Fast Convergence of Selfish Rerouting

Eyal Even-Dar * Yishay Mansour †

April 7, 2004

Abstract

We consider n identical selfish users that route their communication using 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 a simple migration policy, which balances the expected load on the links, and show that if all the users apply it, the rerouting terminates in expected $\tilde{O}(\log\log n + \log^{1.5} m)$ stages. We extend our study by considering Nash rerouting policies, where *every* rerouting phase is a Nash equilibrium, and show similar termination bounds in this setting. We study the structural properties of such Nash rerouting policies, and derive both existence result and an efficient algorithm for the case that the number of links is small.

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.

One can take a different perspective of the routing problem, allowing individual users to specify the path on which their traffic would be routed. In such a setting each individual user optimizes its own utility function, and naturally tries to route its traffic on the least loaded path. An appropriate solution concept for such system is that of Nash equilibrium [16], where no user can benefit from rerouting its traffic. While this distributed view of network routing has many interesting features, it does carry with it the difficulty that the selfish users decisions might negatively impact the overall performance of the system.

Due to the high attractively of a decentralize distributed routing, networks researchers have looked for over a decade into the issues involved in the design and implementation of such a 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, 14, 17].

*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. Supported in part by a grant from ISF and IBM Faculty Award.

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, 18].

While Koutsoupias and Papadimitriou [12] focus on a one stage game, we focus on a dynamic model that can be represented as a repeated game. Therefore, if an equilibrium point is not ex-post Nash[1], users will try to change their policies in the next stage. Our major motivation is to study "how fast Nash equilibrium is reached" rather than "the quality of the Nash equilibrium". Previous works used the Elementary Step System, where only one user can reroute at each stage, to study the dynamics of convergence to a Nash equilibrium [8, 9, 13, 17].

While the Elementary Step System has attractive theoretical properties, it has also very clear drawbacks. First, since only one user is allowed to reroute, 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 of the realistic issues that are involved in the decision to reroute "disappear", such as the fear of a user that after the rerouting it 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 it shares the link with, and therefore the user 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 its traffic to a different link. We assume that each user has access only to the load on the links, which implies that all the 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 $O(\log \log n + \log^{1.5} m)$. 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 the other with 400. The `BALANCE` policy reroutes from the overloaded link 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 expected load on the overloaded link is $300 - 3/4 = 299.25$. Being selfish, 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 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.

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

We study a very natural class of Nash rerouting policies that has the following properties: (1) all the users that share a link have the same policy, (2) no user on underloaded link reroutes, and (3) no user on an overloaded link reroutes to a different overloaded link. We show that if at every stage the such Nash rerouting policy is used, then the expected number of stages is $O(\log\log n + \log^{1.5} m)$. 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 are 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 the user 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 has been interested in various aspects of selfish rerouting for over a decade. In [17] 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 dynamic 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] studies the convergence to Nash equilibrium under the ESS model, for several settings: identical, related and unrelated machines and jobs with identical/different sizes. For each setting different lower and upper bounds on the convergence rate were given. For example, in the case of identical machines but different job sizes, the convergence might take exponential time, under one natural selection policy, and linear time using a different selection policy.

The work of [9] studies a randomized model where each user can select a random delay over continuous time. The continuous time implies that only one user tries to reroute at each specific time, and for this reason the model is very similar to the ESS model. In that model rerouting succeeds if the user lowers its load and otherwise fails (in which case the user does not reroute). The work shows a simple 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.

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 our 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 some what 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 and does not change its assignment. The goal is to minimize the maximum load. In this work we are interest 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 users to follow the prescribed policy.

2 Model Description

Our model is composed from a set of n identical users $x_1, \ldots, x_n$, and a set of m parallel links $M_1, \ldots, M_m$ with speeds $s_1, \ldots s_m$, 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 $L_t(M_k) = n_t(k)/s_k$.

The average load in the system is given by $\bar{L} = n / \sum_{i=1}^{m} s_i$, for identical links this 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 $x_1, \ldots, x_n$ is a product distribution $P_1^t \cdots P_n^t$, where $P_i^t =< p_i^t(1) \ldots p_i^t(m) >$ is a distribution on $M_1, \ldots, M_m$. 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 it 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 loaded links reroute to underloaded links, and we show that there exists a Nash rerouting policy that has this property.

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 be defining the final state equilibrium.

Definition 2.1 *A system is in* Nash-equilibrium *in 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* Nash rerouting policy *at time t if no user can benefit (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 $j \in \arg\min_k E_{P_{-i}^t}[L_{t+1}(M_k)]$.*

*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 $E_{P_{-i}^t}[L_{t+1}(M_j)] \leq \epsilon + \arg\min_k E_{P_{-i}^t}[L_{t+1}(M_k)]$.*

Our convergence proofs will use extensively large deviation bounds, specifically the relative Chernoff bound [2], as stated in Lemma A.1 (Appendix A). Another bound that is used through the paper is a bound on the probability of hitting the mean.

Lemma 2.1 *Let $Z_1, \ldots, Z_n$ be i.i.d $\{0,1\}$-random variables with $\mathbf{P}(Z_i = 1) = p$ and $Z = \sum_{i=1}^{n} Z_i$. Then $\mathbf{P}(Z = \lceil pn \rceil) \geq \frac{1}{e} \frac{1}{\sqrt{2\pi \lceil pn \rceil}}$. If $pn = q$ is an integer then, $\mathbf{P}(Z = q) \geq \frac{1}{\sqrt{2\pi q}}$*

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

Input : Links: M_1, M_2, Users: $X = \{x_1, \ldots, x_n\}$;
Let $d_1 = |L_t(M_1) - L_t(M_2)|$; $t = 1$;
$\texttt{over} = \arg\max\{L_t(M_1), L_t(M_2)\}$;
while $d_t \geq 2$ **do**
  Every 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 links

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 algorithm* `BALANCE` *terminates with probability* $(\frac{1}{2})^\ell \sqrt{1/(2\pi(2.039 d_t)^{(\frac{1}{2})^\ell})}$ *by time* $t + \ell + 1$.

Sketch of Proof: By applying ℓ times the relative Chernoff bound (Lemma A.1), with $\delta = 1/2$ each time, we obtain that the difference in time $t + \ell$ is bounded by $(2.039 d_t)^{(\frac{1}{2})^\ell}$ with probability $(\frac{1}{2})^\ell$. Using Lemma 2.1, 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 3.2 *With probability at least* $1 - \delta$ *algorithm* `BALANCE`*, terminates within* $\tilde{O}(\ln\ln(n) + \ln(1/\delta))$ *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 define two phases of the execution, 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 A.1 with probability at least $1 - \delta'/2$, $d_{t+1} \leq \sqrt{3 d_t \ln(1/\delta')}$ for every stage $t \leq \ell$. Since d_0 is bounded by n, then $d_{t+1} \leq 3 n^{1/2^{t+1}} \ln(1/\delta')$. 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 high probability until the end of stage ℓ. 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 finish with probability at least $1 - \delta/2$. Solving the following equation

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

Input : Links: M_1, M_2, Users $X = \{x_1, \ldots, x_n\}$;
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**
  Every 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 links

we obtain that $\ell = O\left(\ln\ln(n) + \ln(1/\delta)\ln(\ln(1/\delta))\ln\ln(\ln(1/\delta))\right) = \tilde{O}\left(\ln\ln(n) + \ln(1/\delta)\right)$. ∎

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

Corollary 3.3 *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 3.4 *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 one link and $n - d$ on the other, 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 which implies that the user gain is bounded by 1 as well. For users on the underloaded link their best response 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.

Lemma 3.5 *At every stage Algorithm* `NashTwo` *is a Nash rerouting policy.*

Proof: As in Lemma 3.4 we compare the load on two links, one with load $n + d$ and the other with load $n - d$, when excluding one user on the overloaded link. Since $d_t = 2d$, we have

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

Input : Links: M_1, M_2 with speeds $s_1 = 1$ and $s_2 = \alpha > 1$, Users: $X = \{x_1, \ldots, x_n\}$;
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**
  Every x_i *s.t* $g_t(x_i) = M_{\texttt{over}}$ **do in parallel**
    Move with probability $\frac{n_t(\texttt{over}) - s_{\texttt{over}}\bar{L}}{n_t(\texttt{over})}$
  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 `BalanceSpeeds` for two links

Therefore, each user on the overloaded link cannot gain by deviating 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 3.2 and is omitted.)

Theorem 3.6 *Algorithm* `NashTwo` *is a Nash rerouting policy and with probability at least* $1 - \delta$*, reaches a Nash Equilibrium, within* $\tilde{O}(\ln\ln(n) + \ln(1/\delta))$ *stages. Also Algorithm* `NashTwo` *terminates within expected* $O(\ln\ln(n))$ *stages.*

3.2 Different Speed Links

In this section we analyze the case where each link has a different speed, 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 present the algorithm.

Our first aim is to give a convergence rate for Algorithm `BalanceSpeeds`, which is similar to that of Theorem 3.2 (proof omitted).

Theorem 3.7 *Algorithm* `BalanceSpeed` *is a* 1*-Nash rerouting policy and with probability at least* $1 - \delta$ *it terminates within* $\tilde{O}\left(\ln\ln(n) + \ln(1/\delta)\right)$ *stages.*

Our next aim is to present Algorithm `NashTwoSpeeds`, which is a Nash rerouting policy, and show its convergence rate (proof omitted).

Theorem 3.8 *Algorithm* `NashTwoSpeeds` *is Nash rerouting policy and with probability at least* $1 - \delta$ *it terminates within* $\tilde{O}\left(\ln\ln(n) + \ln(1/\delta)\right)$ *stages.*

4 Multiple Links

4.1 Multiple identical links, $n = m$

In this subsection we analyze the case where the number of users equals 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.

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

Algorithm 4: Distributed algorithm `NashTwoSpeeds` for two links with two speeds

Algorithm `RandomJump` works as follows: in each stage a user on 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. (The proof is in Appendix B)

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

4.2 Multiple Links

In this subsection we consider the case that we have 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.

Lemma 4.2 *Algorithm* `BalanceMultipleSpeed` *is a 1-Nash rerouting policy.*

Next we derive our main result, which is a convergence rate for algorithm `BalanceMultipleSpeed`.

Theorem 4.3 *With probability at least $1 - \delta$ algorithm* `BalanceMultipleSpeed` *terminates within $\tilde{O}(\ln\ln(n) + \ln^{1.5}(m) + \ln^{1.5}(1/\delta))$ stages.*

Proof: For simplicity, the proof is given only for identical links. Let ℓ be an upper bound on the number of stages of algorithm `BalanceMultipleSpeed`, we later show that it is bounded with high probability. Since the speeds are identical ($s_i = 1$) we have that $\bar{L} = n/m$ and $d_t(k) = L_t(M_k) - \frac{n}{m}$. Let $over_t = |\texttt{OVER}_t|$, $under_t = |\texttt{UNDER}_t|$, and $unbalanced_t = over_t + under_t$. We divide the convergence to two phases, the first is while there exists a link k such that $d_t(k) > 3\ln(1/\delta')$ and the second starts at the first time t such that for every link k we have $d_t(k) \leq 3\ln(1/\delta')$, where $\delta' = \delta/(3m\ell)$. Since for any link k and time t, with probability $1 - \delta'$ we have that $d_{t+1}(k) \leq \sqrt{3d_t(k)\ln(1/\delta')}$, the first phase ends in $O(\ln\ln(n))$ stages, and during all the second phase we have $d_t(k) \leq 3\ln(1/\delta')$.

Input : Links: $M_1, \ldots, M_m$, users $X = \{x_1, \ldots, x_n\}$
$t = 1$; For every $i \in \{1, \ldots, m\}$ let $d_t(i) = n_t(i) - s_i \bar{L}$;
$\texttt{OVER}_t = \{i | d_t(i) > 0\}$; $\texttt{UNDER}_t = \{i | d_t(i) < 0\}$; $d(t) = \sum_{i \in \texttt{OVER}_t} d_t(i)$;
while $\exists i, j : L_t(M_i) - L_t(M_j) > 1/s_j$ **do**
  Every $g_t(x_i) \in \texttt{OVER}$ **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 $j \in \texttt{UNDER}_t$ w.p. $\frac{|d_t(j)|}{d(t)}$
    end
  end
  $t = t + 1$; For every $i \in \{1, \ldots, m\}$ let $d_t(i) = L_t(M_i) - s_i \bar{L}$; $\texttt{OVER}_t = \{i | d_t(i) > 0\}$;
  $\texttt{UNDER}_t = \{i | d_t(i) < 0\}$ and $d(t) = \sum_{i \in \texttt{OVER}_t} d_t(i)$;
end

Algorithm 5: Distributed algorithm `BalanceMultipleSpeed` for multiple links

We now analyze the second phase. We divide the second phase three subphases. The second subphase starts when $unbalanced_t < \beta \log^{1.5}(1/\delta')$, and the third subpase starts when $unbalanced_t < \gamma \log(1/\delta')$, for some constants $\gamma, \beta > 0$ which will be specified from the proof. Note that $unbalanced_t$ can only decrease over times, since once a link is balanced it will remain this way. We start by analyzing the first subphase where $unbalanced_t \geq \beta \log^{1.5}(1/\delta')$. By Lemma 2.1, since $d_t(k) \leq 3 \ln(1/\delta')$, each unbalanced link terminates in one stage with probability at least $q = 1/\sqrt{6\pi \ln(1/\delta')}$. ¿From standard properties of Binomial distributions, the expected number of underloaded links, $under_t$ is at most $(1/2) unbalanced_{t-1}$. Since links become balanced with probability $q < 0.01$, we have that the expected number of overloaded links at time t, $E[over_t] \geq 0.49 unbalanced_{t-1}$. Let $\texttt{O}_t(i)$ be a random variable that indicates if $i \in \texttt{OVER}_t$, hence we have that $over_t = \sum_{i=1}^{m} \texttt{O}_t(i)$. By [7] the numbers of users on each link, $n_t(i)$, are negatively associated thus the random variables $\texttt{O}_t(i)$ are negatively associated. By [7] we can apply Chernoff bound on the number of overloaded links, $over_t$, to obtain that with probability at least $1 - \delta'$

$$over_t \geq 0.49\ unbalanced_{t-1} - \sqrt{3 unbalanced_{t-1} \ln(1/\delta')} \geq 0.48\ unbalanced_{t-1},$$

for $unbalanced_{t-1} \geq \gamma(\log 1/\delta')$, for some constant $\gamma > 0$. Our main goal is to show that in the first subphase, when there are many unbalanced links, in every stage many links become stable. The expected number of new balanced links at time t is at least $q\, over_{t-1}$. Since each overloaded link becomes balanced with probability q independently, we can apply Chernoff bound on the number of new balanced links, and show that with probability at least $1 - \delta'$,

$$unbalanced_t - unbalanced_{t-1} \geq 0.48 q\, unbalanced_{t-1} - \sqrt{3q\, unbalanced_{t-1} \log(1/\delta')} \geq 0.47 q\, unbalanced_{t-1}$$

for $unbalanced_{t-1} \geq \beta \log^{1.5}(1/\delta')$, for sufficiently large β. This implies that after $O((\log m)/q) = O(\log m \sqrt{\log 1/\delta'})$ the first subphase ends.

In the second subphase we have only $O(\log^{1.5} 1/\delta')$ unbalanced links. Still, at each stage, each unbalanced link becomes balanced with probability q. While $unbalanced_t \geq \gamma(\log(1/\delta'))$ we have that $over_t \geq 0.49 unbalanced_{t-1}$, with probability $1-\delta'$. This implies that with probability at least $1/2$ the number of links that become stable is at least $q\, unbalanced_{t-1} = O(\sqrt{\log(1/\delta')})$. Therefore with probability $1-\delta'$ after $O(\log(1/\delta'))$ we have only $O(\log(1/\delta'))$ unbalanced links. In the third subpahse, we analyze the convergence of the links one after the other. With probability $1-\delta'$ we have that in $O(\log^{1.5}(1/\delta'))$ stages all links are balanced. Adding all the different times,

$$\ell = c_1 \ln\ln(n) + c_2 \ln(m)\sqrt{\ln(\ell m/\delta)} + c_3 \ln(\frac{\ell m}{\delta}) + c_4 \ln^{1.5}(\frac{\ell m}{\delta})$$

we obtain that, $\ell = O(\ln\ln(n) + \ln^{1.5}(m/\delta)$. ∎

Once again we derive the simple corollary regarding the convergence expected time.

Corollary 4.4 *Algorithm* `BalanceMultipleSpeed` *terminates within expected* $\tilde{O}(\ln\ln(n)+\ln^{1.5}(m))$ *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 reroute 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 4.1 *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 $i, j \in$* `OVER` *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 $P_j = P_k$.*

Theorem 4.5 *If during every stage of the rerouting uses some policy in $P^t \in \Pi(g^t)$, the system converges to Nash equilibrium, with probability at least $1-\delta$, within $O(\ln\ln(n)+\ln^{1.5}(m/\delta))$ 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 C, is similar to the proof that a symmetric game has a symmetric Nash Equilibrium and is based on fixed pint theorem.)

Theorem 4.6 *For every configuration $g \in G$, there exists a Nash rerouting policy in $\Pi(g)$.*

A two phases rerouting policy is a policy that behaves as follows: every user on an overloaded link leaves the link with probability p. If a user leaves an overloaded link, then it will move to an underloaded link j with probability q_j. (Note that the various `Balance` algorithms a two phase rerouting policy.) We show that for some configurations no two phase rerouting policy ia a Nash rerouting policy, (proof is in appendix C).

Lemma 4.7 *There is an assignment function, g, for which no two phases rerouting policy is a Nash rerouting policy.*

Next we present an algorithm, which compute Nash rerouting policy, by enumerating the support of the policy.

Theorem 4.8 *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", manuscript.

[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,” pp. 404–413, *STACS 99.*

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

[17] 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.

[18] T. Roughgarden and E, Tardos, “How Bad is Selfish Routing?,” *In the Proceedings of the 41st Annual IEEE Symposium on the Foundations of Computer Science*, pp. 93-102, 2000.

A Section 2

Lemma A.1 *[2] Let $0 < p < 1$, let $Z_1, \ldots, Z_n$ 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,*

$$
\begin{aligned}
\mathbf{P}\left(p \le \hat{p} + \sqrt{\frac{2p\ln(1/\delta)}{n}}\right) &\ge 1-\delta \qquad & 0 \le p \le 1 \\
\mathbf{P}\left(p \ge \hat{p} - \sqrt{\frac{3p\ln(1/\delta)}{n}}\right) &\ge 1-\delta \qquad & \frac{\ln(1/\delta)}{3n} \le p \le 1 \\
\mathbf{P}\left(p \ge \hat{p} - \frac{2\ln(1/\delta)}{n}\right) &\ge 1-\delta \qquad & 0 \le p \le \frac{\ln(1/\delta)}{3n}
\end{aligned}
$$

B Proofs from Subsection 4.1

Proof of Theorem 4.1 : 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 become stable 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. For each underloaded link we terminate if only one user jumps to it. This probability can be lower bounded as follows,

$$
\begin{aligned}
\mathbf{P}(\text{underloaded link becomes stable}) &= \binom{\ell+\tau}{1}(1-\frac{1}{\ell+\tau})^{\ell+\tau-1}\frac{1}{\ell+\tau} \\
&= (1-\frac{1}{\ell+\tau})^{\ell+\tau-1} \ge \frac{1}{e}
\end{aligned}
$$

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 $1 - \delta$. ∎

C Proofs from Subsection 4.3

We first give characterization of every Nash equilibria which will be useful in the proof of Theorem 4.6

Lemma C.1 *In every Nash equilibria, 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 4.6 : We first note that any equilibria must satisfy the third condition. By restricting uses action 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 existing proof. Let Δ be the set of all possible polices, i.e. if $P \in \Delta$ then P_{ij} 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 we define the policy $\bar{P}_{i,j} = \frac{\sum_{i':g(i)=g(i')} P_{i',j}}{\sum_{i':g(i)=g(i')} 1_{P_{i',j}}}$ We define the utility function of the distribution $\bar{P}$ as,

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

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 C.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 4.7 : 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:

$$
\begin{aligned}
(n + d_1 - 1)(1 - p_1) &= n - d_3 + p_1(n + d_1 - 1)q_3 + p_2(n + d_2)q_3 && (1)\\
(n + d_1 - 1)(1 - p_1) &= n - d_4 + p_1(n + d_1 - 1)q_4 + p_2(n + d_2)q_4 && (2)\\
(n + d_2 - 1)(1 - p_2) &= n - d_3 + p_1(n + d_1)q_3 + p_2(n + d_2 - 1)q_3 && (3)\\
(n + d_2 - 1)(1 - p_2) &= n - d_3 + p_2(n + d_1)q_4 + p_2(n + d_2 - 1)q_4 && (4)
\end{aligned}
$$

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

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

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. ∎