that both conditions cannot be satisfied. If $q_3 = q_4$ then the load on M_4 would be strictly smaller which is contradiction. If $p_1 = p_2$, then either the load on M_1 is greater then $n+2$ or the load on M_2 is smaller than $n-1$. In the first case a user on M_1 would leave with probability 1 and in the second case a user on M_2 would stay with probability 1.