Routing in Max-min Fair Networks: A Game Theoretic Approach

Dejun Yang, Guoliang Xue, Xi Fang, Satyajayant Misra and Jin Zhang

Abstract— **In this paper, we study the problem of routing in networks with max-min fair congestion control at the link level. The goal of each user is to maximize its own bandwidth by selecting its path. The problem is formulated as a non-cooperative game. We first prove the existence of Nash Equilibria. This is important, because at a Nash Equilibrium (NE), no user has the incentive to change its routing strategy. In addition, we investigate how the selfish behavior of the users may affect the performance of the network as a whole. We next introduce a novel concept of observed available bandwidth on each link. It allows a user to find a path with maximum bandwidth under max-min fair congestion control in polynomial time. We then present a game based algorithm to compute an NE and prove that by following the natural game course the network converges to an NE. Extensive experiments show that the network can converge to an NE in less than 10 iterations and also significantly improves the fairness compared with other algorithms. Our results have the implication for the future routing protocol design.**

Keywords: Max-min fair bandwidth allocation; non-cooperative game; Nash Equilibrium

1. Introduction

In communication networks, routing to maximize the bandwidth of the selected path is an important problem. Take for instance peer-to-peer networks, where several pairs of peers share volumes of data between each other. For each pair of peers, considered a user, the objective is a selfish one, which is to send as many packets as possible through the network while competing for network resources with other users. Given this selfish objective, a user in the network will change its path if the new path provides a larger bandwidth value even if it is at the expense of other users. To address the issue of allocating bandwidth among competing users, max-min fair bandwidth allocation is widely adopted as a congestion control scheme at the link level [6, 7, 13, 15–17, 20]. It achieves fairness by treating all paths passing through a link equally and assigning an equal share of bandwidth to each of them unless a path receives less bandwidth at another link.

In this paper, we study the problem of non-cooperative routing under max-min fair congestion control, where the goal of each user is to maximize the bandwidth of its chosen path. Two questions that need to be answered to address this problem are, *How can a user efficiently find a path with maximum bandwidth under max-min fair congestion control?* and *Will the network converge to a stable state or oscillate forever?* The first question is a key step in our convergence analysis, since it directly affects the speed of the convergence. It is also an independent problem to study, as we will point out later that the strong correlation among competing paths makes the calculation of available bandwidth on each link nontrivial. The second question is important because oscillating among different paths introduces dramatic overhead, consuming network resources. This paper answers both questions.

For the first question, we introduce the concept of *observed available bandwidth* and prove that it can accurately predict the bandwidth of a path. For the second question, we model the routing problem as a non-cooperative game and employ game theoretic tools to analyze the interaction among users. This question boils down to *the existence of a Nash Equilibrium* and *the convergence of the game*. Two major challenges arise while answering these questions. First, the measurement criterion is non-additive. The bandwidth of the chosen path is determined by one of the links in the path. Such a game is termed the *bottleneck game* in [1]. Second, when choosing a new path, the available bandwidth of a link may depend on the bandwidth of existing paths of other users. However, the bandwidths of these paths in turn depend on the bandwidth of the new path. Therefore the problem is significantly more involved than the traditional maximum capacity path problem [18].

The major contributions of this paper are as follows:

- We define the Maximal-Bandwidth Routing problem (formally defined in Section 3, and denoted as MAXBAR). Max-min fair share bandwidth allocation is used to address the fairness issue when allocating the bandwidth of a link to competing users. We model the MAXBAR problem as a non-cooperative game where each player makes the routing decision selfishly to maximize its own bandwidth.
- We prove the existence of Nash Equilibria in the MAXBAR game, where no player has the incentive to deviate from its chosen path. We then precisely quantify the *price of anarchy* of the MAXBAR game by proving that it is equal to the reciprocal of the number of users. As a byproduct, this gives an approximation to the social optimal solution to the MAXBAR problem.
- We introduce a novel method to compute the available bandwidth on each link. It allows us to find a path with maximum bandwidth for a new user in the network with multiple users and max-min fair share bandwidth allocation on links. This turns out to be non-trivial, as the widest path algorithm in [18] cannot be directly applied due

Yang, Xue, Fang and Zhang are all with the CIDSE Dept. at Arizona State University, Tempe, AZ 85287. Misra is with the CS Dept. at New Mexico State University, Las Cruces, NM 88003. Email: {dejun.yang, xue, xi.fang, zhang.jin}@asu.edu, misra@cs.nmsu.edu. This research was supported in part by ARO grant ARO W911NF-09-1-0467 and NSF grants 0830739 and 0901451. The information reported here does not reflect the position or the policy of the federal government.

to the mutual influence between paths sharing common links. Note that this difficulty was also studied in [15]. However, [15] only gave an approximation algorithm.
- We investigate the behaviors and incentives of the players in the game and present a game based algorithm to compute an NE. We prove that by following the natural game course, the MAXBAR game converges to an NE.

The rest of the paper is organized as follows. In Section 2, we present a brief overview of related work. In Section 3, we describe our system model, define the MAXBAR problem formally and formulate it as a non-cooperative game. In Section 4, we prove the existence of Nash Equilibria and quantify the inefficiency incurred by the lack of cooperation via price of anarchy. In Section 5, we present an efficient algorithm to select a path with maximum bandwidth in a max-min fair network with multiple users. In Section 6, we provide a comprehensive analysis of the MAXBAR game and prove the convergence to an NE. In Section 7, numerical experiments on randomly generated networks demonstrate that the game can converge to an NE rapidly (within 10 iterations) and significantly improve the fairness compared with other algorithms. We conclude this paper in Section 8.

2. **Related Work**

Max-min fair bandwidth allocation or congestion control has been proposed to address the issue of fairly and optimally allocating resources, bandwidth in particular, among multiple competing users [3, 13]. In [13], Jaffe first presented max-min fair bandwidth allocation and proved the optimality of this allocation scheme. He also proved the uniqueness of such an allocation. To implement a max-min fair network, Demers [7] proposed a fair queuing scheduler to be employed on gateways. Ma *et al.* [15] studied how to route in max-min fair networks to improve the total throughput of the network. They also formally described a simple centralized algorithm to calculate the max-min fair bandwidth for each path. Note that the routing algorithm was based on abstract information on the link, which only estimates the accurate available bandwidth. In [16], Mayer *et al.* showed that computing the exact max-min fair bandwidth for each path requires global information and thus designed a new local distributed scheduling algorithm to approximate max-min fair bandwidth allocation. Noting that the max-min fair bandwidth allocation scheme was proposed to achieve fairness at link level, Chen and Nahrstedt [6] extended the concept to the routing level. They defined the *fairness-throughput* by introducing a new set of relational operators to compare two different feasible bandwidth allocations at routing level. The fairness-throughput performance of the bandwidth allocation is maximized if and only if such an allocation is the largest under the relational operator. A max-min fair routing algorithm was proposed to select a path for the new user to maximize the minimum bandwidth allocated to all the users. By allowing the routing to be splittable, Nace *et al.* [17] gave a linear programming based algorithm to compute the optimal max-min fair bandwidth allocation. Schapira *et al.* [20] and Godfrey *et al.* [10] studied the efficiency and incentive compatibility of different congestion control schemes in the network where users' paths are fixed. They also presented a family of congestion control protocols called Probing Increase Educated Decrease and showed that by following any of these protocols, the network converges to a fixed point.

All the previous works mainly focused on either the case where paths are fixed [7, 13, 16] or the case where routing aims to improve the total performance [6, 10, 15, 17, 20]. In contrast, the objective of our work is to investigate the scenario where each user in the network is able to adapt its routing decision based on the current environment and driven by its own selfish objective. There are also important works on stable routing in the literature [9, 11, 12]. We note that these works did not consider max-min fair bandwidth allocation in their models.

3. **System Model and Problem Formulation**

In this section, we first describe the network model. We then discuss a well known congestion control scheme and an algorithm for computing the bandwidth for each path under this congestion control scheme. Finally, we formally formulate the problem we will study in this paper.

A. Network Model

We model the network by a weighted undirected graph denoted by $G = (V, E, b)$, where V is the set of n nodes, E is the set of m links, and b is a weight function such that $b(e) = b(v, w) > 0$ is the bandwidth of link $e = (v, w) \in E$. In the network, there is a collection $\mathcal{U} = \{1, 2, \ldots, K\}$ of *users*. User $i \in \mathcal{U}$ needs to transmit packets from a source node s_i to a destination node t_i over an s_i–t_i path. An s–t path in the network consists of an ordered sequence of vertices $s{=}v_0$, v_1, …, $v_q{=}t$, where $(v_l, v_{l+1}) \in E$ for $0 \leq l < q$. We denote such a path by v_0-v_1-…-v_q. We are only interested in simple paths–for which the nodes in the sequence are distinct. Although there may be multiple s_i–t_i paths, at any given time, user i will use only one of the paths, which is denoted by P_i. We denote the set of paths used by the users as $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$. We denote the set of users sharing link e by $\mathcal{U}(e)$, i.e., $\mathcal{U}(e) = \{x | x \in \mathcal{U} \text{ and } e \in P_x\}$. Let $u(e) = |\mathcal{U}(e)|$ denote the number of users sharing link e.

For routing approach, we will use link-state source routing algorithms as in [15]. In such routing schemes, each node knows the network topology and the state on each link [2, 21]. Thus, it is possible for the node to select its path. In this paper, we consider best-effort flows [15], which assumes that the source node always has sufficient data to transmit.

B. Congestion Control

Since multiple users are competing for bandwidth resources, congestion control is necessary for the management of bandwidth. The employed congestion control needs to satisfy two requirements: (1) the bandwidth allocation is fair and (2) the bandwidth is fully allocated. A simple way to allocate the bandwidth of a link to multiple competing paths is to share it equally among them. However, some paths can only use less than the equal share, while some can use more. Obviously, equal allocation is not desirable. In this paper, we assume that at the link level, *max-min fair bandwidth allocation* (also known as fair queuing) [7, 13] is used for congestion control. Max-min fair bandwidth allocation has been recognized as the optimal throughput-fairness definition [13, 16]. Intuitively, if there are multiple users sharing a common link, each will get a "fair share" of the link bandwidth. If some user cannot use up its fair share bandwidth because it has lower share assigned on another link, the excess bandwidth is "fairly" split among all other users. Such a network with max-min fair congestion control at the link level is called a *max-min fair network*. We denote the bandwidth allocated to path P_i in a max-min fair network by $b(P_i)$ (how to compute the value of $b(P_i)$ will be shown later). We use $\mathbf{b} = (b(P_1), b(P_2), \ldots, b(P_K))$ to denote the Max-min Fair Bandwidth Allocation (MFBA) for an instance $(G, \mathcal{P})$ given by graph $G(V, E, b)$ and the users' paths $\mathcal{P}$. The uniqueness of MFBA has been proved in [20]. While assigning the bandwidth to each path P_i, there must exist at least one link that keeps the path from obtaining more bandwidth. We call such link a *bottleneck* of path P_i. Note that there could be more than one bottleneck for a path. We use $\mathcal{B}(P_i)$ to denote the *set of all bottlenecks* of path P_i. Each bottleneck e of path P_i has two important properties, which can be mathematically expressed as follows:

1) $\sum_{x\in\mathcal{U}(e)} b(P_x) = b(e)$,
2) $b(P_i) \geq b(P_x), \forall x \in \mathcal{U}(e)$.

Property 1) means link e is saturated. We call a link *saturated* if its bandwidth is fully allocated. This property is obvious as otherwise P_i could obtain more bandwidth if there is any available. Property 2) states that there is no path being allocated more bandwidth than P_i on link e. The reason is that if there exists another path P_j allocated more bandwidth, P_i could equally share the bandwidth with P_j due to max-min fair bandwidth allocation and obtain more bandwidth. These two properties have also been proved by Lemma 3 in [6] and Lemma 3 in [13]. It is noted that the proof in [13] is more general in the sense that they allow the total available bandwidth on each link to be less than $b(e)$.

Algorithms for calculating the bandwidth allocation for each path in a max-min fair network have been proposed in [13, 15]. To make our paper self-contained, we illustrate the pseudo code in Algorithm 1. For detailed description and correctness proof, we refer the readers to [13, 15].

The basic idea of Algorithm 1 is that in each iteration, we find a *global bottleneck* e_g, which is defined as the link having the least equal share, that is $e_g = argmin_{e\in E}\frac{b(e)}{u(e)}$. We allocate the equal share of $b(e_g)$ to all users in $\mathcal{U}(e_g)$. Then all the paths of users in $\mathcal{U}(e_g)$ are removed from the network. The link bandwidths are reduced by the bandwidth consumed by removed users. The above procedure is repeated until all the paths have been assigned bandwidth and removed from the network. To illustrate the idea of Algorithm 1, we compute the bandwidth for the example in Fig. 1. In Fig. 1(a), (v_4, t_2) is the e_g selected in the first iteration and $\frac{b(v_4,t_2)}{u(v_4,t_2)} = 3$. Since user 2 (blue dotted) is the only one using link (v_4, t_2), we set $b(P_2) = 3$, remove path P_2 from the network and subtract the bandwidth from all the links along path P_2 (blue dotted). In the resulting network shown in Fig. 1(b), (v_1, v_2) is selected as e_g. There are two paths, path P_1 (red solid) and path P_3 (green dashed), sharing link (v_1, v_2). Each of them obtains bandwidth $\frac{b(v_1,v_2)}{u(v_1,v_2)} = 4$. We set $b(P_1) = b(P_3) = 4$, and remove path P_1 and path P_3. Since there is no more paths left, the algorithm terminates.

Algorithm 1 $ComB(G, b, \mathcal{P}, \mathcal{U})$

```
Input:   Instance $(G, \mathcal{P})$ and user set $\mathcal{U}$.
Output: $b(P_x)$ for all $x \in \mathcal{U}$.
 1: $E_c \leftarrow E$, $\mathcal{P}_c \leftarrow \mathcal{P}$, $b_c(\cdot) \leftarrow b(\cdot)$, $u_c(\cdot) \leftarrow u(\cdot)$;
 2: $b(P_x) \leftarrow 0, \forall x \in \mathcal{U}$;
 3: repeat
 4:     Let $e_g := argmin_{e\in E}\frac{b_c(e)}{u_c(e)}$ in $G(V, E_c, b_c)$;
 5:     $b_{temp} \leftarrow \frac{b_c(e_g)}{u_c(e_g)}$;
 6:     for each player $x \in \mathcal{U}_c(e_g)$ do
 7:         $b(P_x) \leftarrow b_{temp}$;
 8:         for all $e \in P_x$ do
 9:             $b_c(e) \leftarrow b_c(e) - b(P_x)$;
10:             if $b_c(e) = 0$ then $E_c \leftarrow E_c \setminus \{e\}$; end if
11:             $u_c(e) \leftarrow u_c(e) - 1$;
12:         end for
13:         $\mathcal{P}_c \leftarrow \mathcal{P}_c \setminus \{P_x\}$;
14:     end for
15: until $\mathcal{P}_c = \emptyset$
16: return  $b(P_x)$ for all $x \in \mathcal{U}$.
```

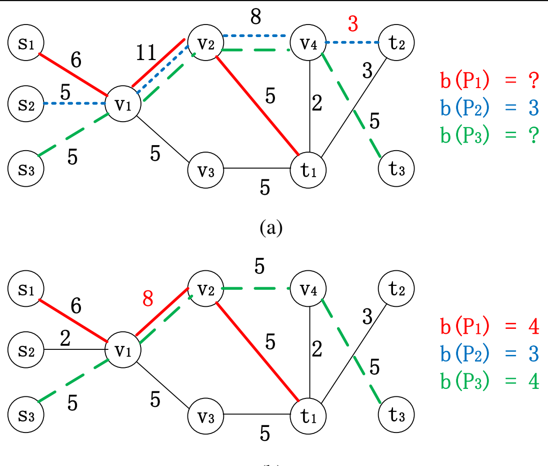


Fig. 1. An example with 3 users. $P_1 = s_1$-v_1-v_2-t_1 (red solid), $P_2 = s_2$-v_1-v_2-v_4-t_2 (blue dotted), and $P_3 = s_3$-v_1-v_2-v_4-t_3 (green dashed).

C. Problem Formulation

We are interested in the problem of routing in the max-min fair network with multiple users, where each user aims to maximize the bandwidth of its own path. We call this problem the ***MAX**imal-**BA**ndwidth **R**outing* (MAXBAR) problem. The MAXBAR problem can be formulated as a non-cooperative game as follows. Each user is a *player* in this game. We define the *strategy* of player i as its path. For player i, the strategy space $\mathcal{P}_i$ consists of all s_i–t_i paths. The strategies of all players define the *strategy vector*: $S = (P_1, P_2, \ldots, P_K) \in \mathcal{S}$, where $\mathcal{S} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_K$. We denote the strategies except player i's by S_{-i}. We denote the *payoff* of player i by $U_i(S)$, which is equal to the bandwidth of path P_i, that is, $U_i(S) = b(P_i)$. We assume that players are rational and selfish. Each player makes independent routing decision to maximize its payoff in the network.

An important subproblem of the MAXBAR problem, which is of independent interest as well, is to select a path for a user to maximize the allocated bandwidth, given the network and other users' paths. This is known as *best response* in game theory.

Definition 3.1: [*Best Response Routing*] Given other users' paths $P_1, \cdots, P_{i-1}, P_{i+1}, \cdots, P_K$, the *best response routing* for user i is to select a path P_i such that $b(P_i)$ is maximized after computing the MFBA for $(G, \mathcal{P})$. □

Finding a best response path for a user is not straightforward. As we learned from previous discussions, the allocated bandwidth for each path can be computed after considering the whole network topology and all path selections. Thus how to compute the available bandwidth on each link before routing is known has not been solved yet. This problem was also studied in [15]. However, they only gave estimated information for each link and their algorithm is an approximation. We will present an efficient solution to this problem in Section 5.

In order to study the strategic interactions of the players, we first introduce the concept of Nash Equilibrium [8].

Definition 3.2: [*Nash Equilibrium*] A strategy vector $S^{ne} = (P_1^{ne}, P_2^{ne}, \ldots, P_K^{ne})$ is called a Nash Equilibrium (NE), if for every player i, we have:

$$U_i(P_i^{ne}, S_{-i}^{ne}) \geq U_i(P_i', S_{-i}^{ne})$$

for every strategy $P_i' \in \mathcal{P}_i$. □

In other words, *in an NE, no player can increase its payoff by unilaterally changing its strategy*.

The *social optimum* in the MAXBAR game is a strategy vector S^* such that the total payoff, i.e. $\sum_{i \in \mathcal{U}} U_i(S)$, is maximized among all $S \in \mathcal{S}$. We use the concept of price of anarchy defined in [14] to quantify the system inefficiency due to selfishness.

Definition 3.3: [*Price of Anarchy*] The *price of anarchy* (POA) of a game is the ratio of the total payoff achieved in a worst possible NE over that of the social optimum. □

Now we introduce more notations. When player i's path is not in the network, we use $\mathbf{b}_{-i}$ to denote the MFBA and use $b_{-i}(P_x)$ to denote the bandwidth of path P_x, where we assume $b_{-i}(P_i) = 0$ as a technical convention. Let $\mathcal{U}_{-i}(e)$ denote the set of players sharing link e and $u_{-i}(e) = |\mathcal{U}_{-i}(e)|$. After player i changes its path to P_i', let $\{P_1', P_2', \ldots, P_K'\}$ denote the paths of all players, where $P_x' = P_x$ if $x \neq i$. Let $\mathbf{b}'$ denote the MFBA and $b'(P_x')$ denote the new bandwidth of path P_x'. Let $\mathcal{U}'(e)$ denote the set of players sharing link e and $u'(e) = |\mathcal{U}'(e)|$. It is clear that $\mathcal{U}'(e) = \mathcal{U}_{-i}(e) \cup \{i\}$ if $e \in P_i'$ and $\mathcal{U}'(e) = \mathcal{U}_{-i}(e)$ otherwise.

4. Existence of Nash Equilibria

To prove the existence of NE, we first show that every time a player changes its path, the minimum bandwidth among the paths whose bandwidths change increases strictly.

Lemma 4.1: Assume that player i selfishly changes its path from P_i to P_i'. Let $\mathcal{U}_= = \{x \in \mathcal{U} | b(P_x) = b'(P_x')\}$, $\mathcal{U}_\uparrow = \{x \in \mathcal{U} | b(P_x) < b'(P_x')\}$ and $\mathcal{U}_\downarrow = \{x \in \mathcal{U} | b(P_x) > b'(P_x')\}$. We have $\min_{x \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b'(P_x') > \min_{x \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b(P_x)$. □

Proof. It is clear that $i \in \mathcal{U}_\uparrow$. Otherwise, player i has no incentive to change its path from P_i to P_i'. First we claim that, *for any $j \in \mathcal{U}_\downarrow$, there exists $k \in \mathcal{U}_\uparrow$, such that $b'(P_j') \geq b'(P_k')$.* Let $e \in \mathcal{B}'(P_j')$ be a bottleneck of P_j' in $\mathbf{b}'$. By Property 2) of bottleneck, we have $b'(P_j') \geq b'(P_x')$, $\forall x \in \mathcal{U}'(e)$. Therefore, it only needs to prove that there exists a player $k \in \mathcal{U}'(e) \cap \mathcal{U}_\uparrow$. If $i \in \mathcal{U}'(e)$, then we can take $k = i$. Next, we consider the case where $i \notin \mathcal{U}'(e)$. Note that $i \notin \mathcal{U}'(e)$ implies that $\mathcal{U}'(e) \subseteq \mathcal{U}(e)$. Assuming to the contrary that $b(P_x) \geq b'(P_x')$, $\forall x \in \mathcal{U}'(e)$, the total bandwidth usage on link e in $\mathbf{b}'$ is

$$b'(P_j') + \sum_{x \in \mathcal{U}'(e) \setminus \{j\}} b'(P_x') < b(P_j) + \sum_{x \in \mathcal{U}(e) \setminus \{j\}} b(P_x) \leq b(e),$$

where the first inequality follows from $j \in \mathcal{U}_\downarrow$ and $\mathcal{U}'(e) \subseteq \mathcal{U}(e)$, the second inequality follows from the feasibility of $\mathbf{b}$. This contradicts the fact that e is a bottleneck in $\mathbf{b}'$, and proves the existence of player k in the case where $i \notin \mathcal{U}'(e)$.

In summary, $\forall j \in \mathcal{U}_\downarrow$, there $\exists k \in \mathcal{U}_\uparrow$ such that

$$b(P_j) > b'(P_j') \geq b'(P_k') > b(P_k).$$

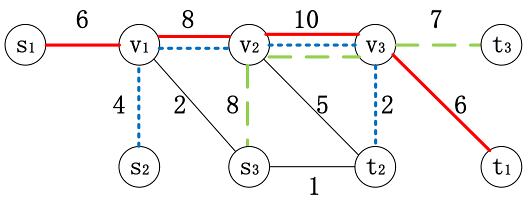
Following this result, we know that

$$\min_{x \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b'(P_x') = b'(P_{k_1}')$$

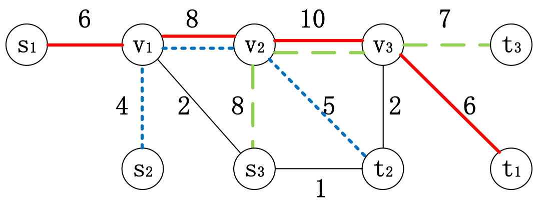
for some player $k_1 \in \mathcal{U}_\uparrow$ and

$$\min_{x \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b(P_x) = b(P_{k_2})$$

for some player $k_2 \in \mathcal{U}_\uparrow$. Since $k_1 \in \mathcal{U}_\uparrow$, we know that $b'(P_{k_1}') > b(P_{k_1}) \geq b(P_{k_2})$. Hence this lemma holds. ■



(a) Before player 2 changes its path



(b) After player 2 changes its path

Fig. 2. An example for Lemma 4.1.

We use the example in Fig. 2 to illustrate the meaning of Lemma 4.1. In this example, we have three players: player 1 (red solid), player 2 (blue dotted) and player 3 (green dashed). From Fig. 2(a) to Fig. 2(b), player 2 changes its path from $P_2 = s_2$-v_1-v_2-v_3-t_2 to $P_2' = s_2$-v_1-v_2-t_2. Before the change, $b(P_1) = 4$, $b(P_2) = 2$ and $b(P_3) = 4$. After the change, $b'(P_1') = 4$, $b'(P_2') = 4$ and $b'(P_3') = 6$. In this example, $\mathcal{U}_= = \{1\}$, $\mathcal{U}_\uparrow = \{2,3\}$ and $\mathcal{U}_\downarrow = \emptyset$. We have $\min\{b'(P_2'), b'(P_3')\} > \min\{b(P_2), b(P_3)\}$.

Now, we are ready to prove the existence of NE in every instance of the MAXBAR game.

Theorem 4.1: For every instance of the MAXBAR game, there exists at least one NE. □

Proof. At every stage of the game, we arrange the bandwidth values of the paths lexicographically in a non-decreasing order, resulting in a vector $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_K)$. In this vector, the minimum bandwidth b_1 is at the most significant coordinate. We have $b_\kappa \le b_{\kappa+1}$ for $1 \le \kappa < K$. For any two vectors $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_K)$ and $\vec{\mathbf{b}}_l' = (b_1', b_2', \ldots, b_K')$, $\vec{\mathbf{b}}_l < \vec{\mathbf{b}}_l'$ in lexicographic order if and only if:

- $b_1 < b_1'$, or
- $\exists\ 1 < \tau \le K$, $b_\kappa = b_\kappa'$ for $1 \le \kappa < \tau$ and $b_\tau < b_\tau'$.

By Lemma 4.1, we conclude that every time a player changes its path, the ordering $\vec{\mathbf{b}}_l$ increases lexicographically. We know that there are a finite number of paths for each player. Thus the number of different path configurations is finite as well. As each path configuration corresponds to one vector, we pick the path set corresponding to the largest vector as the strategies for the players. We conclude that this is an NE. ■

While we know the existence of NE, we still have open questions to answer. How to *efficiently* find a path with maximum bandwidth in a max-min fair network? Will the MAXBAR game converge to an NE? We will answer these questions in Sections 5 and 6, respectively.

Now, we quantify the worst-case "penalty" incurred by the lack of cooperation among the players in this game using the concept of *price of anarchy* (POA). Recall that POA is the ratio of the total bandwidth of the worst NE to the total bandwidth of the social optimum among all strategies.

Theorem 4.2: For the MAXBAR game, POA $= \frac{1}{K}$. □

Proof. We prove this theorem by proving an upper bound in Lemma 4.2 and a lower bound in Lemma 4.3. ■

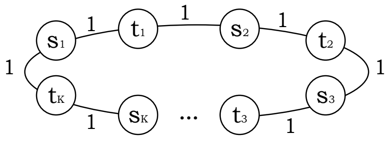


Fig. 3. An example for which the POA is $\frac{1}{K}$.

Lemma 4.2: For the MAXBAR game, POA $\le \frac{1}{K}$. □

Proof. We prove this lemma with the help of an example. Fig. 3 depicts (partly) a network with K players. In this network, the bandwidth of each link is 1. All the source-destination pairs are located clockwise on a ring topology. The source and destination for the same player are next to each other. Clearly, there are only two s_i–t_i paths for each player i, the counterclockwise path s_i-t_{i-1}-s_{i-1}-…-t_1-s_1-t_K-s_K-…-t_i and the clockwise path s_i-t_i. If each player i chooses the counterclockwise s_i–t_i path, the resulting strategy vector is an NE. The reason is that if any player i defects from the current strategy and chooses the clockwise s_i–t_i path, it results in the same bandwidth $1/K$. The total payoff in this NE is 1.

Next, we consider the social optimum where each player chooses the clockwise path. The total payoff is K. This shows that the POA of the MAXBAR game is at most $1/K$. ■

Lemma 4.3: For the MAXBAR game, POA $\ge \frac{1}{K}$. □

Proof. Let $(P_1, P_2, \ldots, P_K)$ be any NE of the MAXBAR game. Let $(P_1^*, P_2^*, \ldots, P_K^*)$ be the social optimum. We first claim that $b(P_i) \ge \frac{b^*(P_i^*)}{K}$ *for any player i, where* $b^*(P_i^*)$ *is the bandwidth of* P_i^* *in the social optimum.* Since $(P_1, P_2, \ldots, P_K)$ is an NE, no player has an incentive to change its path. Then we have

$$b(P_i) \ge b'(P_i^*) \ge \frac{b(e^*)}{K}, \tag{4.1}$$

where e^* is a bottleneck of P_i^* after player i unilaterally changes its path from P_i to P_i^*. The second inequality follows from the fact that each link can be shared by at most K players. In the social optimum, we have $b^*(P_i^*) \le b(e)$ for any $e \in P_i^*$. Plugging it into (4.1), we proved our claim. Based on the claim, we have the total payoff is

$$\textstyle\sum_{x\in\mathcal{U}} b(P_x) \ge \frac{\sum_{x\in\mathcal{U}} b^*(P_x^*)}{K}$$

for any NE. Hence, we have $b(NE_{worst}) \ge \frac{b(OPT)}{K}$, where $b(NE_{worst})$ is the total bandwidth in the worst Nash Equilibrium and $b(OPT)$ is the total bandwidth in the social optimum. Therefore, we have $\frac{b(NE_{worst})}{b(OPT)} \ge \frac{1}{K}$. ■

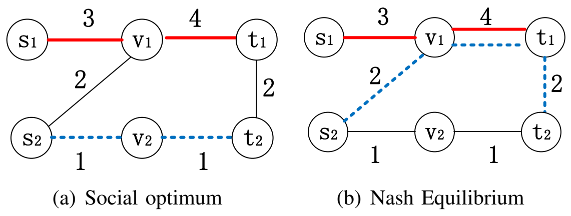


Fig. 4. An example showing that a social optimum is not an NE.

Remark 4.1. Note that the social optimum in Fig 3 is also an NE. Nevertheless, a simple example in Fig. 4 shows that *a social optimum is not necessarily an NE.*

Remark 4.2. Though the social optimum is the most desirable for the network, efficient algorithms to compute it are still open. Simple brute fore algorithms can take exponential time, since the number of s-t paths for a single player is exponential in the size of the network.

5. Best Response Routing in Max-min Fair Networks

An important step in the MAXBAR game is for a player to decide whether it has incentive to change its strategy unilaterally. Intuitively, it is natural for the player to unilaterally change its strategy to one that would give it the maximum payoff. However, the payoff of the chosen path depends on other players' strategies due to the competition among players sharing the same links. Obviously, the player can try all of its strategies and pick the one that gives it maximum payoff. However, it may take exponential time as the number of strategies of the user may not be polynomially bounded.

In this section, we introduce the novel concept of *observed available bandwidth* (formally defined later in this section) and prove the following facts: **(1) the observed available bandwidth on all links can be computed in $O(K\log K + mK)$ time; (2) the widest s_i-t_i path with regard to the observed available bandwidth is a best response routing for player** i. Hence, player i can compute its best response routing in polynomial time. Therefore, *player i has the incentive to change its strategy if and only if the payoff corresponding to its best response strategy is larger than its current strategy.* Given the challenges outlined at the beginning of this section, our results are significant. Although the facts are seemingly simple, the proofs are quite involved, which are the subjects of the rest of this section.

To get an intuition for calculating the available bandwidth, we take the link in Fig. 5 as an example. In this example, we assume that player $i = 4$ needs to find a path. Further assume that $\mathcal{U}_{-i}(e) = \{1, 2, 3\}$ and $b(e) = 11$. Also, $b_{-i}(P_1) = 1$, $b_{-i}(P_2) = 3$, and $b_{-i}(P_3) = 7$. After player i joins, it is clear that player 1 would not lose its bandwidth share, since it has less than the equal share, that is, $b_{-i}(P_1) = 1 < \frac{11}{4}$. If player i competes the bandwidth with players 2 and 3 for the residual bandwidth of 10, each of them gets bandwidth of $\frac{10}{3}$. We know before i joins, player 2 only uses bandwidth of 3 which is less than $\frac{10}{3}$. Therefore, only i and 3 will compete for the residual bandwidth and get bandwidth of $\frac{7}{2}$ each.

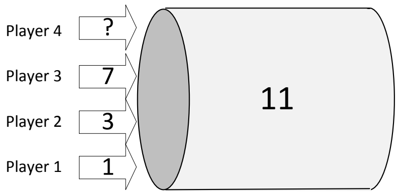


Fig. 5. A link with max-min fair bandwidth allocation, where there are three players before player 4 joins.

To capture the process we conducted above, we introduce the concept of *observed available bandwidth*. Assume that all players except i have their paths fixed. Now player i needs to find a path with maximum bandwidth in the current network. The *observed available bandwidth* $b^o(e)$ of link $e \in E$ is

$$b^o(e) = \frac{b(e) - \sum_{x\in\mathcal{U}_{-i}(e)\setminus\tilde{\mathcal{U}}_{-i}(e)} b_{-i}(P_x)}{|\tilde{\mathcal{U}}_{-i}(e)| + 1}, \tag{5.1}$$

where

$$\tilde{\mathcal{U}}_{-i}(e) = \{x | x \in \mathcal{U}_{-i}(e) \text{ and } b_{-i}(P_x) \geq \frac{b(e) - \sum\limits_{y\in\hat{\mathcal{U}}_{-i}(e,x)} b_{-i}(P_y)}{u_{-i}(e) - |\hat{\mathcal{U}}_{-i}(e,x)| + 1}\}$$

is the set of players such that for any player x in this set, the new bandwidth $b'(P'_x)$ is at least as large as the bandwidth of the new path P'_i of player i and

$$\hat{\mathcal{U}}_{-i}(e,x) = \{y | y \in \mathcal{U}_{-i}(e) \text{ and } b_{-i}(P_y) < b_{-i}(P_x)\}$$

is the set of players who are using less bandwidth than player x on link e. If we first sort the paths according to their bandwidth values, then for each link e we can compute both $\tilde{\mathcal{U}}_{-i}(e)$ and $\hat{\mathcal{U}}_{-i}(e,x)$ in $O(K)$ additional time. Thus we can compute $b^o(e)$ for all links $e \in E$ in $O(K\log K + mK)$ time. Accordingly, the *observed bandwidth* of the new path P'_i is

$$b^o(P'_i) = \min\nolimits_{e\in P'_i} b^o(e), \tag{5.2}$$

and the set of *observed bottlenecks* of path P'_i is

$$\mathcal{B}^o(P'_i) = argmin_{e\in P'_i} b^o(e).$$

Considering the example in Fig. 5, we have $\hat{\mathcal{U}}_{-i}(e,1) = \emptyset$, $\hat{\mathcal{U}}_{-i}(e,2) = \{1\}$, and $\hat{\mathcal{U}}_{-i}(e,3) = \{1,2\}$. The set $\tilde{\mathcal{U}}_{-i}(e)$ is $\{3\}$. Therefore, $b^o(e) = \frac{11-1-3}{1+1} = \frac{7}{2}$.

The properties of the observed available bandwidth are summarized in the following lemmas, which will be used in the later proofs in the rest of this section.

Lemma 5.1: Assume that $x \in \tilde{\mathcal{U}}_{-i}(e)$. For all $z \in \mathcal{U}_{-i}(e)$, if $b_{-i}(P_z) \geq b_{-i}(P_x)$, then $z \in \tilde{\mathcal{U}}_{-i}(e)$. □

Proof. It is obvious that if $b_{-i}(P_z) = b_{-i}(P_x)$, then $z \in \tilde{\mathcal{U}}_{-i}(e)$. Next, we prove that if $b_{-i}(P_z) > b_{-i}(P_x)$, then $z \in \tilde{\mathcal{U}}_{-i}(e)$. Let $\mathcal{X} = \hat{\mathcal{U}}_{-i}(e,z) \setminus \hat{\mathcal{U}}_{-i}(e,x)$. We have

$$\begin{aligned}
& b_{-i}(P_z) - \frac{b(e) - \sum_{y\in\hat{\mathcal{U}}_{-i}(e,z)} b_{-i}(P_y)}{u_{-i}(e) - |\hat{\mathcal{U}}_{-i}(e,z)| + 1} \\
= \;& b_{-i}(P_z) - \frac{b(e) - \sum\limits_{y\in\hat{\mathcal{U}}_{-i}(e,x)} b_{-i}(P_y) - \sum\limits_{y\in\mathcal{X}} b_{-i}(P_y)}{u_{-i}(e) - (|\hat{\mathcal{U}}_{-i}(e,x)| + |\mathcal{X}|) + 1} \\
> \;& b_{-i}(P_x) - \frac{b(e) - \sum\limits_{y\in\hat{\mathcal{U}}_{-i}(e,x)} b_{-i}(P_y) - |\mathcal{X}| b_{-i}(P_x)}{u_{-i}(e) - (|\hat{\mathcal{U}}_{-i}(e,x)| + |\mathcal{X}|) + 1} && (5.3) \\
\geq \;& 0, && (5.4)
\end{aligned}$$

where (5.3) follows from the fact that $y \in \mathcal{U}_{-i}(e) \setminus \hat{\mathcal{U}}_{-i}(e,x)$ implies $b_{-i}(P_y) \geq b_{-i}(P_x)$, and (5.4) follows from the fact that $x \in \tilde{\mathcal{U}}_{-i}(e)$. Hence we have $z \in \tilde{\mathcal{U}}_{-i}(e)$. ■

Lemma 5.2: For all $y \in \mathcal{U}_{-i}(e)$, if $y \in \tilde{\mathcal{U}}_{-i}(e)$, we have $b_{-i}(P_y) \geq b^o(e)$; otherwise, we have $b_{-i}(P_y) < b^o(e)$. □

Proof. Let x be the player whose path has the minimum bandwidth in $\tilde{\mathcal{U}}_{-i}(e)$. Thus we have $\hat{\mathcal{U}}_{-i}(e,x) \subseteq \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)$. For all $y \in \mathcal{U}_{-i}(e)$, if $b_{-i}(P_y) \geq b_{-i}(P_x)$, it follows from Lemma 5.1 that $y \in \tilde{\mathcal{U}}_{-i}(e)$. Thus we have $\hat{\mathcal{U}}_{-i}(e,x) \supseteq \mathcal{U}_{-i}(e)\setminus\tilde{\mathcal{U}}_{-i}(e)$. Therefore $\hat{\mathcal{U}}_{-i}(e,x) = \mathcal{U}_{-i}(e)\setminus\tilde{\mathcal{U}}_{-i}(e)$. Since $x \in \tilde{\mathcal{U}}_{-i}(e)$, we have

$$b_{-i}(P_x) \geq \frac{b(e) - \sum\limits_{y\in\hat{\mathcal{U}}_{-i}(e,x)} b(P_y)}{u_{-i}(e) - |\hat{\mathcal{U}}_{-i}(e,x)| + 1} \tag{5.5}$$

$$\begin{aligned} &= \frac{b(e) - \sum_{y \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b_{-i}(P_y)}{|\tilde{\mathcal{U}}_{-i}(e)| + 1} \\ &= b^o(e). \end{aligned} \tag{5.6}$$

Moreover, we have $b_{-i}(P_y) \geq b_{-i}(P_x) \geq b^o(e)$, $\forall y \in \tilde{\mathcal{U}}_{-i}(e)$. Next, we prove the second part of the lemma. If $y \notin \tilde{\mathcal{U}}_{-i}(e)$, we know $b_{-i}(P_y) < b_{-i}(P_x)$. Now assume that z is the player whose path has the maximum bandwidth in $\mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)$. Then, we have $b_{-i}(P_y) = b_{-i}(P_z)$, $\forall y \in \hat{\mathcal{U}}_{-i}(e,x) \setminus \hat{\mathcal{U}}_{-i}(e,z)$. Let $\mathcal{Z} = \hat{\mathcal{U}}_{-i}(e,x) \setminus \hat{\mathcal{U}}_{-i}(e,z)$. We have

$$\begin{aligned} & b_{-i}(P_z) - b^o(e) \\ = \;& b_{-i}(P_z) - \frac{b(e) - \sum_{y \in \hat{\mathcal{U}}_{-i}(e,x)} b_{-i}(P_y)}{u_{-i}(e) - |\hat{\mathcal{U}}_{-i}(e,x)| + 1} & (5.7) \\ = \;& b_{-i}(P_z) - \frac{b(e) - (\sum_{y \in \hat{\mathcal{U}}_{-i}(e,z)} b_{-i}(P_y) + \sum_{y \in \mathcal{Z}} b_{-i}(P_y))}{u_{-i}(e) - (|\hat{\mathcal{U}}_{-i}(e,z)| + |\mathcal{Z}|) + 1} \\ = \;& b_{-i}(P_z) - \frac{b(e) - \sum_{y \in \hat{\mathcal{U}}_{-i}(e,z)} b_{-i}(P_y) - |\mathcal{Z}| b_{-i}(P_z)}{u_{-i}(e) - (|\hat{\mathcal{U}}_{-i}(e,z)| + |\mathcal{Z}|) + 1} \\ < \;& 0, & (5.8) \end{aligned}$$

where (5.7) follows from (5.5) and (5.6), (5.8) follows from the fact that $z \notin \tilde{\mathcal{U}}_{-i}(e)$. In addition, we know that $b_{-i}(P_y) \leq b_{-i}(P_z) < b^o(e)$, $\forall y \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)$. ■

We now prove that the observed available bandwidth defined above accurately calculates the bandwidth on each link in the sense that after we choose a path with maximum observed bandwidth and reallocate the bandwidth for each path using Algorithm 1, the new allocated bandwidth of the path is equal to the observed bandwidth.

We use proof by contradiction, and the sketch of our proof is as follows. If the new allocated bandwidth of the path is not equal to its observed bandwidth, two cases may happen: (1) the path is allocated more bandwidth than the observed bandwidth, or (2) the path is allocated less bandwidth than the observed bandwidth. For each case, we show that it will lead to a chain reaction, which results in a contradiction. We analyze two phenomena that may occur and cause the chain reaction after a player chooses its new path based on the observed available bandwidth. In Lemma 5.3 (resp. Lemma 5.4), we show that the decrease (resp. increase) of the bandwidth of a path must be directly related to the increase (resp. decrease) of that of another path. More importantly, the relation between new bandwidth values of these two paths satisfies certain rules. In order to facilitate the understanding of these lemmas, an example is presented in Fig. 6.

Lemma 5.3: Let P'_i be the new s_i-t_i path chosen by player i based on observed available bandwidth. We have

1) Let $e \in \mathcal{B}'(P'_i)$ be a bottleneck of path P'_i in $\mathbf{b}'$. If $b'(P'_i) < b^o(P'_i)$, then $\exists k \in \mathcal{U}'(e)$, such that (**1a**) $b'(P'_k) > b_{-i}(P_k)$ and (**1b**) $b'(P'_i) \geq b'(P'_k)$.
2) Let $e \in \mathcal{B}'(P'_j)$ be a bottleneck of path P'_j in $\mathbf{b}'$. If $b'(P'_j) < b_{-i}(P_j)$, then $\exists k \in \mathcal{U}'(e)$, such that (**2a**)$b'(P'_k) > b_{-i}(P_k)$ and (**2b**) $b'(P'_j) \geq b'(P'_k)$. □

Proof. We prove Part 1) and 2) separately:

- **Part 1):** $b'(P'_i) < b^o(P'_i)$.
  By Property 2) of bottleneck, we know that $b'(P'_i) \geq b'(P'_x)$, $\forall x \in \mathcal{U}'(e)$. Thus it suffices to prove (**1a**). We prove (**1a**) by contradiction. Assume that $b'(P'_x) \leq b_{-i}(P_x)$, $\forall x \in \mathcal{U}'(e) \setminus \{i\}$. The total bandwidth usage on link e in $\mathbf{b}'$ is

$$\begin{aligned} & \textstyle\sum_{x \in \mathcal{U}'(e)} b'(P'_x) \\ = \;& b'(P'_i) + \sum_{x \in \tilde{\mathcal{U}}_{-i}(e)} b'(P'_x) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b'(P'_x) \\ \leq \;& (|\tilde{\mathcal{U}}_{-i}(e)| + 1) b'(P'_i) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b'(P'_x) & (5.9) \\ < \;& (|\tilde{\mathcal{U}}_{-i}(e)| + 1) b^o(P'_i) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b_{-i}(P_x) & (5.10) \\ \leq \;& b(e), & (5.11) \end{aligned}$$

  where (5.9) follows from (**1b**), (5.10) follows from the condition $b'(P'_i) < b^o(P'_i)$ and the assumption $b'(P'_x) \leq b_{-i}(P_x)$, and (5.11) follows from (5.2) and (5.1). This contradicts the fact that $e \in \mathcal{B}'(P'_i)$, because e should be saturated in $\mathbf{b}'$ according to Property 1). This completes the proof of Part 1).
- **Part 2):** $b'(P'_j) < b_{-i}(P_j)$.
  By Property 2) of bottleneck, we know that $b'(P'_j) \geq b'(P'_x)$, $\forall x \in \mathcal{U}'(e)$. Thus it suffices to prove (**2a**). If $i \in \mathcal{U}'(e)$, we can take $k = i$ and we are done with the proof. Next, we consider the case where $i \in \mathcal{U}'(e)$. We prove (**2a**) by contradiction. Assume that $b'(P'_x) \leq b_{-i}(P_x)$, $\forall x \in \mathcal{U}'(e) \setminus \{j\}$. Note that $i \notin \mathcal{U}'(e)$ implies $\mathcal{U}_{-i}(e) = \mathcal{U}'(e)$. The total bandwidth usage on link e in $\mathbf{b}'$ is

$$b'(P'_j) + \sum_{x \in \mathcal{U}'(e) \setminus \{j\}} b'(P'_x) < b_{-i}(P_j) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \{j\}} b_{-i}(P_x) \leq b(e),$$

  where the second inequality follows from the feasibility of $\mathbf{b}_{-i}$. This contradicts the fact that $e \in \mathcal{B}'(P'_j)$. Therefore, Part 2) holds.

We have finished the proof of this lemma. ■

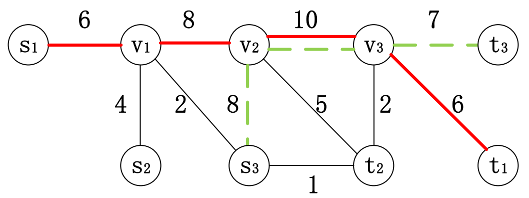
Consider the example shown in Fig. 6. Before player 2 chooses its path in Fig. 6(a), $b_{-2}(P_1) = 5$ (red solid) and $b_{-2}(P_3) = 5$ (green dashed). After player 2 chooses path P'_2 (blue dotted) in Fig. 6(b), the bandwidth of player 1's path (red solid) decreases to $b'(P'_1) = 4$. The reason is that $(v_1, v_2) \in \mathcal{B}'(P'_1)$ and $(v_1, v_2) \in P'_2$, which is corresponding to Part 2) in Lemma 5.3. We also have $b'(P'_2) = b'(P'_3) = 4$.

Lemma 5.4: Let P'_i be the new s_i-t_i path chosen by player i based on observed available bandwidth. We have

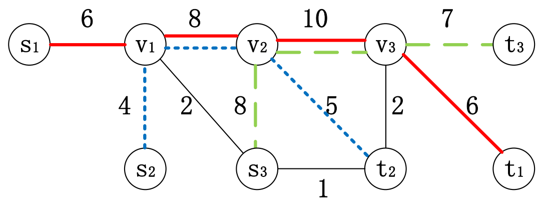
1) Let $e \in \mathcal{B}^o(P'_i)$ be an observed bottleneck of path P'_i. If $b'(P'_i) > b^o(P'_i)$, then $\exists k \in \mathcal{U}'(e) \setminus \{i\}$, such that (**1a**) $b'(P'_k) < b_{-i}(P_k)$ and (**1b**) $b'(P'_i) > b'(P'_k)$.
2) Let $e \in \mathcal{B}_{-i}(P_j)$ be a bottleneck of path P_j in $\mathbf{b}_{-i}$. If $b'(P'_j) > b_{-i}(P_j)$, then $\exists k \in \mathcal{U}_{-i}(e) \setminus \{j\}$, such that (**2a**) $b'(P'_k) < b_{-i}(P_k)$ and (**2b**) $b'(P'_j) > b'(P'_k)$. □

Proof. We prove Part 1) and 2) separately:

- **Part 1):** $b'(P'_i) > b^o(P'_i)$.



(a) Before player 2 chooses its path



(b) After player 2 chooses its path

Fig. 6. An example for Lemma 5.3 and Lemma 5.4.

We prove this by contradiction. Assuming to the contrary that $b'(P'_x) \geq b_{-i}(P_x)$ or $b'(P'_x) \geq b'(P'_i)$, $\forall x \in \mathcal{U}'(e) \setminus \{i\}$, we have the following claims:
(1) *For all $x \in \tilde{\mathcal{U}}_{-i}(e)$, we have $b'(P'_x) \geq b^o(P'_i)$.*
If $b'(P'_x) \geq b_{-i}(P_x)$, we have

$$b'(P'_x) \geq b_{-i}(P_x) \geq b^o(e) = b^o(P'_i),$$

where the second inequality follows from Lemma 5.2 and the equality follows from the fact that $e \in b'(P'_x)$. If $b'(P'_x) \geq b'(P'_i)$, we have $b'(P'_x) \geq b'(P'_i) > b^o(P'_i)$.
(2) *For all $x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)$, we have $b'(P'_x) \geq b_{-i}(P_x)$.*
If $b'(P'_x) \geq b'(P'_i)$, we have

$$b'(P'_x) \geq b'(P'_i) > b^o(P'_i) = b^o(e) > b_{-i}(P_x),$$

where the last inequality follows from Lemma 5.2.
Note that $\mathcal{U}'(e) = \mathcal{U}_{-i}(e) \cup \{i\}$. The total bandwidth usage on link e in $\mathbf{b}'$ is

$$\begin{aligned} \sum_{x \in \mathcal{U}'(e)} b'(P'_x) &= b'(P'_i) + \sum_{x \in \tilde{\mathcal{U}}_{-i}(e)} b'(P'_x) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b'(P'_x) \\ &> (|\tilde{\mathcal{U}}_{-i}(e)| + 1) b^o(P'_i) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b_{-i}(P_x) \quad (5.12) \\ &= (|\tilde{\mathcal{U}}_{-i}(e)| + 1) b^o(e) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \tilde{\mathcal{U}}_{-i}(e)} b_{-i}(P_x) \\ &= b(e), \quad (5.13) \end{aligned}$$

where (5.12) follows from the condition of Part 1) and two claims, and (5.13) follows from (5.1). This contradicts the feasibility of $\mathbf{b}'$. Thus we have proved Part 1).

- **Part 2):** $b'(P'_j) > b_{-i}(P_j)$.
We prove this by contradiction. Assume that $b'(P'_x) \geq b_{-i}(P_x)$ or $b'(P'_x) \geq b'(P'_j)$, $\forall x \in \mathcal{U}_{-i}(e) \setminus \{j\}$. If $b'(P'_x) \geq b'(P'_j)$, we have

$$b'(P'_x) \geq b'(P'_j) > b_{-i}(P_j) \geq b_{-i}(P_x),$$

where we used the condition of Part 2) and $e \in \mathcal{B}_{-i}(P_j)$. Thus we have $b'(P'_x) \geq b_{-i}(P_x)$, $\forall x \in \mathcal{U}_{-i}(e) \setminus \{j\}$. Then, considering the fact that $\mathcal{U}_{-i}(e) \subseteq \mathcal{U}'(e)$, the total bandwidth usage on link e in $\mathbf{b}'$ is

$$b'(P'_j) + \sum_{x \in \mathcal{U}'(e) \setminus \{j\}} b'(P'_x) > b_{-i}(P_j) + \sum_{x \in \mathcal{U}_{-i}(e) \setminus \{j\}} b_{-i}(P_x) = b(e),$$

where the equality follows from the fact that $e \in \mathcal{B}_{-i}(P_j)$. This violates the feasibility of $\mathbf{b}'$. We have proved Part 2). ■

In Fig. 6, the bandwidth of player 3's path (green dashed) increases from $b_{-2}(P_3) = 5$ to $b'(P'_3) = 6$, due to the decrease of the bandwidth of player 1's path (red solid), from $b_{-2}(P_1) = 5$ to $b'(P'_1) = 4$. We also have $b'(P'_3) > b'(P'_1)$.

Based on Lemma 5.3 and Lemma 5.4, we prove in the following an important theorem, which states that the bandwidth of the new path is equal to the observed bandwidth.

Theorem 5.1: Let P'_i be the new s_i-t_i path chosen by player i based on the observed available bandwidth. Then $b'(P'_i) = b^o(P'_i)$. □

Proof. First, we prove that $b'(P'_i) \geq b^o(P'_i)$. On contrary, assume that $b'(P'_i) < b^o(P'_i)$. We will derive a contradiction. By Lemma 5.3, there must exist a player j, such that $b'(P'_j) > b_{-i}(P_j)$ and $b'(P'_i) \geq b'(P'_j)$. Then, by Lemma 5.4, there must exist a player k, such that $b'(P'_k) < b_{-i}(P_k)$ and $b'(P'_j) > b'(P'_k)$. Hence we have $b'(P'_i) \geq b'(P'_j) > b'(P'_k)$. Repeating this argument, we may end in two cases: (1) the sequence terminates at player i; (2) the sequence forms a loop. For case (1), we obtain $b'(P'_i) > b'(P'_i)$, which is obviously wrong. For case (2), it contradicts the fact that the sequence is in a non-increasing order.

Using a similar logic, we can prove that $b'(P'_i) \leq b^o(P'_i)$. This implies that $b'(P'_i) = b^o(P'_i)$. ■

Remark 5.1. As a direct consequence of Theorem 5.1, player i has an incentive to change its strategy if and only $b^o(P'_i) > b(P_i)$. Also, P'_i is the best response strategy for player i.

6. Converging to Nash Equilibrium

In this section, we present a game based algorithm, listed in Algorithm 2, to compute an NE of the MAXBAR game. The idea of the algorithm is as follows. In the initialization stage (Line 2), each player i chooses an initial s_i–t_i path regardless of other players. Without loss of generality, each player chooses a path with maximum bandwidth using algorithm MCP(G, s_i, t_i, b) [18]. Then Algorithm 2 proceeds in a round-robin fashion. At every stage, there can be only one player changing its path. Such assumption is common in game theory and essential to avoid oscillation. When a player plans to change its path, it follows these steps:

1) Extract its path from the graph (Line 5).
2) Calculate the observed available bandwidth for each link in the resulting network (Line 7).
3) Find a path with the maximum observed bandwidth (Line 8).
4) If the observed bandwidth of the new path is greater than its current bandwidth, it switches to the new path; otherwise, it keeps the same path (Line 9).

The process stops when no player can improve its bandwidth by changing to another path.

Algorithm 2 Game Based Algorithm

```
Input: Graph $G(V,E,b)$ and set $\mathcal{U}$ of players $\{1,\ldots,K\}$.
Output: A Nash Equilibrium $\mathcal{P}$.
1: $\mathcal{P} \leftarrow \emptyset$;
2: $P_i \leftarrow \mathsf{MCP}(G,s_i,t_i,b)$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$, $\forall i \in \mathcal{U}$;
3: repeat
4:   for each player $i \in \mathcal{U}$ do
5:     $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P_i\}$;
6:     $(b_{-i}(P_1),\ldots,b_{-i}(P_K)) \leftarrow ComB(G,b,\mathcal{P},\mathcal{U})$;
7:     Compute $b^o(e)$ for all $e \in E$ using (5.1);
8:     $P_i' \leftarrow \mathsf{MCP}(G,s_i,t_i,b^o)$;
9:     if $b^o(P_i') > b(P_i)$ then $P_i \leftarrow P_i'$; end if
10:    $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$;
11:    $(b(P_1),\ldots,b(P_K)) \leftarrow ComB(G,b,\mathcal{P},\mathcal{U})$;
12:  end for
13: until there is no path changed
14: return $\mathcal{P}$.
```

The correctness and the convergence speed of Algorithm 2 is captured in the following theorem.

Theorem 6.1: For every instance of the MAXBAR game, it converges to a set $\mathcal{P}$ of paths in $O((K\log K + Km)(Km)^K)$ time, where m is the number of links and K is the number of players. Moreover, $\mathcal{P}$ is an NE of the MAXBAR game. □

To prove this theorem, we need the following lemma, which shows an important property of a global bottleneck.

Lemma 6.1: Let $\mathbf{b}$ be an MFBA for instance $(G,\mathcal{P})$. If e_g is a global bottleneck, then $b(P_x) = \frac{b(e_g)}{u(e_g)}$, $\forall x \in \mathcal{U}(e_g)$. □

Proof. First, we claim that *if* $e \in \mathcal{B}(P_i)$*, then* $b(P_i) \geq \frac{b(e)}{u(e)}$. Considering both Properties 1) and 2) of bottleneck e, we have

$$b(e) = \sum_{x\in\mathcal{U}(e)} b(P_x) \leq u(e)\cdot b(P_i).$$

Thus the claim is proved. Based on this claim and the fact that e_g is a global bottleneck, we have

$$b(P_x) \geq \frac{b(e_g)}{u(e_g)}, \forall x \in \mathcal{U}(e_g). \tag{6.1}$$

Assume $\exists y \in \mathcal{U}(e_g)$ such that $b(P_y) > \frac{b(e_g)}{u(e_g)}$. The total bandwidth usage on e_g is $\sum_{x\in\mathcal{U}(e_g)} b(P_x) > b(e_g)$, contradicting the feasibility of $\mathbf{b}$. We have proved that $b(P_x) = \frac{b(e_g)}{u(e_g)}$, $\forall x \in \mathcal{U}(e_g)$. ■

Proof of Theorem 6.1: By Lemma 4.1, we conclude that every time a player changes its path, the ordering $\vec{\mathbf{b}}_l$ increases lexicographically. Now we prove an upper bound on the number of times the ordering can increase. By Lemma 6.1, we know that a global bottleneck must be equally shared by all paths using it. As a result, the number of different possible values of b_1 is bounded by $O(Km)$. For each possible value of b_1, there are at most K players whose paths correspond to this value. If the value of b_1 and the corresponding path P_x stay the same, the number of different possible values of b_2 is $O(Km)$ as well. The reason is that we can subtract b_1 from the bandwidth of each link along P_x, and remove x from the player set. This resulting graph is a smaller instance and all the lemmas still hold. Repeating this analysis for all the coordinates, we conclude that the number of times that the lexicographic ordering can increase is bounded by $O((Km)^K)$. We assume that the network is connected, which implies that $m \geq n-1$. Then the time complexity of Algorithm 1 is $O(Km)$. Computing $b^o(e)$ for all $e \in E$ takes $O(K\log K + Km)$ time. In addition, the time complexity of $\mathsf{MCP}(G,b,\mathcal{P})$ is $O(m)$ [18]. Therefore the time complexity of Algorithm 2 is $O((K\log K + Km)(Km)^K)$. By Theorem 5.1, the returned strategy set is an NE, since no player can improve its payoff by changing its path unilaterally. ■

Remark 6.1. Our extensive experiments in Section 7 show that the MAXBAR game converges to an NE in less than 10 iterations. This indicates that our theoretical bound $O((Km)^K)$ on the number of iterations is quite conservative.

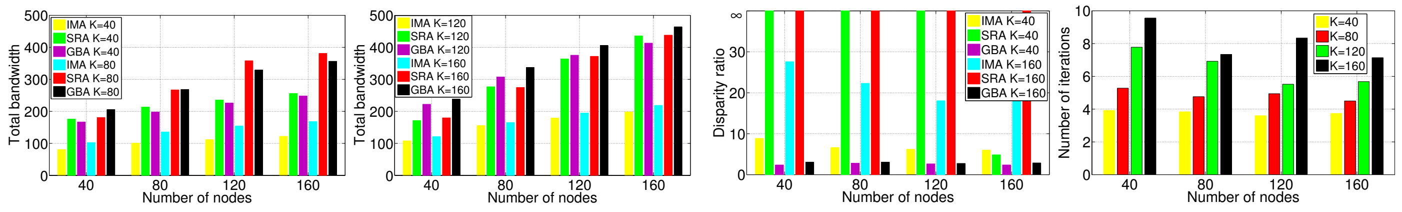
Remark 6.2. As shown in the example in Section 4, there could be more than one NE. If the initial set of strategies were different from the one computed in Line 2 of Algorithm 2, Lines 3–14 may lead to a *different* NE. However, Lines 3–14 of the algorithm will always lead to some NE.

In order to enforce the users in the network to follow the game course, we use a token-based protocol, where a token is circulated among the users–only the user with the token has the opportunity to change its path. The distributed implementation of $ComB(G,b,\mathcal{P},\mathcal{U})$ were proposed by [5, 13]. The information needed by Equation (5.1) to compute the observed available bandwidth is sent to each user by the link-state algorithm for determining the new path.

7. Numerical Results

In this section, we verify the convergence analysis and evaluate the performance of Algorithm 2 (denoted as GBA, short for Game Based Algorithm) on network topologies generated by BRITE [4]. We compared GBA with two other routing algorithms. In the first algorithm, each user acts independently and attempts to maximize its bandwidth as much as possible. This scenario is similar to anarchy and we denote it by IMA (Independent Maximization Algorithm). In the second algorithm, the bandwidth allocation for the users is done sequentially. A user is chosen randomly from the set of users that have not been allocated bandwidth. Each user chooses a widest path in the residual network, and has a bandwidth equal to that of the chosen path. This procedure is repeated until all users are considered for bandwidth allocation. This technique is similar to the Resource reSerVation Protocol (RSVP) [19], with the difference being that each user is allocated the maximum possible bandwidth in the residual network. We denote this scheme by SRA (Sequential Reservation Algorithm).

BRITE [4] is a widely used Internet topology generator. We used the Waxman model [22] with default values for $\alpha = 0.15$ and $\beta = 0.2$. According to the Waxman model, if d_{vw} denotes the Euclidean distance between two nodes v and w, the probability of having a bidirectional link (v,w) between v and w is given by $\beta \times \exp\left(\frac{d_{vw}}{\alpha\cdot L}\right)$, where L is the maximum distance between two nodes. The nodes of the graph were deployed randomly in a square region of size 1000×1000 m^2. We used four network sizes with $40, 80, 120,$ and 160 nodes



(a) Total bandwidth ($K = 40, 80$) (b) Total bandwidth ($K = 120, 160$) (c) Bandwidth disparity ratio (d) Number of iterations

Fig. 7. Numerical results

each. The corresponding number of edges was 160, 320, 480, and 640 respectively. For each network size, we used BRITE to generate 10 different connected network topologies, where the link bandwidth was drawn from a uniform distribution in the range $[1, 10]$. We used 4 different values of K: 40, 80, 120, and 160. For each value of K and a topology, we randomly generated 5 test cases by randomly choosing the source-destination pairs for the users. Therefore, we used 50 test cases for each value of K and a network size. The results reported here were the averages over these 50 test cases.

Figs. 7(a) and 7(b) show the total bandwidth obtained by IMA, SRA and GBA. We observe that GBA always outperforms IMA. This is as expected, because IMA uses less information in decision making. SRA and GBA have similar performance. An interesting observation is that GBA outperforms SRA when the user density is high and SRA performs better otherwise. When the user density is high, the system resource is sufficiently utilized. Some earlier reservations in SRA may block other users. However, sharing in GBA may bring more gain in terms of total bandwidth than SRA. When the user density is low, the blocking probability of SRA is low. In addition, GBA may cause multiple users to compete for the links with large bandwidth values, leaving links with small bandwidth values unutilized.

Fig. 7(c) shows the bandwidth disparity ratio, defined as the ratio of the highest bandwidth of a user over the lowest one. Due to space limit, we only showed the result for $K = 40$ and 160. We observe that GBA is the fairest, with a disparity ratio ranging between 2 and 3. The disparity ratio of IMA is at least twice that of GBA for $K = 40$, and is more than six times that of GBA. SRA has the worst disparity ratio, having the value of ∞ for all cases except the case with 160 nodes and 40 users. This is as expected, because some users will be blocked in SRA and therefore have zero bandwidth.

Fig. 7(d) shows that the average number of iterations of GBA, where an iteration is from Line 3 to Line 13 in Algorithm 2. We observe that the average number of iterations is no more than 10 in all cases studied, which is far less than the theoretic bound in Theorem 6.1.

To summarize, extensive experiments show that our algorithm converges to an NE rapidly and achieves very good fairness as well as total bandwidth.

8. Conclusions

In this paper, we studied the problem of non-cooperative routing under max-min fair bandwidth allocation, where each user aims to maximize its own bandwidth. We proved the existence of Nash Equilibria, where no user has an incentive to change its current path. We also quantified the effect of the selfish behavior on the total performance of the network. Then we introduced the concept of *observed available bandwidth*, which allows us to find a path with maximum bandwidth in polynomial time. It is both a key step for our main analysis and of independent interest. We next presented a game based algorithm to compute an NE and proved that the network converges to an NE if all users follow the natural game course.

References

[1] R. Banner and A. Orda; Bottleneck routing games in communication networks; *IEEE INFOCOM'2006*, pp. 1–12.

[2] J. Behrens and J.J. Garcia-Luna-Aceves; Distributed, scalable routing based on link-state vectors; *ACM SIGCOMM'1994*, pp. 136–147.

[3] D. Bertsekas and R. Gallager; *Data Networks*, Prentice Hall, 1987.

[4] BRITE; http://www.cs.bu.edu/brite/.

[5] A. Charny, D.D. Clark and R. Jain; Congestion control with explicit rate indication; *IEEE ICC'1995*, pp. 1954–1963.

[6] S. Chen and K. Nahrstedt; Maxmin fair routing in connection-oriented networks; *Euro-PDS'1998*, pp. 163–168.

[7] A. Demers; Analysis and simulation of a fair queueing algorithm; *ACM SIGCOMM'1989*, pp.1–12.

[8] D. Fudenberg and J. Tirole; *Game Theory*. MIT press, 1991.

[9] L. Gao and J. Rexford; Stable internet routing without global coordination; *ACM SIGMETRICS'2000*, pp. 307–317.

[10] P.B. Godfrey, M. Schapira, A. Zohar and S. Shenker; Incentive compatibility and dynamics of congestion control; accepted by *ACM SIGMETRICS'2010*.

[11] T.G. Griffin and G. Wilfong; A safe path vector protocol; *IEEE INFOCOM'2000*, pp. 490-499.

[12] T.G. Griffin, F.B. Shepherd and G. Wilfong; The stable paths problem and interdomain routing; *IEEE/ACM Transactions on Networking*, Vol. 10(2002), pp. 232-243.

[13] J. Jaffe; Bottleneck flow control; *IEEE Transaction on Communications*, Vol. 29(1981), pp. 954-962.

[14] E. Koutsoupias and C. Papadimitriou; Worst-case equilibria; *STACS'1999*, pp. 404–413.

[15] Q. Ma, P. Steenkiste and H. Zhang; Routing high-bandwidth traffic in max-min fair share networks; *ACM SIGCOMM'1996*, pp. 206–217.

[16] A. Mayer, Y. Ofek and M. Yung; Approximating max-min fair rates via distributed local schedulingwith partial information; *IEEE INFOCOM'1996*, pp. 928–936.

[17] D. Nace, N. Doan, E. Gourdin and B. Liau; Computing optimal max-min fair resource allocation for elastic flows; *IEEE/ACM Transactions on Networking*, Vol. 14 (2006), pp. 1272–1281.

[18] A.P. Punnen; A linear time algorithm for the maximum capacity path problem; *European Journal of Operational Research*, Vol. 53(1991), pp. 402–404.

[19] RFC: Http://tools.ietf.org/html/rfc2205

[20] M. Schapira and A. Zohar; Congestion control game; Technical Report 2008; http://leibniz.cs.huji.ac.il.

[21] M. Steenstrup; RFC1479: Inter-Domain Policy Routing Protocol Specification: Version 1; 1993.

[22] B.M. Waxman; Routing of multipoint connections; *IEEE JSAC*, Vol. 6(1988), pp. 1617–1622.