A Game Theoretic Approach to Stable Routing in Max-Min Fair Networks

Dejun Yang, *Member, IEEE*, Guoliang Xue, *Fellow, IEEE*, Xi Fang, *Member, IEEE*, Satyajayant Misra, *Member, IEEE*, and Jin Zhang, *Member, IEEE*

***Abstract*— In this paper, we present a game theoretic study of the problem of routing in networks with max-min fair congestion control at the link level. The problem is formulated as a non-cooperative game, in which each user aims to maximize its own bandwidth by selecting its routing path. We first prove the existence of Nash Equilibria. This is important, because at a Nash Equilibrium (NE), no user has any incentive to change its routing strategy—leading to a stable state. In addition, we investigate how the selfish behavior of 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, when paths of other users are fixed. 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 simulations show that the algorithm converges to an NE within 10 iterations and also achieves better fairness compared with other algorithms.**

1. Introduction

Routing is the process of selecting paths in a network along which to send data packets. In communication networks, the choice of a route between a source-destination pair has a significant bearing on the resulting bandwidth. For example, in peer-to-peer networks, there may be several pairs of peers sharing volumes of data between each other. The objective of each pair of peers, considered as a user, is to send as many packets as possible through the network while competing for network resources against other users. With this selfish objective, a user will change its path if the new path provides a larger bandwidth value even at the cost of other users. Since multiple users may compete for the bandwidth on the same link, it is necessary to have a congestion control scheme to allocate bandwidth among competing users. Hence max-min fair bandwidth allocation has been widely adopted as a congestion control scheme at the link level [6, 8, 16, 18–20, 23]. The max-min fair bandwidth allocation scheme treats 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 model the network using a directed graph, and present a game theoretic study 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. We call this problem the Maximal-Bandwidth Routing problem. Two questions arise while addressing this problem: *How can a user efficiently find a path with maximum bandwidth under max-min fair congestion control, when the paths of all other users are given?* and *Will the network oscillate or converge to a stable state?* The first question is critical to our convergence analysis, since it directly affects the convergence speed. 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 challenging. The second question is important because oscillation among different paths introduces dramatic overhead, consuming network resources. This paper answers both questions.

In answering the first question, we introduce the concept of *observed available bandwidth* and prove that it can accurately predict the bandwidth of a path. In answering 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 Nash Equilibria* and *the convergence of the game*. One major challenge arises while answering these questions. While selecting 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.

The major contributions of this paper are as follows:

- We formulate the Maximal-Bandwidth Routing problem (MAXBAR) as a non-cooperative strategic game where each player makes the routing decision selfishly to maximize its bandwidth. In Section 7, we generalize it to the case where each user has a bandwidth demand.
- We prove the existence of Nash Equilibria in the MAXBAR game, where no player has any incentive to deviate from its chosen path. We also prove a lower bound and an upper bound on the *price of anarchy* of the MAXBAR game, which is a concept quantifying the system degradation due to selfish behavior of users. As a byproduct, this also gives an approximation to the social optimal solution to the MAXBAR problem.
- We introduce a novel concept of observed available bandwidth to compute the available bandwidth on each link. It empowers the efficient computation of the best response strategy for each user. This is non-trivial, as the traditional widest path algorithm cannot be directly

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 W911NF-09-1-0467 and NSF grants 0901451 and 1115129. The information reported here does not reflect the position or the policy of the federal government. This is an extended and enhanced version of the paper [26] that appeared in IEEE ICNP 2010.

applied due to the mutual influence between paths sharing common links [18].
- We investigate the behavior 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 this paper is organized as follows. In Section 2, we present a brief overview of related work. In Section 3, we describe our system model, present the MAXBAR problem where each user would like to have as much bandwidth as possible, 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, we study a generalization of the MAXBAR problem where each user has a bandwidth demand, instead of aiming to have as much bandwidth as possible. In Section 8, we present numerical results on randomly generated networks. These results show that the game converges to an NE rapidly (within 7 iterations on average and 10 iterations at worst) and achieves better fairness compared with other algorithms. We conclude this paper in Section 9.

2. **Related Work**

Congestion control is a critical task in communication networks to address the issue of fairly and optimally allocating resources, bandwidth in particular, among multiple competing users. Max-min fair bandwidth allocation has been proposed as one of the congestion control schemes [3, 16]. This scheme was first presented in [16]. The author also proved the optimality and the uniqueness of the allocation. In [8], Demers proposed a fair queuing scheduler, which is employed on each gateway, to implement a max-min fair network. In [18], Ma *et al.* studied how to route in max-min fair networks to improve the total throughput of the network. To calculate the max-min fair bandwidth for each path, they also presented a centralized algorithm. Note that the information used by the routing algorithm is abstract and only an estimate of the accurate available bandwidth. Showing that computing the max-min fair bandwidth requires global information, Mayer *et al.* [19] designed a local distributed scheduling algorithm to approximate max-min fair bandwidth allocation.

Chen and Nahrstedt [6] extended the concept of max-min fairness to the routing level, since the max-min fair bandwidth allocation scheme was proposed to achieve fairness at link level. They defined the *fairness-throughput* and introduced 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. They also proposed a max-min fair routing algorithm to select a path for the new user to maximize the minimum bandwidth allocated to all users. In [20], Nace considered a model, where the routing is splittable, and gave a linear programming based algorithm to compute the optimal max-min fair bandwidth allocation. Schapira *et al.* [23] and Godfrey *et al.* [13] 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 [8, 16, 19] or the case where routing aims to improve the total performance [6, 13, 18, 20, 23]. 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. The game formulation of this scenario falls into the category of *bottleneck game* [1]. There are also important works on stable routing in the literature [12, 14, 15]. However, these works do not consider max-min fair bandwidth allocation in their models.

3. **System Model and Problem Formulation**

We first describe the network model and discuss the well known max-min fair congestion control scheme. We then formulate the problem studied in this paper.

A. Network Model

We model the network by a directed edge-weighted 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 $\mathcal{U} = \{1, 2, \dots, N\}$ of *users*. User $i \in \mathcal{U}$ needs to transmit packets from a source node $s_i \in V$ to a destination node $t_i \in V$ over an s_i–t_i path. An s–t path in the network consists of an ordered sequence of vertices $s{=}v_0,\ v_1,\ \dots,\ v_q{=}t$, where $(v_l, v_{l+1}) \in E$ for $0 \le l < q$. We denote such a path by v_0-v_1-$\cdots$-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 uses only one path, which is denoted by P_i. We denote the set of paths currently used by the users as $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$. We denote the set of users currently sharing link e by $\mathcal{U}_e(\mathcal{P})$, i.e., $\mathcal{U}_e(\mathcal{P}) = \{i | i \in \mathcal{U} \text{ and } e \in P_i\}$.

For routing approach, we will use link-state source routing algorithms as in [18]. In such routing schemes, each node knows the network topology and the state information on each link [2, 24]. Thus it is possible for the node to select its path. In this paper, we consider best-effort flows [18] and assume that every 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 use only less than the equal share (due to some bottlenecks), while some can use more. Hence, 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) [8, 16] is used for congestion control. Max-min fair bandwidth allocation has been recognized as the optimal throughput-fairness definition [16, 19]. Intuitively, if there are multiple users sharing a common link, each user will get a "fair share" of the link's bandwidth. If some user cannot use up its fair share bandwidth because it has a lower share assigned on another link, the excess bandwidth is "fairly" split among all other users of this link. 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 user i in a max-min fair network by $b_i(\mathcal{P})$ (how to compute the value of $b_i(\mathcal{P})$ will be shown later). Since user i will use only one path at any given time, we will say *the bandwidth of user i* instead of *the bandwidth of user i's path* when the path is clear from the context. We use $\mathbf{b}(\mathcal{P}) = (b_1(\mathcal{P}), b_2(\mathcal{P}), \dots, b_N(\mathcal{P}))$ to denote the Max-min Fair Bandwidth Allocation (MFBA) given users' paths $\mathcal{P}$. The uniqueness of MFBA has been proved in [23]. 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 $\mathcal{B}_i(\mathcal{P})$ 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) $\sum_{j \in \mathcal{U}_e(\mathcal{P})} b_j(\mathcal{P}) = b(e)$,
2) $b_i(\mathcal{P}) \geq b_j(\mathcal{P}), \forall j \in \mathcal{U}_e(\mathcal{P})$.

Property 1) means that link e is saturated. We call a link *saturated* if its bandwidth is fully allocated. This property is obvious as otherwise e is not a link that keeps P_i from obtaining more bandwidth. 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 in Lemma 3 of [6] and Lemma 3 of [16].

Algorithms for calculating the bandwidth allocation for each path in a max-min fair network have been proposed in [16, 18]. 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 [16, 18].

The basic idea of Algorithm 1 is that in each iteration, we find a *global bottleneck* $\bar{e}$, which is defined as the link having the least equal share, i.e., $\bar{e} = \arg\min_{e \in E} \frac{b(e)}{|\mathcal{U}_e(\mathcal{P})|}$. We allocate the equal share of $b(\bar{e})$ to all users in $\mathcal{U}_{\bar{e}}(\mathcal{P})$. Then all the paths of users in $\mathcal{U}_{\bar{e}}(\mathcal{P})$ are removed from the network. The link bandwidths are reduced by the bandwidth consumed by the 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 $\bar{e}$ selected in the first iteration and $\frac{b(v_4,t_2)}{|\mathcal{U}_{(v_4,t_2)}(\mathcal{P})|} = 3$. Since user 2 (blue dotted) is the only one using link (v_4, t_2), we set $b_2(\mathcal{P}) = 3$, remove path P_2 from the network and

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

input : Network G, path set $\mathcal{P}$ and user set $\mathcal{U}$
output: $b_i(\mathcal{P})$ for all $i \in \mathcal{U}$

- $b_i(\mathcal{P}) \leftarrow 0, \forall i \in \mathcal{U}$;
- **repeat**
  - Let $\bar{e} := argmin_{e \in E} \frac{b(e)}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}$ in $G(V, E, b)$;
  - $b_{temp} \leftarrow \frac{b(\bar{e})}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}$;
  - **foreach** *player* $i \in \mathcal{U}_{\bar{e}}(\mathcal{P})$ **do**
    - $b_i(\mathcal{P}) \leftarrow b_{temp}$;
    - **foreach** $e \in P_i$ **do**
      - $b(e) \leftarrow b(e) - b_i(\mathcal{P})$;
      - **if** $b(e) = 0$ **then** $E \leftarrow E \setminus \{e\}$;
    - **end**
    - $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P_i\}$;
  - **end**
- **until** $\mathcal{P} = \emptyset$;
- **return** $b_i(\mathcal{P})$ for all $i \in \mathcal{U}$;

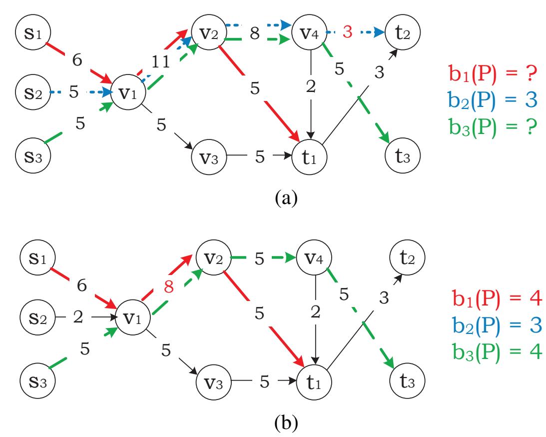


Fig. 1. An example with 3 users. $P_1 = s_1$-v_1-v_2-t_1 (red solid), $P_2 = s_2$-v_1-v_2-v_4-t_2 (blue dotted), and $P_3 = s_3$-v_1-v_2-v_4-t_3 (green dashed).

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 $\bar{e}$. There are two paths, P_1 (red solid) and P_3 (green dashed), sharing link (v_1, v_2). Each of them obtains bandwidth $\frac{b(v_1,v_2)}{|\mathcal{U}_{(v_1,v_2)}(\mathcal{P})|} = 4$. We set $b_1(\mathcal{P}) = b_3(\mathcal{P}) = 4$, and remove path P_1 and path P_3. Since there is no more paths left, the algorithm terminates.

C. Problem Formulation

In this paper, we study the problem of routing in a max-min fair network with multiple *selfish* users, where each user selects its path to maximize its bandwidth. We call this problem the ***MAX**imal-**BA**ndwidth **R**outing* (MAXBAR) problem. We are interested in the following questions:

Q1. How does each user select the path to maximally increase its bandwidth?

Q2. Will the routing oscillate forever or converge to a stable state, where no user can increase its bandwidth by unilaterally changing its path?

Q3. If the answer to Q2 is converging to a stable state, how is the social welfare in the stable state compared to that in the optimal solution with centralized control?

The MAXBAR problem can be formulated as a non-cooperative game, called MAXBAR game, as follows. Each user is a *player* in this game. We define the *strategy* of player i as its path P_i. A strategy profile of all players is then $\mathcal{P}$. We denote the strategies except player i's by $\mathcal{P}_{-i}$. We define the *utility* of player i as the bandwidth $b_i(\mathcal{P})$ of path P_i. Since players are selfish but rational, each player makes independent routing decisions to maximize its own utility. When player i's path is not in the network, we use $\mathbf{b}(\mathcal{P}_{-i})$ to denote the MFBA and use $b_j(\mathcal{P}_{-i})$ to denote the bandwidth of path P_j, where we assume $b_i(\mathcal{P}_{-i}) = 0$ as a technical convention. Let $\mathcal{P}|^i P'_i$ denote the path profile where player i changes its path to P'_i and others remain the same. When the context is clear, we use $\mathcal{P}|^i$ instead of $\mathcal{P}|^i P'_i$ for notational simplicity. Let $\mathbf{b}(\mathcal{P}|^i)$ denote the MFBA and $b_j(\mathcal{P}|^i)$ denote the new bandwidth of user j's path. It is clear that $\mathcal{U}_e(\mathcal{P}|^i) = \mathcal{U}_e(\mathcal{P}_{-i}) \cup \{i\}$ if $e \in P'_i$ and $\mathcal{U}_e(\mathcal{P}|^i) = \mathcal{U}_e(\mathcal{P}_{-i})$ otherwise.

An important subproblem of the MAXBAR problem, which is of independent interest, is how to select a path 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 $P_1, \cdots, P_{i-1}, P_{i+1}, \cdots, P_N$, the *best response routing* for user i is a path P_i such that $b_i(\mathcal{P})$ is maximized over all s_i–t_i paths. □

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 the routing is known has not been solved yet. This problem was also studied in [18]. However, the authors only gave estimated information for each link and their algorithm is approximate. 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 [11].

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

$$b_i(\mathcal{P}^{ne}) \geq b_i(\mathcal{P}^{ne}|^i P'_i)$$

for every strategy P'_i, where P'_i is an s_i–t_i path. □

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

The *social optimum* in the MAXBAR game is a strategy profile $\mathcal{P}^*$ such that the total utility, i.e. $\sum_{i \in \mathcal{U}} b_i(\mathcal{P}^*)$, is maximized among all $\mathcal{P}$. We use the concept of price of anarchy defined in [17] 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 utility achieved in a worst possible NE over that of the social optimum. □

Table I lists frequently used notations.

4. Existence of Nash Equilibria

As a crucial step in proving the existence of NE, we show that every time a player changes its path, the minimum bandwidth of the players, whose bandwidths change, increases strictly.

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Description
G	graph representing the network
V, E	node set, link set
v, w	node
$e, \bar{e}$	link and global bottleneck
$b(e)$	bandwidth of link e
$\mathcal{U}$, N	user (player) set, number of users (players)
i, j, k, u	user (player)
s_i, t_i	source node and destination node of user i
P_i	path (strategy) of user i
$\mathcal{P}$	path (strategy) set of users
$\mathcal{P}_{-i}$	path (strategy) set of users except i
$\mathcal{P}\|^i P'_i$	path (strategy) set with user i's path changed to P'_i
$\mathcal{P}\|^i$	abbreviation of $\mathcal{P}\|^i P'_i$ when P'_i is clear from the context
$\mathcal{U}_e(\mathcal{P})$	set of users whose paths share link e for given $\mathcal{P}$
$b_i(\mathcal{P})$	bandwidth (utility) of user i for given $\mathcal{P}$
$\mathbf{b}(\mathcal{P})$	bandwidth (utility) vector of all users for given $\mathcal{P}$

Lemma 4.1: Assume that player i unilaterally changes its path from P_i to P'_i, such that $b_i(\mathcal{P}) < b_i(\mathcal{P}|^i)$. We have $\min_{j \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b_j(\mathcal{P}|^i) > \min_{j \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b_j(\mathcal{P})$, where $\mathcal{U}_= = \{j \in \mathcal{U} | b_j(\mathcal{P}) = b_j(\mathcal{P}|^i)\}$, $\mathcal{U}_\uparrow = \{j \in \mathcal{U} | b_j(\mathcal{P}) < b_j(\mathcal{P}|^i)\}$ and $\mathcal{U}_\downarrow = \{j \in \mathcal{U} | b_j(\mathcal{P}) > b_j(\mathcal{P}|^i)\}$. □

Proof. It is clear that $i \in \mathcal{U}_\uparrow$, since $b_i(\mathcal{P}) < b_i(\mathcal{P}|^i)$. First we claim that, *for any* $j \in \mathcal{U}_\downarrow$*, there exists* $k \in \mathcal{U}_\uparrow$*, such that* $b_j(\mathcal{P}|^i) \geq b_k(\mathcal{P}|^i)$. Let $e \in \mathcal{B}_j(\mathcal{P}|^i)$ be a bottleneck of P_j after player i changes its path. By Property 2) of bottleneck, we have $b_j(\mathcal{P}|^i) \geq b_k(\mathcal{P}|^i), \forall k \in \mathcal{U}_e(\mathcal{P}|^i)$. Therefore, we only need to prove that there exists a player $k \in \mathcal{U}_e(\mathcal{P}|^i) \cap \mathcal{U}_\uparrow$. If $i \in \mathcal{U}_e(\mathcal{P}|^i)$, then we can take $k = i$. Next, we consider the case where $i \notin \mathcal{U}_e(\mathcal{P}|^i)$. Note that $\mathcal{U}_e(\mathcal{P}) \setminus \{i\} = \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$, since only player i changes its path. Therefore $i \notin \mathcal{U}_e(\mathcal{P}|^i)$ implies that $\mathcal{U}_e(\mathcal{P}|^i) \subseteq \mathcal{U}_e(\mathcal{P})$. Assuming to the contrary that $b_k(\mathcal{P}) \geq b_k(\mathcal{P}|^i), \forall k \in \mathcal{U}_e(\mathcal{P}|^i)$, the total bandwidth usage on link e in $\mathbf{b}(\mathcal{P}|^i)$ is

$$b_j(\mathcal{P}|^i) + \sum_{k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}} b_k(\mathcal{P}|^i) < b_j(\mathcal{P}) + \sum_{k \in \mathcal{U}_e(\mathcal{P}) \setminus \{j\}} b_k(\mathcal{P}) \leq b(e),$$

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

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

$$b_j(\mathcal{P}) > b_j(\mathcal{P}|^i) \geq b_k(\mathcal{P}|^i) > b_k(\mathcal{P}). \tag{4.1}$$

Following inner pair of (4.1), we know that

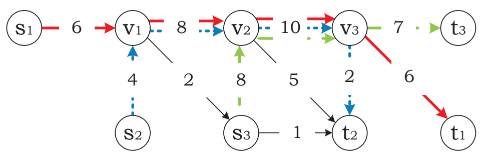
$$\min_{j \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b_j(\mathcal{P}|^i) = b_{k_1}(\mathcal{P}|^i)$$

for some player $k_1 \in \mathcal{U}_\uparrow$. Following outer pair of (4.1), we know that

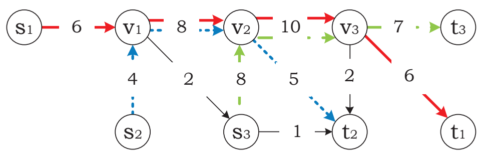
$$\min_{j \in \mathcal{U}_\downarrow \cup \mathcal{U}_\uparrow} b_j(\mathcal{P}) = b_{k_2}(\mathcal{P})$$

for some player $k_2 \in \mathcal{U}_\uparrow$. Since $k_1 \in \mathcal{U}_\uparrow$, we know that $b_{k_1}(\mathcal{P}|^i) > b_{k_1}(\mathcal{P}) \geq b_{k_2}(\mathcal{P})$. Hence this lemma holds. ■

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



(a) Before player 2 changes its path



(b) After player 2 changes its path

Fig. 2. An example for Lemma 4.1.

change, $b_1(\mathcal{P}) = 4$, $b_2(\mathcal{P}) = 2$, and $b_3(\mathcal{P}) = 4$. After the change, $b_1(\mathcal{P}|^2) = 4$, $b_2(\mathcal{P}|^2) = 4$, and $b_3(\mathcal{P}|^2) = 6$. In this example, $\mathcal{U}_= = \{1\}$, $\mathcal{U}_\uparrow = \{2, 3\}$, and $\mathcal{U}_\downarrow = \emptyset$. We have $\min\{b_2(\mathcal{P}|^2), b_3(\mathcal{P}|^2)\} > \min\{b_2(\mathcal{P}), b_3(\mathcal{P})\}$.

We now prove the existence of NE in the MAXBAR game.

Theorem 4.1: There exists at least one NE in the MAXBAR game. □

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 $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_N)$. In this vector, the minimum bandwidth b_1 is at the most significant coordinate. We have $b_\kappa \leq b_{\kappa+1}$ for $1 \leq \kappa < N$. For any two vectors $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_N)$ and $\vec{\mathbf{b}}'_l = (b'_1, b'_2, \ldots, b'_N)$, $\vec{\mathbf{b}}_l < \vec{\mathbf{b}}'_l$ in lexicographic order if and only if:

1) $b_1 < b'_1$, or
2) $\exists\ 1 < \tau \leq N$ s.t. $b_\kappa = b'_\kappa$ for $1 \leq \kappa < \tau$ 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 strategy profiles is finite as well. As each strategy profile corresponds to one vector, we pick the one corresponding to the largest vector as the strategies for the players. We conclude that such strategy profile is an NE as no player can improve its utility by unilaterally changing its strategy. ■

While we know the existence of NE, there are still 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, $\frac{1}{N} \leq$ POA $\leq \frac{2}{N}$. □

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

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

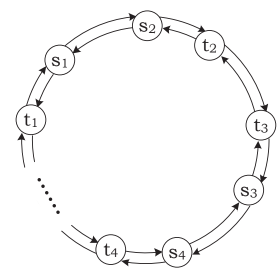
Proof. Let $\mathcal{P}^{ne} = \{P_1^{ne}, P_2^{ne}, \ldots, P_N^{ne}\}$ be any NE of the MAXBAR game. Let $\mathcal{P}^* = \{P_1^*, P_2^*, \ldots, P_N^*\}$ be the social optimum. We first claim that $b_i(\mathcal{P}^{ne}) \geq \frac{b_i(\mathcal{P}^*)}{N}$ *for any player* i*, where* $b_i(\mathcal{P}^*)$ *is the bandwidth of* P_i^* *in the social optimum.* Since $\mathcal{P}^{ne}$ is an NE, no player has any incentive to change its path, i.e.,

$$b_i(\mathcal{P}^{ne}) \geq b_i(\mathcal{P}|^i P_i^*) \geq \frac{b(e^*)}{N}, \tag{4.2}$$

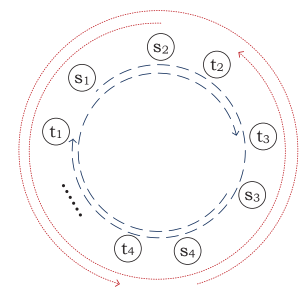
where e^* is a bottleneck of P_i^* after player i unilaterally changes its path from P_i^{ne} to P_i^*. The second inequality follows from the fact that each link can be shared by at most N players. In the social optimum, we have $b_i(\mathcal{P}^*) \leq b(e)$ for any $e \in P_i^*$. Plugging it into (4.2), we proved our claim. Based on the claim, the total utility is

$$\textstyle\sum_{i \in \mathcal{U}} b_i(\mathcal{P}^{ne}) \geq \frac{\sum_{i \in \mathcal{U}} b_i(\mathcal{P}^*)}{N} = \frac{b(OPT)}{N} \tag{4.3}$$

for any NE, where $b(OPT)$ is the total bandwidth in the social optimum. Since (4.3) holds for any NE, we have POA $\geq \frac{1}{N}$. ■



(a) Network topology



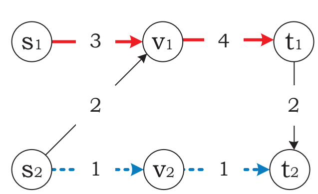
(b) Players' strategies

Fig. 3. An example where the POA is $\frac{2}{N}$.

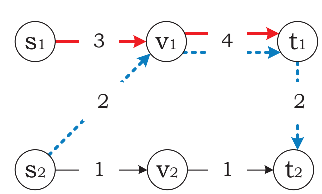
Lemma 4.3: For the MAXBAR game, POA $\leq \frac{2}{N}$. □

Proof. We prove this lemma with the help of an example. Fig. 3 depicts (partly) a network with N players. In this network, the bandwidth of each link is 1. As shown in Fig. 3(a), all the source-destination pairs with odd indices are located counterclockwise on a ring topology, while those with even indices are located clockwise. 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 with odd index (resp. even index), the clockwise (resp. counterclockwise) path s_i-s_{i+1}-t_{i+1}-…-s_N-t_N-t_1-s_1-…-t_i and the counterclockwise (resp. clockwise) path s_i-t_i. As shown in Fig. 3(b), if each player i with odd index chooses the clockwise s_i–t_i path and each player i with even index chooses the counterclockwise s_i–t_i path, the resulting strategy profile is an NE with $b_i(\mathcal{P}) = \frac{2}{N}$ for each player i. Because if any player i deviates from the current strategy and chooses the clockwise s_i–t_i path, it results in the bandwidth $\frac{2}{N+2}$. The total utility in this NE is 2.

Next, we consider the social optimum, where players with odd indices choose the counterclockwise paths and players with even indices choose the clockwise paths. The total utility is N. Hence the POA of the MAXBAR game is at most $2/N$. ■



(a) Social optimum



(b) Nash Equilibrium

Fig. 4. A social optimum is not necessarily an NE.

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

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

Remark 4.3. We do not know whether the bounds for the POA are tight. Either proving the tightness of these bounds or deriving tighter bounds is a topic for future research.

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 any 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 utility. However, the utility of the chosen path depends on other players' strategies due to the competition among players sharing links with this chosen path. Obviously, the player can try all its strategies and pick the one giving it the maximum utility. However, this 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 $O(Nm + N \log N)$ 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 an incentive to change its strategy if and only if the utility corresponding to its best response strategy is larger than that corresponding to 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.

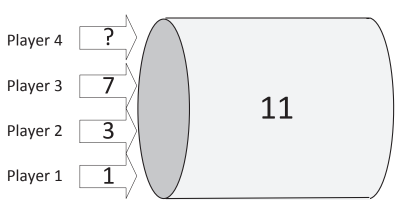


Fig. 5. A link e with max-min fair bandwidth allocation, where there are three players before player 4 joins.

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 $\mathcal{U}_e(\mathcal{P}_{-i}) = \{1, 2, 3\}$ and $b(e) = 11$. Also, $b_1(\mathcal{P}_{-i}) = 1$, $b_2(\mathcal{P}_{-i}) = 3$, and $b_3(\mathcal{P}_{-i}) = 7$. After player i joins, it is clear that player 1 would not lose its bandwidth share, since it has less than the equal share, i.e., $b_1(\mathcal{P}_{-i}) = 1 < \frac{11}{4}$. 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 of 7 and get bandwidth of $\frac{7}{2}$ each.

To capture the process we conducted above, we introduce the concept of *observed available bandwidth*. Assume that all players except i have their paths chosen. Now player i needs to find a path with maximum bandwidth in the current network. For any link e and player $j \in \mathcal{U}_e(\mathcal{P}_{-i})$, let

$$\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j) = \{k | k \in \mathcal{U}_e(\mathcal{P}_{-i}) \text{ and } b_k(\mathcal{P}_{-i}) < b_j(\mathcal{P}_{-i})\}$$

denote the set of players who are using less bandwidth than player j on link e. Let

$$\tilde{\mathcal{U}}_e(\mathcal{P}_{-i}) = \{j | j \in \mathcal{U}_e(\mathcal{P}_{-i}) \text{ and } b_j(\mathcal{P}_{-i}) \geq \frac{b(e) - \sum_{k \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)} b_k(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - |\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)| + 1}\}$$

denote the set of players such that for any player j in this set, the new bandwidth $b_j(\mathcal{P}|^i)$ is at least as large as the bandwidth of the new path P'_i of player i. The *observed available bandwidth* $b^o(e)$ of link $e \in E$ is

$$b^o(e) = \frac{b(e) - \sum_{j \in \mathcal{U}_e(\mathcal{P}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}_{-i})}{|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1}. \tag{5.1}$$

If we first sort the paths according to their bandwidth values, then for each link e we can compute $\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, 1), \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, 2), \ldots, \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, N)$, and $\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$ in $O(N)$ additional time. Thus we can compute $b^o(e)$ for all links $e \in E$ in $O(Nm + N \log N)$ time. Accordingly, the *observed bandwidth* of the new path P'_i is

$$b^o_i(\mathcal{P}|^i) = \min_{e \in P'_i} b^o(e), \tag{5.2}$$

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

$$\mathcal{B}^o_i(\mathcal{P}|^i) = \arg\min_{e \in P'_i} b^o(e).$$

Considering the example in Fig. 5, we have $\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, 1) = \emptyset$, $\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, 2) = \{1\}$, and $\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, 3) = \{1, 2\}$. The set $\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$ 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 four lemmas, which will be used in later proofs in the rest of this section.

Lemma 5.1: Assume that $j \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. For all $u \in \mathcal{U}_e(\mathcal{P}_{-i})$, $b_u(\mathcal{P}_{-i}) \geq b_j(\mathcal{P}_{-i})$ implies $u \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. □

Proof. It is obvious that if $b_u(\mathcal{P}_{-i}) = b_j(\mathcal{P}_{-i})$, then $u \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. Next, we prove that if $b_u(\mathcal{P}_{-i}) > b_j(\mathcal{P}_{-i})$, then $u \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. Let $\mathcal{K} = \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, u) \setminus \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)$. We have

$$\begin{aligned}
& b_u(\mathcal{P}_{-i}) - \frac{b(e) - \sum_{k \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, u)} b_k(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - |\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, u)| + 1} \\
&= b_u(\mathcal{P}_{-i}) - \frac{b(e) - \sum_{k \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)} b_k(\mathcal{P}_{-i}) - \sum_{k \in \mathcal{K}} b_k(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - (|\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)| + |\mathcal{K}|) + 1} \\
&> b_j(\mathcal{P}_{-i}) - \frac{b(e) - \sum_{k \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)} b_k(\mathcal{P}_{-i}) - |\mathcal{K}| b_j(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - (|\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)| + |\mathcal{K}|) + 1} \quad (5.3) \\
&\geq 0, \quad (5.4)
\end{aligned}$$

where (5.3) follows from the fact that $k \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, j)$ implies $b_k(\mathcal{P}_{-i}) \geq b_j(\mathcal{P}_{-i})$, and (5.4) follows from the fact that $j \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. Hence we have $u \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. ■

Lemma 5.2: If $j \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$, then $b_j(\mathcal{P}_{-i}) \geq b^o(e)$. If $j \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$, then $b_j(\mathcal{P}_{-i}) < b^o(e)$. □

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

$$
\begin{aligned}
b_x(\mathcal{P}_{-i}) &\geq \frac{b(e) - \sum_{j \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x)} b_j(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - |\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x)| + 1} &(5.5)\\
&= \frac{b(e) - \sum_{j \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}_{-i})}{|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1}\\
&= b^o(e). &(5.6)
\end{aligned}
$$

Therefore $b_x(\mathcal{P}_{-i}) \geq b^o(e)$. This implies the first part of the lemma, since $b_j(\mathcal{P}_{-i}) \geq b_x(\mathcal{P}_{-i})$ for any $j \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$.

Next, we prove the second part of the lemma. If $j \notin \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$, we know that $b_j(\mathcal{P}_{-i}) < b_x(\mathcal{P}_{-i})$. Now assume that y is the player whose path has the maximum bandwidth in $\mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. Then, we have $b_j(\mathcal{P}_{-i}) = b_y(\mathcal{P}_{-i})$, $\forall j \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x) \setminus \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)$. Let $\mathcal{J} = \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x) \setminus \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)$. We have

$$
\begin{aligned}
& b_y(\mathcal{P}_{-i}) - b^o(e)\\
= \;& b_y(\mathcal{P}_{-i}) - \frac{b(e) - \sum_{j \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x)} b_j(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - |\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, x)| + 1} &(5.7)\\
= \;& b_y(\mathcal{P}_{-i}) - \frac{b(e) - (\sum_{j \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)} b_j(\mathcal{P}_{-i}) + \sum_{j \in \mathcal{J}} b_j(\mathcal{P}_{-i}))}{|\mathcal{U}_e(\mathcal{P}_{-i})| - (|\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)| + |\mathcal{J}|) + 1}\\
= \;& b_y(\mathcal{P}_{-i}) - \frac{b(e) - \sum_{j \in \hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)} b_j(\mathcal{P}_{-i}) - |\mathcal{J}| b_y(\mathcal{P}_{-i})}{|\mathcal{U}_e(\mathcal{P}_{-i})| - (|\hat{\mathcal{U}}_e(\mathcal{P}_{-i}, y)| + |\mathcal{J}|) + 1}\\
< \;& 0, &(5.8)
\end{aligned}
$$

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

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 the maximum observed bandwidth and reallocate the bandwidth for each path using Algorithm 1, the new allocated bandwidth of the path is equal to its observed bandwidth.

We use proof by contradiction. 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 one 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 the observed available bandwidth. We have the following:

1) If $b_i(\mathcal{P}|^i) < b_i^o(\mathcal{P}|^i)$, then $\exists k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$, such that **1a**) $b_k(\mathcal{P}|^i) > b_k(\mathcal{P}_{-i})$ and **1b**) $b_k(\mathcal{P}|^i) \leq b_i(\mathcal{P}|^i)$, where $e \in \mathcal{B}_i(\mathcal{P}|^i)$ is a bottleneck of path P_i'.
2) If $b_j(\mathcal{P}|^i) < b_j(\mathcal{P}_{-i})$ for some $j \in \mathcal{U}$, then $\exists k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}$, such that **2a**) $b_k(\mathcal{P}|^i) > b_k(\mathcal{P}_{-i})$ and **2b**) $b_k(\mathcal{P}|^i) \leq b_j(\mathcal{P}|^i)$, where $e \in \mathcal{B}_j(\mathcal{P}|^i)$ is a bottleneck of path P_j after player i changes its path. □

Proof. We prove 1) and 2) separately:

We first prove 1). Assume that $b_i(\mathcal{P}|^i) < b_i^o(\mathcal{P}|^i)$.

By Property 2) of bottleneck, we know that $b_i(\mathcal{P}|^i) \geq b_j(\mathcal{P}|^i)$, $\forall j \in \mathcal{U}_e(\mathcal{P}|^i)$. Thus it suffices to prove that $\exists k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$, such that **1a**) holds. We prove this by contradiction. Assume that $b_j(\mathcal{P}|^i) \leq b_j(\mathcal{P}_{-i})$, $\forall j \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$. The total bandwidth usage on link e in $\mathbf{b}(\mathcal{P}|^i)$ is

$$
\begin{aligned}
& \textstyle\sum_{j \in \mathcal{U}_e(\mathcal{P}|^i)} b_j(\mathcal{P}|^i)\\
= \;& b_i(\mathcal{P}|^i) + \sum_{j \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}|^i) + \sum_{j \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}|^i)\\
\leq \;& (|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1) b_i(\mathcal{P}|^i) + \sum_{j \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}|^i) &(5.9)\\
< \;& (|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1) b_i^o(\mathcal{P}|^i) + \sum_{j \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_j(\mathcal{P}_{-i}) &(5.10)\\
\leq \;& b(e), &(5.11)
\end{aligned}
$$

where (5.9) follows from Property 2) of bottleneck, (5.10) follows from the condition $b_i(\mathcal{P}|^i) < b_i^o(\mathcal{P}|^i)$ and the assumption $b_j(\mathcal{P}|^i) \leq b_j(\mathcal{P}_{-i})$, and (5.11) follows from (5.2) and (5.1). This contradicts the fact that $e \in \mathcal{B}_i(P_i')$, because e should be saturated in $\mathbf{b}(\mathcal{P}|^i)$ according to Property 1) of bottleneck. This completes the proof of 1).

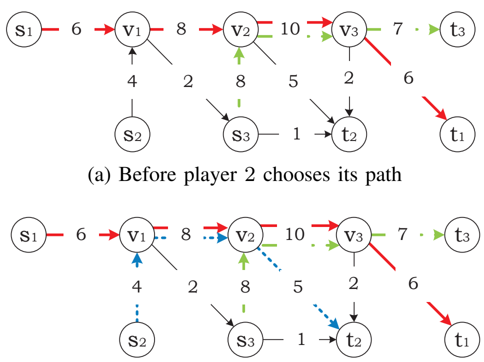
We now prove 2). Assume that $b_j(\mathcal{P}|^i) < b_j(\mathcal{P}_{-i})$.

By Property 2) of bottleneck, we know that $b_j(\mathcal{P}|^i) \geq b_k(\mathcal{P}|^i)$, $\forall k \in \mathcal{U}_e(\mathcal{P}|^i)$. Thus it suffices to prove that $\exists k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}$ such that **2a**) holds. The condition $b_j(\mathcal{P}|^i) < b_j(\mathcal{P}_{-i})$ implies that $i \neq j$. If $i \in \mathcal{U}_e(\mathcal{P}|^i)$, we can take $k = i$ and $b_i(\mathcal{P}|^i) > 0 = b_i(\mathcal{P}_{-i})$. Next, we consider the case where $i \notin \mathcal{U}_e(\mathcal{P}|^i)$. We prove **2a**) by contradiction. Assume that $b_k(\mathcal{P}|^i) \leq b_k(\mathcal{P}_{-i})$, $\forall k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}$. Note that $i \notin \mathcal{U}_e(\mathcal{P}|^i)$ implies $\mathcal{U}_e(\mathcal{P}_{-i}) = \mathcal{U}_e(\mathcal{P}|^i)$. The total bandwidth usage on link e in $\mathbf{b}(\mathcal{P}|^i)$ is

$$
b_j(\mathcal{P}|^i) + \sum_{k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}} b_k(\mathcal{P}|^i) < b_j(\mathcal{P}_{-i}) + \sum_{k \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \{j\}} b_k(\mathcal{P}_{-i}) \leq b(e),
$$

where the first inequality follows from the condition $b_j(\mathcal{P}|^i) < b_j(\mathcal{P}_{-i})$ and the assumption $b_k(\mathcal{P}|^i) \leq b_k(\mathcal{P}_{-i})$, $\forall k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{j\}$, and the second inequality follows from the feasibility of $\mathbf{b}(\mathcal{P}_{-i})$. This contradicts the fact that $e \in \mathcal{B}_j(\mathcal{P}|^i)$. Therefore, 2) holds.

We have finished the proof of this lemma. ■



(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.

Fig. 6 illustrates Part 2) of Lemma 5.3 with $i = k = 2$ and $j = 1$. From Fig. 6(a), we observe that $b_1(\mathcal{P}_{-2}) = 5$. From Fig. 6(b), we observe that $b_1(\mathcal{P}|^2) = 4$ and $b_2(\mathcal{P}|^2) = 4$. We note that the bandwidth of player 1 (red solid) decreases from 5 to 4 and the bandwidth player 2 (blue dotted) increases from 0 to 4. We also note that $b_2(\mathcal{P}|^2) \le b_1(\mathcal{P}|^2)$.

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

1) If $b_i(\mathcal{P}|^i) > b_i^o(\mathcal{P}|^i)$, then $\exists k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$, such that **1a)** $b_k(\mathcal{P}|^i) < b_k(\mathcal{P}_{-i})$ and **1b)** $b_k(\mathcal{P}|^i) < b_i(\mathcal{P}|^i)$, where $e \in \mathcal{B}_i^o(\mathcal{P}|^i)$ is an observed bottleneck of path P_i'.
2) If $b_j(\mathcal{P}|^i) > b_j(\mathcal{P}_{-i})$, then $\exists k \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \{j\}$, such that
2a) $b_k(\mathcal{P}|^i) < b_k(\mathcal{P}_{-i})$ and **2b)** $b_k(\mathcal{P}|^i) < b_j(\mathcal{P}|^i)$, where $e \in \mathcal{B}_j(\mathcal{P}_{-i})$ is a bottleneck of path P_j when player i's path is not in the network. □

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

We first prove 1). Assume that $b_i(\mathcal{P}|^i) > b_i^o(\mathcal{P}|^i)$.

We prove 1) by contradiction. Assuming to the contrary that $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$ or $b_k(\mathcal{P}|^i) \ge b_i(\mathcal{P}|^i)$, $\forall k \in \mathcal{U}_e(\mathcal{P}|^i) \setminus \{i\}$, we have the following two claims:

Claim 1: *For all* $k \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$, *we have* $b_k(\mathcal{P}|^i) \ge b_i^o(\mathcal{P}|^i)$.

When $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$ is true, we have

$$b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i}) \ge b^o(e) = b_i^o(\mathcal{P}|^i),$$

where the second inequality follows from Lemma 5.2 and the equality follows from the fact that $e \in \mathcal{B}_i^o(\mathcal{P}|^i)$. When $b_k(\mathcal{P}|^i) \ge b_i(\mathcal{P}|^i)$ is true, we have $b_k(\mathcal{P}|^i) \ge b_i(\mathcal{P}|^i) > b_i^o(\mathcal{P}|^i)$, due to the condition $b_i(\mathcal{P}|^i) > b_i^o(\mathcal{P}|^i)$.

Claim 2: *For all* $k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$, *we have* $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$.

Let k be any player in $\mathcal{U}_e(\mathcal{P}_{-i}) \backslash \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})$. We need to prove that $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$. According to the contrary assumption at the beginning of this proof, we only need to prove for the case where $b_k(\mathcal{P}|^i) \ge b_i(\mathcal{P}|^i)$ is true. In this case, we have

$$b_k(\mathcal{P}|^i) \ge b_i(\mathcal{P}|^i) > b_i^o(\mathcal{P}|^i) = b^o(e) > b_k(\mathcal{P}_{-i}),$$

where the last inequality follows from Lemma 5.2.

Note that $\mathcal{U}_e(\mathcal{P}|^i) = \mathcal{U}_e(\mathcal{P}_{-i}) \cup \{i\}$. The total bandwidth usage on link e in $\mathbf{b}(\mathcal{P}|^i)$ is

$$\begin{aligned}
&\sum_{k \in \mathcal{U}_e(\mathcal{P}|^i)} b_k(\mathcal{P}|^i) && (5.12)\\
&= b_i(\mathcal{P}|^i) + \sum_{k \in \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_k(\mathcal{P}|^i) + \sum_{k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_k(\mathcal{P}|^i) \\
&> (|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1) b_i^o(\mathcal{P}|^i) + \sum_{k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_k(\mathcal{P}_{-i}) && (5.13)\\
&= (|\tilde{\mathcal{U}}_e(\mathcal{P}_{-i})| + 1) b^o(e) + \sum_{k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \tilde{\mathcal{U}}_e(\mathcal{P}_{-i})} b_k(\mathcal{P}_{-i}) \\
&= b(e), && (5.14)
\end{aligned}$$

where (5.13) follows from the condition of 1) and the two claims, and (5.14) follows from (5.1). This contradicts the feasibility of $\mathbf{b}(\mathcal{P}|^i)$. Thus we have proved 1).

We now prove 2). Assume that $b_j(\mathcal{P}|^i) > b_j(\mathcal{P}_{-i})$.

We prove 2) by contradiction. Assume to the contrary that $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$ or $b_k(\mathcal{P}|^i) \ge b_j(\mathcal{P}|^i)$, $\forall k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \{j\}$. When $b_k(\mathcal{P}|^i) \ge b_j(\mathcal{P}|^i)$ is true, we have

$$b_k(\mathcal{P}|^i) \ge b_j(\mathcal{P}|^i) > b_j(\mathcal{P}_{-i}) \ge b_k(\mathcal{P}_{-i}),$$

where we used the condition of 2) and the fact that $e \in \mathcal{B}_j(\mathcal{P}_{-i})$. Thus we have $b_k(\mathcal{P}|^i) \ge b_k(\mathcal{P}_{-i})$, $\forall k \in \mathcal{U}_e(\mathcal{P}_{-i}) \setminus \{j\}$. Then, considering the fact that $\mathcal{U}_e(\mathcal{P}_{-i}) \subseteq \mathcal{U}_e(\mathcal{P}|^i)$, the total bandwidth usage on link e in $\mathbf{b}(\mathcal{P}|^i)$ is

$$b_j(\mathcal{P}|^i) + \sum_{k \in \mathcal{U}_e(\mathcal{P}|^i) \backslash \{j\}} b_k(\mathcal{P}|^i) > b_j(\mathcal{P}_{-i}) + \sum_{k \in \mathcal{U}_e(\mathcal{P}_{-i}) \backslash \{j\}} b_k(\mathcal{P}_{-i}) = b(e),$$

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

Fig. 6 illustrates Part 2) of Lemma 5.4 with $i = 2$, $j = 3$, and $k = 1$. From Fig. 6(a), we observe that $b_3(\mathcal{P}_{-2}) = 5$ and $b_1(\mathcal{P}_{-2}) = 5$. From Fig. 6(b), we observe that $b_3(\mathcal{P}|^2) = 6$ and $b_1(\mathcal{P}|^2) = 4$. We note that the bandwidth of player 3 (green dashed) increases from 5 to 6, but the bandwidth of player 1 (red solid) decreases from 5 to 4. We also note that $b_1(\mathcal{P}|^2) < b_3(\mathcal{P}|^2)$.

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 its 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 $b_i(\mathcal{P}|^i) = b_i^o(\mathcal{P}|^i)$. □

Proof. First, we prove that $b_i(\mathcal{P}|^i) \ge b_i^o(\mathcal{P}|^i)$. To the contrary, assume that $b_i(\mathcal{P}|^i) < b_i^o(\mathcal{P}|^i)$. We will derive a contradiction. By Part 1) of Lemma 5.3, we know that

$$\exists j, s.t., b_j(\mathcal{P}|^i) > b_j(\mathcal{P}_{-i}) \text{ and } b_j(\mathcal{P}|^i) \le b_i(\mathcal{P}|^i). \quad (5.15)$$

By the first inequality of (5.15) and Part 2) of Lemma 5.4, we know that

$$\exists k, s.t., b_k(\mathcal{P}|^i) < b_k(\mathcal{P}_{-i}) \text{ and } b_k(\mathcal{P}|^i) < b_j(\mathcal{P}|^i). \quad (5.16)$$

By the first inequality of (5.16) and Part 2) of Lemma 5.3, we know that

$$\exists j_1, s.t., b_{j_1}(\mathcal{P}|^i) > b_{j_1}(\mathcal{P}_{-i}) \text{ and } b_{j_1}(\mathcal{P}|^i) \le b_k(\mathcal{P}|^i). \quad (5.17)$$

By the first inequality of (5.17) and Part 2) of Lemma 5.4, we know that

$$\exists k_1, s.t., b_{k_1}(\mathcal{P}|^i) < b_{k_1}(\mathcal{P}_{-i}) \text{ and } b_{k_1}(\mathcal{P}|^i) < b_{j_1}(\mathcal{P}|^i). \quad (5.18)$$

Repeating (5.17) and (5.18), we obtain a sequence i, k, k_1, $k_2, \ldots$, such that $b_i(\mathcal{P}|^i) > b_k(\mathcal{P}|^i) > b_{k_1}(\mathcal{P}|^i) > b_{k_2}(\mathcal{P}|^i) > \cdots$. Since the number of users is finite, there must be a user that is repeated an infinite number of times in the above sequence of users. This is a contradiction, since the corresponding sequence of bandwidth values is strictly decreasing. This contradiction proves that $b_i(\mathcal{P}|^i) \geq b_i^o(\mathcal{P}|^i)$.

Using a similar logic, we can prove that $b_i(\mathcal{P}|^i) \leq b_i^o(\mathcal{P}|^i)$. This implies that $b_i(\mathcal{P}|^i) = b_i^o(\mathcal{P}|^i)$. ■

Remark 5.1. As a direct consequence of Theorem 5.1, player i has an incentive to change its strategy if and only if $b_i^o(\mathcal{P}|^i) > b_i(\mathcal{P})$. 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 the paths of other players. Without loss of generality, each player chooses a path with maximum bandwidth using an algorithm denoted by $\mathsf{WP}(G, s_i, t_i, b)$. 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 the following steps:

1) Compute its current bandwidth (Line 5).
2) Calculates the observed available bandwidth for each link in the resulting network (Lines 6 and 7).
3) Finds 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.

In Algorithm 2, $\mathsf{WP}(G, s_i, t_i, b)$ returns a path with maximum bandwidth from s_i to t_i in graph G with bandwidth function b. The basic idea of Algorithm 2 is as follows. First (Line 2), each player i chooses an initial s_i–t_i path regardless of other players. Next, in a round-robin fashion (Lines 3-11), each player changes its path to improve its utility, when possible. This is referred to as the *best-response move* in [21]. The process stops when no player can improve its bandwidth by changing to another path.

The correctness and an upper on the convergence speed of Algorithm 2 are captured in the following theorem.

Theorem 6.1: For every instance of the MAXBAR game, Algorithm 2 converges to a set $\mathcal{P}$ of paths in $O((Nm + n\log n + N\log N)(Nm)^N)$ time, where N is the number of players, m is the number of links, and n is the number of nodes. 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 the global bottleneck.

Algorithm 2: Game Based Algorithm

input : Network $G(V, E, b)$ and set $\mathcal{U}$ of players $\{1, \ldots, N\}$
output: A Nash Equilibrium $\mathcal{P}$

1 $\mathcal{P} \leftarrow \emptyset$;
2 $P_i \leftarrow \mathsf{WP}(G, s_i, t_i, b)$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$, $\forall i \in \mathcal{U}$;
3 **repeat**
4   **foreach** *player* $i \in \mathcal{U}$ **do**
5     $(b_1(\mathcal{P}), \ldots, b_N(\mathcal{P})) \leftarrow ComB(G, b, \mathcal{P}, \mathcal{U})$;
6     $(b_1(\mathcal{P}_{-i}), \ldots, b_N(\mathcal{P}_{-i})) \leftarrow ComB(G, b, \mathcal{P}_{-i}, \mathcal{U})$;
7   Compute $b^o(e)$ for all $e \in E$ using (5.1);
8     $P_i' \leftarrow \mathsf{WP}(G, s_i, t_i, b^o)$;
9   **if** $b^o(P_i') > b_i(\mathcal{P})$ **then** $\mathcal{P} \leftarrow \mathcal{P}|^i P_i'$;
10   **end**
11 **until** *there is no path changed*;
12 **return** $\mathcal{P}$;

Lemma 6.1: Let $\mathcal{P}$ be a path set of the users and $\mathbf{b}(\mathcal{P})$ be the corresponding MFBA. Let $\bar{e}$ be a global bottleneck. We then have $b_j(\mathcal{P}) = \frac{b(\bar{e})}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}$, $\forall j \in \mathcal{U}_{\bar{e}}(\mathcal{P})$. □

Proof. First, we claim that *for any* $e \in \mathcal{B}_i(\mathcal{P})$ *for some* i, *we have* $b_i(\mathcal{P}) \geq \frac{b(e)}{|\mathcal{U}_e(\mathcal{P})|}$. Considering both Properties 1) and 2) of bottleneck e, we have

$$b(e) = \sum_{j \in \mathcal{U}_e(\mathcal{P})} b_j(\mathcal{P}) \leq |\mathcal{U}_e(\mathcal{P})| \cdot b_i(\mathcal{P}).$$

Thus the claim is proved. Based on this claim and the fact that $\bar{e}$ is a global bottleneck, we have

$$b_j(\mathcal{P}) \geq \frac{b(\bar{e})}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}, \forall j \in \mathcal{U}_{\bar{e}}(\mathcal{P}). \quad (6.1)$$

Assume that $\exists k \in \mathcal{U}_{\bar{e}}(\mathcal{P})$ such that $b_k(\mathcal{P}) > \frac{b(\bar{e})}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}$. The total bandwidth usage on $\bar{e}$ is $\sum_{j \in \mathcal{U}_{\bar{e}}(\mathcal{P})} b_j(\mathcal{P}) > b(\bar{e})$, contradicting the feasibility of $\mathbf{b}(\mathcal{P})$. Hence we have proved that $b_j(\mathcal{P}) = \frac{b(\bar{e})}{|\mathcal{U}_{\bar{e}}(\mathcal{P})|}$, $\forall j \in \mathcal{U}_{\bar{e}}(\mathcal{P})$. ■

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 $O(Nm)$. For each possible value of b_1, there are at most N players whose paths correspond to this value. If the value of b_1 and the corresponding path P_i stay the same, the number of different possible values of b_2 is $O(Nm)$. The reason is that we can subtract b_1 from the bandwidth of each link along P_i, and remove i 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 $O((Nm)^N)$. The time complexity of Algorithm 1 is $O(Nm)$. Recall that computing $b^o(e)$ for all $e \in E$ takes $O(Nm + N\log N)$ time. In addition, the time complexity of $\mathsf{WP}(G, b, \mathcal{P})$ is $O(m+n\log n)$ by using a variant of Dijkstra's shortest path algorithm [9, 10]. Therefore the time complexity of Algorithm 2 is $O((Nm + n \log n + N \log N)(Nm)^N)$. By Theorem 5.1, the returned $\mathcal{P}$ is an NE, since no player can improve its utility by changing its path unilaterally. ■

Remark 6.1. Our extensive simulations in Section 7 show that the MAXBAR game converges to an NE within 10 iterations. This indicates that our theoretical bound $O((Nm)^N)$ 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 3–12 may lead to a *different* NE. However, Lines 3–12 of the algorithm will always lead to some NE.

Remark 6.3 In our algorithm, we require that only one player can change its path each time. This is essential to the convergence of the algorithm. We use an example to show that oscillation may occur when this requirement is violated. As shown in Fig. 7, assume that player 1's path is s_1-v_1-v_2-v_3-t_1 and player 2's path is s_2-v_1-v_2-v_3-t_2 at certain point of the game. If the players are allowed to change their paths simultaneously, player 1 and player 2 would change their paths to s_1-v_1-v_4-v_3-t_1 and s_2-v_1-v_4-v_3-t_2, respectively. Because both of them expect that they can increase their bandwidth from 1 to 2. Since two players change their paths simultaneously, the allocated bandwidth for each player is actually 1.5. Now both players would change their paths back to the previous ones because they expect to increase their bandwidth from 1.5 to 2. Therefore the network will oscillate between Fig. 7(a) and Fig. 7(b) if simultaneous path change is allowed.

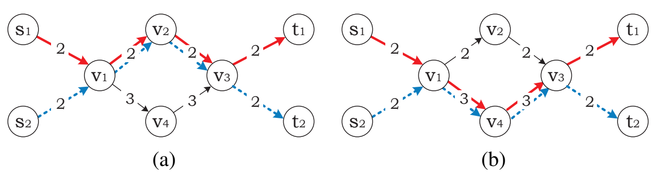


Fig. 7. Oscillation when simultaneous path change is allowed

One way to enforce the users in the network to follow the game course is to use a token-based protocol, where a token is circulated among the users in a round-robin fashion–only the user with the token has the opportunity to change its path. This token-based protocol can guarantee the convergence of Algorithm 2. A distributed implementation of $ComB(G, b, \mathcal{P}, \mathcal{U})$ were proposed by [5, 16]. The information needed by (5.1) to compute the observed available bandwidth is sent to each user by the link-state algorithm for determining the new path.

7. Generalization of MAXBAR

We have studied the MAXBAR problem where users have infinite bandwidth demand. In this section, we generalize the MAXBAR problem and consider the case where each user has a bandwidth demand of $\gamma_i > 0$. We denote this generalized problem as MAXBAR$_\gamma$. The difference between the MAXBAR$_\gamma$ problem and the MAXBAR problem is that we need to consider user's bandwidth demand while allocating bandwidth. Each user i will only use up to γ_i bandwidth and is not interested in switching to a path with more bandwidth as long as its bandwidth demand is met. The MAXBAR problem is a special case of the MAXBAR$_\gamma$ problem, as we can consider that $\gamma_i = \infty$ in the MAXBAR problem. It is seemingly necessary for us to redesign the ComB algorithm, and analyze the existence of NEs and convergence of routing again. However, we will show that we can transform any instance of the MAXBAR$_\gamma$ problem to a corresponding instance of the MAXBAR problem, and study the MAXBAR problem using the algorithms and analysis in previous sections.

Let $\mathcal{I}_\gamma = ((V, E, b), \mathcal{U}, \gamma)$ be an instance of the MAXBAR$_\gamma$ problem, where $G = (V, E, b)$ is the edge-weighted graph for the network. We build a corresponding instance $\mathcal{I} = ((V', E', b'), \mathcal{U}')$ of the MAXBAR problem (where $G' = (V', E', b')$ is the edge-weighted graph for the corresponding network) as follows. Corresponding to each node $v \in V$, V' contains a node v. Corresponding to each link $(v, w) \in E$, E' contains a link (v, w) and $b'(v, w) = b(v, w)$. Corresponding to each source $s_i \in V$, V' contains an *additional node* s'_i and E' contains an *additional link* (s'_i, s_i) with bandwidth $b'(s'_i, s_i) = \gamma_i$. Corresponding to each user $i \in \mathcal{U}$, $\mathcal{U}'$ contains a user i, who needs to transmit packets from s'_i to t_i in G'. Fig. 8 illustrates this transformation.

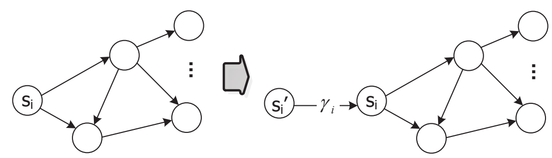


Fig. 8. Transforming an instance of the MAXBAR$_\gamma$ problem to a corresponding instance of the MAXBAR problem

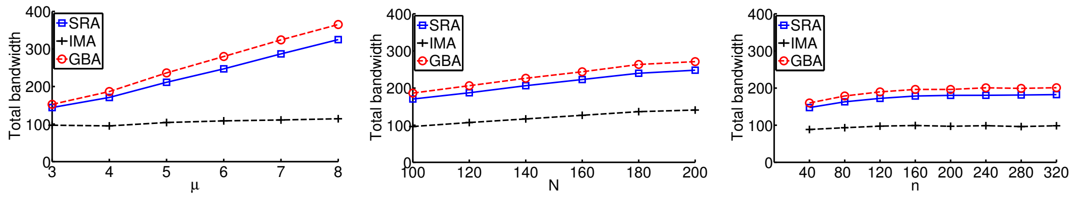
Note that although we allow users to have as much bandwidth as possible in the MAXBAR problem, the special link (s'_i, s_i) ensures that user i will only compete for bandwidth up to the demand γ_i. It is clear that the MFBA for $\mathcal{I}_\gamma$ can be obtained by computing the MFBA for $\mathcal{I}$. Therefore all the lemmas and theorems for the MAXBAR problem still hold for the MAXBAR$_\gamma$ problem.

8. Numerical Results

In this section, we evaluate the performance and verify the convergence analysis of Algorithm 2 (denoted as GBA) on network topologies generated by BRITE [4].

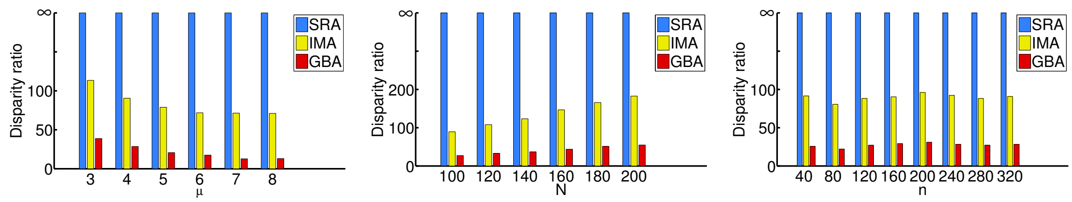
A. Simulation Setup

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. We denote this algorithm 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. It then 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) [22], 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).



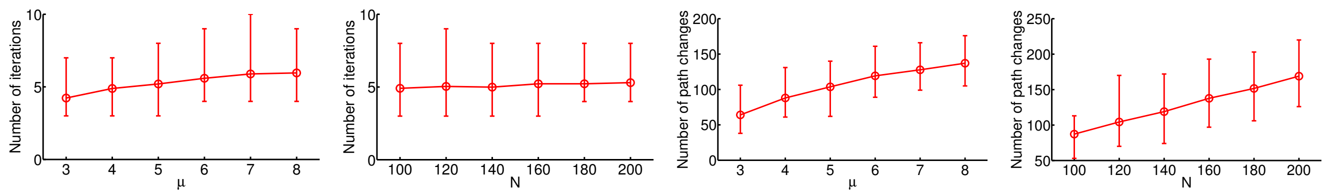
(a) Impact of μ ($n = 120, N = 100$) (b) Impact of N ($n = 120, \mu = 4$) (c) Impact of n ($\mu = 4, N = 100$)

Fig. 9. Total bandwidth



(a) Impact of μ ($n = 120, N = 100$) (b) Impact of N ($n = 120, \mu = 4$) (c) Impact of n ($\mu = 4, N = 100$)

Fig. 10. Disparity ratio



(a) Number of iterations (b) Number of iterations (c) Number of path changes (d) Number of path changes

Fig. 11. Convergence speed. For (a) and (c), $n = 120$ and $N = 100$. For (b) and (d), $n = 120$ and $\mu = 4$.

BRITE [4] is a widely used Internet topology generator. We used the Waxman model [25] 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 directed link (v, w) from v to 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 varied the number of nodes n from 40 to 320 with increment of 40 and set the number of links to $m = \mu n$, where μ is the link density and was varied from 3 to 8. We varied the number of users N from 100 to 200 with increment of 20. For each network size, we used BRITE to generate different network topologies, where the link bandwidth was drawn from a uniform distribution in the range $[1, 10]$. For each setting, we randomly generated 100 test cases and averaged the results.

Performance Metrics:

- Total bandwidth: the sum of the bandwidth of all users.
- Bandwidth disparity ratio: the ratio of the highest bandwidth over the lowest bandwidth among the users.
- Convergence speed: the number of the round-robin iterations (Lines 3–11 in Algorithm 2) or the number of path changes (Line 9 in Algorithm 2).

B. Results Analysis

1) Total Bandwidth: Fig. 9 shows the total bandwidth obtained by SRA, IMA 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, because some users can reserve most of the bandwidth resources in SRA. We also notice that the total bandwidth in Fig. 9(c) increases first and almost remains the same after $n = 240$. This is because the bandwidth of some users has reached the maximum value at $n = 240$.

2) Disparity Ratio: Fig. 10 shows the bandwidth disparity ratio obtained by SRA, IMA and GBA. We observe that GBA is the fairest. SRA has the worst disparity ratio with the value of ∞ for all settings. This is because some users will be blocked and have zero bandwidth in SRA, as other users have reserved all the bandwidth on the links connecting their sources and destinations. We also see that the disparity ratios of IMA and GBA are independent of n, as shown in Fig. 10(c), but decrease when the user density, $\frac{N}{m}$, becomes lower, as shown in Fig. 10(a) and Fig. 10(b). The reason is that when the user density is low, users have a low probability of sharing common links and hence competing the bandwidth. These results are not unexpected, as SRA and IMA are not designed to achieve small disparity ratios.

3) Convergence Speed: Fig. 11(a) and Fig. 11(b) show the number of iterations before GBA converges. We observe that the number of iterations is within 10 in all cases. Fig. 11(c) and Fig. 11(d) show the number of path changes before GBA converges. The theoretical bound on the number of path changes is $O((Nm)^N)$ in Theorem 6.1. However, as we can see, the number of path changes in the simulations is significantly less than the theoretical bound. Another observation is that GBA converges slower when the link density μ is high, as shown in Fig. 11(c). The reason is that when each node has more links, a user is highly likely to find a path with higher bandwidth if the current path results in low bandwidth, due to the competition from newly joined paths. According to Theorem 6.1, the number of path changes is independent of n. Our simulation results also confirm this proof and thus are omitted due to the space limitations.

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

9. **Conclusions**

In this paper, we formulated the problem of routing in networks with max-min fair bandwidth allocation as a non-cooperative game, where each user aims to maximize its own bandwidth. We proved the existence of Nash Equilibria, where no user has any incentive to unilaterally change its path. We derived both a lower bound and an upper bound of the system degradation, due to the selfish behavior of users. Finding a path with maximum bandwidth in the max-min fair network is both a key step for our main analysis and of independent interest. To this end, we introduced a novel concept of *observed available bandwidth* to accurately predict the available bandwidth on each link. 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. Note that the theoretical convergence speed proved in this paper does not change even when an approximate Nash Equilibrium [7] is considered. Deriving a tighter bound on the time complexity of the convergence speed is a future research direction. Through extensive simulations, we showed that the network can converge to an NE within 10 iterations and also achieve better fairness compared with other algorithms.

References

[1] R. Banner and A. Orda, "Bottleneck routing games in communication networks," *IEEE JSAC*, vol. 25, pp. 1173–1179, Aug. 2007.

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

[3] D. Bertsekas, R. Gallager, and P. Humblet, *Data networks*. Prentice-hall New York, 1987.

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

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

[6] S. Chen and K. Nahrstedt, "Maxmin fair routing in connection-oriented networks," in *Proc. Euro-Parallel and Distributed Systems Conf*, 1998.

[7] C. Daskalakis, P. Goldberg, and C. Papadimitriou, "The complexity of computing a nash equilibrium," in *ACM STOC*, 2006, pp. 71–78.

[8] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *ACM SIGCOMM*, 1989, pp. 1–12.

[9] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, pp. 269–271, 1959.

[10] M. Fredman and R. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *JACM*, vol. 34, pp. 596–615, 1987.

[11] D. Fudenberg and J. Tirole, *Game theory*. MIT Press, 1991.

[12] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 681–692, 2001.

[13] P. Godfrey, M. Schapira, A. Zohar, and S. Shenker, "Incentive compatibility and dynamics of congestion control," in *ACM SIGMETRICS*, 2010, pp. 95–106.

[14] T. Griffin, F. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 232–243, 2002.

[15] T. Griffin and G. Wilfong, "A safe path vector protocol," in *IEEE INFOCOM*, 2000, pp. 490–499.

[16] J. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. 29, pp. 954–962, 1981.

[17] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *ACM STACS*, 1999, pp. 404–413.

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

[19] A. Mayer, Y. Ofek, and M. Yung, "Approximating max-min fair rates via distributed local scheduling with partial information," in *IEEE INFOCOM*, 1996, pp. 928–936.

[20] D. Nace, N. Doan, E. Gourdin, and B. Liau, "Computing optimal max-min fair resource allocation for elastic flows," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 1272–1281, 2006.

[21] A. Nahir, A. Orda, and A. Freund, "Topology design and control: a game-theoretic perspective," in *IEEE INFOCOM*, 2009, pp. 1620–1628.

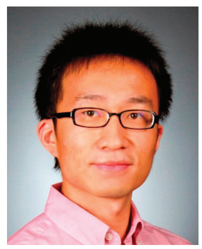
[22] RFC. [Online]. Available: Http://tools.ietf.org/html/rfc2205

[23] M. Schapira and A. Zohar, "Congestion-control games," 2008, a working paper. [Online]. Available: http://leibniz.cs.huji.ac.il/tr/1060.pdf

[24] M. Steenstrup, "Inter-domain policy routing protocol specification: Version 1," 1993.

[25] B. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, 1988.

[26] D. Yang, G. Xue, X. Fang, S. Misra, and J. Zhang, "Routing in max-min fair networks: A game theoretic approach," in *IEEE ICNP*, 2010, pp. 1–10.

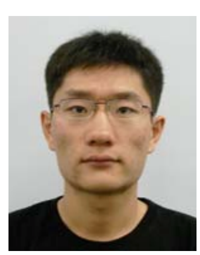


Dejun Yang (Student Member 2008) received his B.S. from Peking University, Beijing, China, in 2007. Currently he is a Ph.D. candidate in the School of Computing, Informatics, and Decision Systems Engineering (CIDSE) at Arizona State University. His research interests include economic and optimization approaches to networks, crowdsourcing, smart grid, big data, and cloud computing. He has received Best Paper Awards at IEEE MASS'2011, IEEE ICC'2011, 2012, and a Best Paper Award Runner-up at IEEE ICNP'2010.

Guoliang Xue (Member 1996, Senior Member 1999, Fellow, 2011) is a Professor of Computer Science at Arizona State University. His research interests include survivability, security, and resource allocation issues in networks. He has published over 200 papers in these areas. He is an Associate Editor of IEEE/ACM Transactions on Networking and IEEE Network. He was a Keynote Speaker at IEEE LCN'2011, and served as a TPC co-Chair of IEEE INFOCOM'2010. He is an IEEE Fellow.

Xi Fang (Student Member 2009) received the B.S. and M.S. degrees from Beijing University of Posts and Telecommunications, China. Currently he is a Ph.D student in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. He has received Best Paper Awards at IEEE MASS'2011 and IEEE ICC'2011, and a Best Paper Award Runner-up at IEEE ICNP'2010. His research interests include wireless networks, optical networks, cloud computing, and smart grids.

Satyajayant Misra (Student Member 2005, Member 2009) is an assistant professor at New Mexico State University. His research interests include anonymity, security, and survivability in wireless networks and protocol design for supercomputing and smart grid architectures. He serves on the editorial boards of the IEEE Communications Surveys and Tutorials and the IEEE Wireless Communications Magazine. He served as the TPC Vice-Chair of Information Systems for IEEE INFOCOM'2012.



Jin Zhang (Student Member 2010) received the B.E. degree from Beijing University of Posts and Telecommunications, China, in 2008. Currently, he is working toward the Ph.D. degree in the Department of Computer Science and Engineering at Arizona State University, Tempe. His research interest is in the area of designing and analyzing efficient medium access protocols that are robust against adversarial jamming in wireless networks.