Routing-independent Fairness in Capacitated Networks

Gábor Rétvári, József J. Bíró, Tibor Cinkler

High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
H-1117, Magyar Tudósok körútja 2., Budapest, Hungary
E-mail: {retvari, biro, cinkler}@tmit.bme.hu

***Abstract*—The problem of fair and feasible allocation of user throughputs in capacitated networks is investigated. The main contribution of the paper is an extension of network fairness, and in particular, max-min fairness from the traditional "fixed-path" model to a more versatile, routing-independent model. We show that the set of throughput configurations realizable in a capacitated network makes up a polyhedron, which gives rise to a max-min fair allocation completely analogous to the conventional one.**

I. Introduction

In this paper we address the problem of allocating scarce resources in a network so that every user gets a fair share, for some reasonable definition of fairness. For example, a fair allocation would be such that every user gets the same share, and the allocation is maximal in the sense that there does not exist any larger, even and feasible allocation. We shall focus on the fair allocation problem that arises most often in networking: compute a fair rate at which users can send data in a telecommunications network, whose links are of limited capacity.

Perhaps the most practical way to understand the context of this paper is through an example. Consider the simple directed network of Fig. 1a, and suppose that there are 3 source-destination pairs (or users or commodities): $(1, 5)$, $(2, 5)$ and $(3, 5)$. All the edge capacities are uniformly 1. Now, the task is to compute a transmission rate (or throughput, for short) for each user that is on the one hand feasible (so it can be routed in the network without violating the edge capacities) and, on the other hand, satisfies some fairness criteria. For example, according to the above naive interpretation of fairness, we would allocate $\frac{1}{2}$ amount of throughput for each user. This allocation is certainly feasible and gives even share to each user, and it is also maximal in this regard.

Amongst the many different definitions of fairness perhaps the most prevailing one is max-min fairness. A max-min fair allocation is, roughly speaking, such that we cannot increase the throughput of any of the users without decreasing the throughput of some other user, which is already smaller [1]. Max-min fairness is a simple yet powerful fairness criterion, and consequently it has grown to be an essential ingredient in diverse fields of networking, like flow control protocols [2], bandwidth sharing in ATM networks [3], etc.

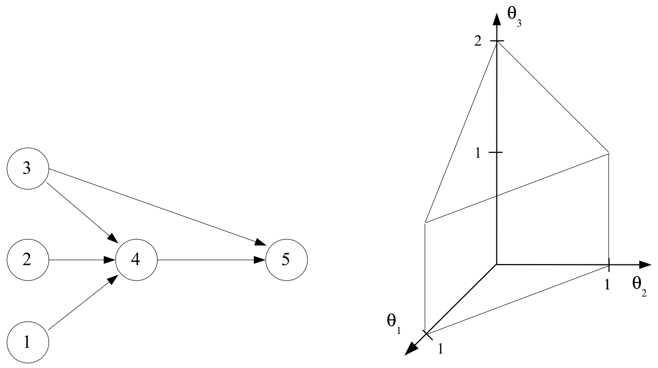
Max-min fairness is most easily described in a network model, where a single path is assigned to each user, and this path remains fixed during the lifetime of the communication. Here, the task is to compute a rate at which users can send data to their path, so that the allocation is max-min fair and neither of the edges gets overloaded. A very useful tool to solve this problem is the notion of bottlenecks [4]. A bottleneck edge, with respect to a certain user, is an edge with the properties that *(i)* it is filled to capacity and *(ii)* the user has the maximum throughput amongst the users whose path traverses the edge. Bottlenecks are very tightly coupled with max-min fairness, for it can be shown that an allocation of throughputs is max-min fair over some fixed single-path routing, if and only if all the users have a bottleneck edge.

From the practical standpoint, the importance of this *bottleneck argumentation* is multi-faceted. First, as the name suggests, bottlenecks point to certain shortages of resources in the network that, given the selected set of paths, constrain the fair allocation. Additionally, bottlenecks substantiate a fast algorithm, the so called *water-filling algorithm*, to find a max-min fair allocation [4]: we increase the throughput of the users at the same pace until an edge gets saturated. Then we fix the throughput of the users whose path passes through the saturated edge and keep on increasing others. The procedure is repeated until eventually a bottleneck is found for each user, and the allocation obtained is guaranteed to be the max-min fair allocation.

Assume that, in the sample network of Fig. 1, path $1 \to 4 \to 5$ is assigned to user $(1, 5)$, path $2 \to 4 \to 5$ to user $(2, 5)$, and the direct path $3 \to 5$ to user $(3, 5)$, respectively. Then, the edge that first becomes saturated as the water-filling algorithm proceeds is edge $(4, 5)$, which becomes the bottleneck edge for users $(1, 5)$ and $(2, 5)$. So the throughput of both of these users is fixed at $\frac{1}{2}$, and only the throughput of user $(3, 5)$ is increased any more. This, in turn, gets saturated at a throughput of 1 unit. The final max-min fair allocation is represented by the vector $[\frac{1}{2}, \frac{1}{2}, 1]$, using the order of users set out above.

Curiously, the actual selection of paths influences the emer-

*This work has been done as a part of the European sixth framework research project IP NOBEL (www.ist-nobel.org).



(a) Topology (b) $T(G_u)$

Figure 1: A sample network and the set of throughputs realizable in it. All edge capacities equal to 1. There are 3 source-destination pairs $(1, 5)$, $(2, 5)$ and $(3, 5)$, whose throughput is denoted by θ_1, θ_2 and θ_3, respectively.

gent max-min fair allocation to a great extent. For example, if the path of user $(3, 5)$ is changed to $3 \to 4 \to 5$, then the max-min fair throughput vector turns to $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. If we assign both paths $3 \to 5$ and $3 \to 4 \to 5$ to user $(3, 5)$ with the restriction that traffic must be split equally between the two paths (using multiple forwarding paths with preset splitting ratio is permitted by the fixed-path model), then the max-min fair allocation ends up to be $[\frac{2}{5}, \frac{2}{5}, \frac{2}{5}]$. Apparently, different routings give rise to different max-min fair allocations, which is somewhat unnatural since, after all, it is the network that determines feasible allocations. Accordingly, we should first compute a max-min fair allocation that is only dependent on the network itself, and only after this we should pick a routing that realizes it.

Recently, several attempts have been made to address this *general max-min fair allocation problem* using lexicographical optimization [5], [6]. These works are based on the observation that a max-min fair allocation is lexicographically maximal above the set of all feasible routings, so successive linear programming can be invoked to obtain it. The approach taken in [6] is, however, more general: it not only states the existence and the uniqueness of a max-min fair allocation over *any* compact and convex set, which the set of all possible routings certainly is, but it also gives an algorithm, called Max-min Programming, to compute it over any such set. While these excellent works provide adequate quantitative treatment, an in-depth qualitative analysis, which would reveal the intricate relationship between the specifics of a network and the emergent max-min fair allocation, is still absent in the literature. For instance, it is still not clear whether or not the bottleneck argumentation and, consequently, the water-filling algorithm generalize from the fixed-path model to the routing-independent, generic model, and if yes, then in what particular form bottlenecks arise. These questions have gone mostly unresolved so far, albeit their relevance has been very clearly pointed out [6, Section "When bottleneck and water-filling become less obvious"].

In this paper, we offer affirmative answers to these important questions. After a quick roundup on the notation in Section II, we shall introduce a novel polyhedral description of the throughput allocations realizable in a network (see Section III). This polyhedral description is so concise that for simple networks we can as well easily visualize it (see Fig. 1b) and it allows us to gain interesting new insights into the general max-min fair allocation problem. In Section IV we present our treatment and then, in Section V, we reveal how the bottleneck argumentation extends to the routing-independent case. Finally, in Section VI we summarize our contributions.

Reading this paper requires a minimal understanding of the theory of network flows and linear programming. To make it more readable even to the less mathematically inclined, all the proofs are deferred to the Appendix. For a good introductory material on polyhedra the reader is referred to [7].

II. Preliminaries

In this section, we present the most important notations and conventions we shall use throughout this paper. A vector will be denoted by a lowercase letter. Most of the time, the ith coordinate of a vector v will be referred to as v_i, but in some cases, to stress that we are dealing with a specific coordinate, we shall use the notation $(v)_i$. What now follows is a list of the notation we shall use in the sequel:

- $G(V, E)$: a directed graph, with the set of nodes V ($|V| = n$) and the set of directed edges E ($|E| = m$).
- u: the column m-vector of edge capacities.
- $(s_k, d_k) : k \in \mathcal{K}, \mathcal{K} = \{1, \dots, K\}$: the set of source-destination pairs (users or commodities).

Note that the graph $G(V, E)$, the edge capacities u and the set of source-destination pairs $(s_k, d_k) : k \in \mathcal{K}$ together describe a *network*, which will be referred to as G_u for brevity. Hereafter, we shall only deal with networks that satisfy certain, rather mild, regularity conditions:

Definition 1: A network G_u is *regular*, if

- a path exists in G_u from s_k to d_k for each $k \in \mathcal{K}$ and
- all edge capacities are finite and strictly positive.

It is easy to see that any network can be reduced to a collection of regular networks by eliminating edges with zero capacity and fixing the throughput of the un-connected users at zero. The further notation goes on as follows:

- e_i: the canonical unit vector (of proper size implied by the context) with 1 in the position corresponding to the ith coordinate and all zero otherwise.
- $\mathbf{1}$: an all-one vector of proper size.
- $\mathcal{P}_k$: the set of all directed paths from s_k to d_k in $G(V, E)$ for some $k \in \mathcal{K}$.
- Δ_k: an $m \times |\mathcal{P}_k|$ matrix. The column corresponding to path $P \in \mathcal{P}_k$ holds the path-arc incidence vector of P.
- f_k: a column vector of path-flows, whose coordinate corresponding to path $P \in \mathcal{P}_k$ denotes the amount of flow sent by user k to path P.
- f: a column-vector of f_ks: $f = [f_1, f_2, \dots, f_K]$. In fact, f represents a *routing* in G_u.

- θ_k: the throughput of some user $k \in \mathcal{K}$, that is, the aggregate flow that flows from s_k to d_k. The vector of throughputs is a column K-vector θ.
- $\beta\theta \leq b$: an inequality constraining the set of throughputs, where β is a row K-vector and b is a scalar. An $\beta\theta \leq b$ is *valid* for some set T, if $\forall\theta \in T : \beta\theta \leq b$.
- $T(G_u)$: the set of throughputs realizable in the network G_u, subject to edge capacity constraints.
- $\mathcal{S}$: a separating edge set, that is, a set of edges $\mathcal{S} \subseteq E$ whose removal from the network would destroy all the directed s_k to d_k paths for at least one user $k \in \mathcal{K}$.
- $\mathcal{K}_{\mathcal{S}}$: the set of users disconnected by some separating edge set $\mathcal{S}$.

III. The Throughput Polytope

The central problem we investigate in this paper is to determine a feasible and fair allocation of user throughputs in a capacitated network, independently of paths fixed beforehand in any ways. A plausible way to attack this problem would be to describe the entire set of possible flow routings f and throughput allocations θ as a giant set and then search for the fair allocation in this very set. Consider the following formulation:

$$
\begin{aligned}
M(G_u) = \{[f,\theta] : \quad & \textstyle\sum_{k\in\mathcal{K}} \Delta_k f_k \leq u & & (1)\\
& \mathbf{1} f_k = \theta_k & \forall k \in \mathcal{K} \quad & (2)\\
& f_k \geq 0, \theta_k \geq 0 & \forall k \in \mathcal{K} \ \} \quad & (3)
\end{aligned}
$$

Readers more proficient in network flow theory might find this formulation familiar, since $M(G_u)$ is in fact the set of feasible solutions of the family of *multicommodity flow problems*. Here (1) requires that, for all edges, the sum of all path-flows routed to the edge does not exceed the capacity of that edge; (2) produces the throughput for each user by summing up the flow traveling along each of its paths; and (3) requires the flows and throughputs to be non-negative.

While using $M(G_u)$ to deduce a routing-independent fair allocation is clearly viable (see e.g. [5]), it is unfortunate in many regards, most notably because $M(G_u)$ usually has a plethora of problem variables, the majority of which completely redundant. This is because, at the moment, we are only interested in a fair throughput allocation but not in the way this allocation is accommodated in the network, apart from the requirement that the allocation must be realizable by some legitimate routing. Therefore, instead of studying the full-fledged set $M(G_u)$ one might rather choose to eliminate the path-flow variables f all together from the description. The emergent set, denoted hereafter by $T(G_u)$, contains all the possible throughput allocations feasible in G_u:

Definition 2: $T(G_u) = \{\theta : \exists f \text{ so that } [f,\theta] \in M(G_u)\}$.
For the sample network of Fig. 1a, the corresponding set of feasible throughput allocations is depicted in Fig. 1b. To obtain it, we reason as follows. Let θ_1 denote the throughput of user $(1,5)$, θ_2 of user $(2,5)$ and θ_3 of user $(3,5)$, respectively. Since we can not push more flow than 1 via the edge $(4,5)$, which is traversed by all the potential paths of user $(1,5)$ and $(2,5)$, we have that $\theta_1+\theta_2 \leq 1$. Furthermore, after routing 1 unit of flow of user $(3,5)$ along the edge $(3,5)$, every additional ϵ units of flow of this user have to traverse edge $(4,5)$, decreasing the aggregate throughput remaining available to user $(1,5)$ and $(2,5)$ by exactly ϵ units. So, $\theta_1 + \theta_2 + \theta_3 \leq 2$. It can be shown that these inequalities, together with the restriction that the throughputs are non-negative, give rise to a complete and irredundant description of the set of throughputs realizable in the network of Fig. 1a:

$$
\begin{aligned}
T(G_u) = \{[\theta_1,\theta_2,\theta_3] : \quad & \theta_1 + \theta_2 \leq 1 & (4)\\
& \theta_1 + \theta_2 + \theta_3 \leq 2 & (5)\\
& \theta_1, \theta_2, \theta_3 \geq 0 \quad \} & (6)
\end{aligned}
$$

Observe how all the constraints turned out to be linear. Sets of similar kind are called polyhedra, which might be familiar as these are exactly the geometric objects that underlie linear programming. A polyhedron is basically an intersection of finitely many halfspaces, and as such, closed and convex. Additionally, a bounded polyhedron is called a *polytope*. The result below reveals that the set $T(G_u)$ is not coincidentally polyhedral in our example.

Proposition 1: $T(G_u)$ is a polyhedron. Provided that G_u is regular, $T(G_u)$ is a polytope.

Henceforward, we shall only deal with regular networks, and so we shall refer to $T(G_u)$ as the *throughput polytope*. But not just that $T(G_u)$ is a polytope with "nice" properties like convexity and compactness, it has yet another interesting quality that makes it even more attractive to work with: observe that in the formulation (4)–(6), every coefficient and also the right-hand-side of all the inequalities are non-negative. This, as the next result claims, is again not coincidental, but instead a very important general property of throughput polytopes, one that we shall exploit in the next section when we shall discuss fair allocations arising in $T(G_u)$.

Proposition 2: For a regular network G_u, the corresponding throughput polytope can always be transformed to the following standard form:

$$T(G_u) = \{\theta \geq 0 : \beta_i\theta \leq b_i, \quad \forall i \in \mathcal{I}\} \ ,$$

where $\mathcal{I}$ is a (finite) index set and for each $i \in \mathcal{I}$ it holds that β_i is a row K-vector with $\beta_i \geq 0$ and b_i is a positive scalar.

IV. Max-min Fair Allocations

In the previous section, we introduced the throughput polytope as the lower-dimensional projection of the set of all feasible routings and throughput allocations, with the path-flow variables eliminated. In this section, we shall study various sorts of fair allocations arising in a network by means of the corresponding throughput polytope.

When deciding which particular throughput allocation to offer for the users, the first requirement one has to consider is that the allocation must be feasible. Feasibility is, however, easy to assure in our model: one might choose whatever $\theta \in T(G_u)$ and the construct then automatically assures that this θ will be realizable by some legitimate routing. The second requirement is the allocation must preclude unnecessary wastage of network resources. A way to achieve this would be to select an allocation that is maximal in the sense that "allocating more to some user is only possible at the expense of allocating less to some other user". Such "maximal" allocations are called *Pareto-efficient*. Unfortunately, Pareto-efficiency allows for allocations where one user gets everything, which is not really fair (observe for instance that the allocation $[0,0,2]$ is Pareto-efficient in the network of Fig. 1a). Therefore, we must choose one particular θ_0 that is on the one hand feasible and (Pareto) efficient and, on the other hand, offers some sorts of fair treatment to the users. Max-min fairness has become the allocation strategy of choice in many areas of networking thanks to the remarkably simple fairness rule it implements: "there is no way to make anybody better off without hurting somebody else, who is already poorer". In the case of throughput allocations in capacitated networks, the formal definition is as follows:

Definition 3: An allocation of throughputs θ_0 is max-min fair, if it is feasible and $\forall \theta \in T(G_u) : (\theta)_k > (\theta_0)_k \Rightarrow \exists l \in \mathcal{K} \setminus \{k\}$, so that $(\theta)_l < (\theta_0)_l$ and $(\theta_0)_l \leq (\theta_0)_k$.

It is by far not evident whether or not this definition makes sense in the case of $T(G_u)$ or, in fact, how many max-min fair allocations it yields. Though, the following claim states that the notion of max-min fairness over $T(G_u)$ is well-defined:

Proposition 3: Let G_u be a regular network. Then there exists a max-min fair allocation over $T(G_u)$, and it is unique.

Being now safe that the general max-min fair allocation problem is soluble, we now move on to investigate how to actually compute that solution. In this process the following result, which relates the max-min fair allocation θ_0 to certain inequalities holding with strict equality at θ_0, will be of great help:

Theorem 1: Some $\theta_0 \in T(G_u)$ is max-min fair, if and only if for each $k \in \mathcal{K}$ there exists an inequality $\beta\theta \leq b$, the so called *bottleneck inequality*, such that:

i) $\beta \geq 0$
ii) $\forall \theta \in T(G_u) : \beta\theta \leq b$ (so the inequality is *valid*)
iii) $\beta\theta_0 = b$
iv) $\forall l \in \mathcal{K} : (\beta)_l > 0$ if and only if $(\theta_0)_l \leq (\theta_0)_k$

What is remarkable in this result is that bottleneck inequalities work very much like bottleneck edges in the fixed-path model (hence the name). With this analogy in mind we could rephrase Theorem 1 as: *an allocation of throughputs is max-min fair in the generic sense, if and only if all users have a bottleneck (inequality).* This formulation is exactly the same as the one given for the fixed-path model, only the definition of bottlenecks differs somewhat. Interestingly, the analogy goes even further since not just bottlenecks but the water-filling algorithm too extends to the general max-min fair allocation problem. Recall that the water-filling algorithm is based on the idea to generate a bottleneck for at least one user in every iteration, no matter in which form bottlenecks are defined. Provided that the bottlenecks arise in the form of a bottleneck inequality, Theorem 1 guarantees that what we eventually obtain by running the water-filling algorithm on the throughput polytope is exactly the max-min fair allocation. Thus, the second important consequence of this theorem is that *the water-filling algorithm is correct to search for a max-min fair allocation over $T(G_u)$.*

Consider the network of Fig. 1a, and execute the water-filling algorithm using the throughput polytope (4)–(6) depicted in Fig. 1b. As the first step, increase the throughput of all the users at the same pace. This amounts to, starting from the origin, moving along the direction $[1,1,1]$ as long as some of the users gets blocked. This occurs at the point $[\frac{1}{2},\frac{1}{2},\frac{1}{2}]$, where the constraint (4) becomes active. As a matter of fact, this constraint will be the bottleneck for users $(1,5)$ and $(2,5)$. The only user not yet blocked is $(3,5)$, whose throughput can be increased to 1. The resultant allocation $\theta_0 = [\frac{1}{2},\frac{1}{2},1]$ is max-min fair, and the bottleneck inequality of the last user turns out to be constraint (5) (note that it is generally *not* true that the bottleneck inequalities correspond to the constraints of $T(G_u)$).

The final question that remained to be answered is that, once we computed the max-min fair allocation θ_0, how to actually obtain a routing that realizes it. That is, we need to find path-flows $f : [\theta_0, f] \in M(G_u)$. This amounts to solving a multicommodity flow problem, which can be done in polynomial time [8]. The computed path-flows will then supply a rate at which users have to distribute their traffic to their paths and an actual routing, which, once established in the network, will automatically realize the max-min fair throughput allocation θ_0 using the exact same distributed flow control and queuing techniques as in the fixed-path model [9].

V. A Bottleneck Argumentation

So far, we have shown how the concept of bottlenecks extends from the fixed-path max-min fairness problem to the generic case. Analogously to the traditional model, we could obtain an "if and only if" relation between the existence of bottlenecks for each user and max-min fairness, which also guaranteed the correctness of the water-filling algorithm. Quite regrettably, however, our bottlenecks are currently defined in terms of valid inequalities, which, being more of a polyhedral concept than a network theoretical one, is not really descriptive. In this section, we translate this bottleneck argumentation to the more palpable concept of separating edge sets, whose properties show remarkable similarity to the properties of "bottleneck edges" in the fixed-path model.

In the heart of the fixed-path model there lies the notion of bottleneck edges. A bottleneck edge is one that blocks any increase in the throughput of the user it belongs to. This is because *(i)* it is filled up to capacity when we realize the max-min fair allocation, and *(ii)* the corresponding user has the maximum throughput amongst the users that might want to use that edge. This conventional interpretation fails in the generic model, since neither the set of paths nor the users of a particular edge are fixed.

A bottleneck edge blocks one particular, fixed path of some user. To block *all* paths we have to treat an entire set of edges, a so called separating edge set, which, when removed from the network, destroys all directed paths connecting the source to the destination node. This suggests the idea to search for the generalization of bottleneck edges in the form of *bottleneck separating edge sets*. What remained to be done is to translate the defining properties of bottleneck edges to separating edge sets.

Let θ_0 be max-min fair in a regular network G_u and choose some user $k \in \mathcal{K}$. Additionally, suppose that we have somehow found the corresponding bottleneck separating edge set $\mathcal{S}_k$ and let $\mathcal{K}_{\mathcal{S}_k} \subseteq \mathcal{K}$ denote the set of users, whose source node is separated away from the respective destination node by $\mathcal{S}_k$. First, we reformulate the following property of bottleneck edges: a user's throughput is maximal at its bottleneck edge amongst the ones that utilize that edge. But "utilizers" of separating edge sets are exactly the users that are separated away by it, so for $\mathcal{S}_k$ it must hold that:

Property 1: $l \in \mathcal{K}_{\mathcal{S}_k} \Leftrightarrow (\theta_0)_l \leq (\theta_0)_k$.

The second defining property of bottleneck edges is that they are always filled to capacity when we realize the max-min fair allocation and, furthermore, in the fixed-path model there is no way for the traffic of the blocked users to circumvent this bottleneck. We translate this property to separating edge sets as follows:

Property 2: For any routing f that realizes θ_0, it holds that

$$\forall\ (i,j) \in \mathcal{S}_k : \sum_{l \in \mathcal{K}_{\mathcal{S}_k}} \sum_{P \in \mathcal{P}_l : (i,j) \in P} (f_l)_P = u_{ij}$$

In words, Property 2 insists that a bottleneck separating edge set is always saturated by the flow of the users separated away by it, no matter how we route the max-min fair allocation in the network. Therefore, any increase in the throughput of some user would decrease the throughput of some other user that utilizes the same (bottleneck) separating edge set, and consequently, whose throughput is already smaller (by Property 1), and this property is independent of the actual routing. Interestingly, these properties give rise to a bottleneck argumentation completely analogous to the conventional one:

Theorem 2: An allocation of throughputs $\theta_0 \in T(G_u)$ is max-min fair, if and only if each user has a bottleneck separating edge set exhibiting both Property 1 and Property 2.

It is now fairly easy to find the bottleneck separating edge sets corresponding to the network of Fig. 1a. User $(1,5)$ and $(2,5)$ have the same bottleneck separating edge set constituted by the single edge $(4,5)$. Observe that *i)* removing this edge would cut away the endpoints of these users, *ii)* the throughput of these users is maximal amongst the set of users separated away by the edge and finally *iii)* this edge will always be filled to capacity no matter how we route the max-min fair allocation $\theta_0 = [\frac{1}{2}, \frac{1}{2}, 1]$ in the network (in fact, we have only one option to choose from). We kindly encourage the reader to check that these three properties readily apply to the bottleneck separating edge set of user $(3,5)$: $\{(3,5),(4,5)\}$.

As a final remark, we note that our bottleneck argumentation contains the conventional one as a special case. To see this, it is enough to restrict each user to use one single path and observe that bottleneck separating edge sets degrade to the conventional bottleneck edges in this case.

VI. CONTRIBUTIONS

Traditionally, fair allocation of user throughputs has been considered in the case when the path of the users is fixed for the lifetime of the communication. In this model, users get whatever "fair" share of network resources the actual routing allows them to receive. However, a user might ask rightfully: "Why has exactly *this* routing been implemented in the network instead of another one, which would be more beneficial for me within the current throughput allocation strategy?" This argumentation holds some merit, because in the fixed-path model the throughput allocated to a user depends quite heavily on the route taken by the traffic of that user, which, within the network architectures of our days, the user is not quite empowered to affect. In this paper we argued that it is much more natural to make throughput allocation strategies independent of routing, and we have extended the most commonly used fairness criterion, max-min fairness, to this generic case. If the throughput was determined independently of the actual routing, then no one would have the right to complain since it was the network, a given entity, that decided which particular share of network resources a user receives.

To solve this general throughput allocation problem, first we introduced the throughput polytope, a polyhedral description of the range of throughput configurations realizable in a capacitated network. This construct is notable, not only because it helped us to characterize max-min fair allocations in the generic, routing-independent model, but also because it can easily help to do the same with other notions of fairness, like proportional fairness or utility fairness. For instance, our treatment is almost straightforward to extend to the case for weighted max-min fairness or min-max fairness.

Next, we showed that there always exists a unique routing-independent max-min fair throughput allocation in a regular network and it can be obtained by the water-filling algorithm. Although this algorithm is faster than the ones available in the literature [5], its practical utility for solving the generic max-min fair allocation problem is limited, because it necessitates the throughput polytope to have been computed in advance, which itself requires substantial computational efforts and might very well turn out to be intractable in large networks [10]. Finally, exploiting the special structure of the throughput polytope we extended the well-known bottleneck argumentation from the fixed-path model to the generic one. Note that this result is more universal than one might think at the first glance, because our proofs remain valid for any other down-monotone polyhedron that satisfies the non-negativity requirements in Proposition 2.

REFERENCES

[1] J. M. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, vol. 29, pp. 954–962, July 1981.

[2] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas of Communication*, vol. 9, pp. 1024–1039, Sept. 1991.

[3] A. F. T. Committee, "Traffic Management Specification - Version 4.0." ATM Forum/95-0013R13, Feb 1996.

[4] D. P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[5] D. Nace and L. Doan, "A polynomial approach to the fair multi-flow problem." Tech. Rep., Heudiasyc, UTC, available online: http://www.hds.utc.fr/~dnace/recherche/Publication/TR-MMF.pdf, 2002.
[6] J. Y. Le Boudec and B. Radunovic, "A unified framework for max-min and min-max fairness with applications," in *Proceedings of 40th Annual Allerton Conference on Communication, Control, and Computing*, Oct 2002.
[7] G. Ziegler, *Lectures on Polytopes*, vol. 152 of *Graduate Texts in Mathematics*. New York: Springer, 1998.
[8] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. New York: John Wiley & Sons, second ed., 1990.
[9] J. Le Boudec, "Rate adaptation, congestion control and fairness: a tutorial." available online: http://ica1www.epfl.ch/PS_files/LEB3132.pdf, Feb 2005.
[10] G. Rétvári, J. J. Bíró, and T. Cinkler, "Fairness in capacitated networks: a polyhedral approach." in Proceedings of INFOCOM'07, May 2007.
[11] M. Iri, "On an extension of the maximum-flow minimum-cut theorem to multicommodity flows," *Journal of the Operations Research Society of Japan*, vol. 13, no. 3, pp. 129–135, 1971.
[12] K. Onaga and O. Kakusho, "On feasibility conditions of multicommodity flows in networks," *IEEE Transactions on Circuit Theory*, vol. 18, no. 4, pp. 425–429, 1971.

APPENDIX

Proof of Proposition 1: $M(G_u)$ is, by definition (1)–(3), an intersection of finitely many halfspaces, so it is a polyhedron. By Definition 2, $T(G_u)$ is the orthogonal projection of $M(G_u)$ to the space spanned by θ, therefore it is itself too a polyhedron [7]. Finally, compactness for regular networks is straightforward. ■

Proof of Proposition 2: Applying Černikov's block-elimination method [7] to $M(G_u)$, we have that row K-vectors β and row m-vectors w lying in the projection cone

$$W(G_u) = \{[\beta, w] : \quad \textstyle\sum_{(i,j)\in P} w_{ij} \geq \beta_k \quad \forall k \in \mathcal{K}, \forall P \in \mathcal{P}_k \\ w \geq 0 \ \}$$

generate *all* the inequalities of $T(G_u)$: $T(G_u) = \{\theta \geq 0 : \beta\theta \leq wu,\ \forall[\beta, w] \in W(G_u)\}$. In fact, it is enough to take the inequalities generated by the *extreme rays* of $W(G_u)$, so $\mathcal{I}$ is finite. Observe that here, vectors w can be thought of as non-negative *edge weights*, while the kth coordinate of β, β_k, is less than, or equal to the length of the shortest path from s_k to d_k over the edge weights w. (Note that in all practically important cases $(\beta)_k$ in fact attains the length of the shortest path.) Hence, β is non-negative and, for a regular G_u, wu is strictly positive. See alternative proofs in [11] and [12] (the Japanese Theorem). ■

Proof of Proposition 3: Since $T(G_u)$ is a polytope, it is, by nature, convex and compact. Then the existence and uniqueness of the max-min fair vector is guaranteed by [6, Theorem 1 and Proposition 1]. ■

Proof of Theorem 1: Let θ_0 be max-min fair and, for each $k \in \mathcal{K}$, construct the vector θ' so that

$$(\theta')_l = \begin{cases} (\theta_0)_k & \text{if } (\theta_0)_l > (\theta_0)_k \\ (\theta_0)_l & \text{otherwise} \end{cases} \tag{7}$$

We prove that the sum of the constraints of $T(G_u)$ binding at θ' satisfies all the requirements of the theorem.

Let $T(G_u) = \{\theta \geq 0 : \beta_i\theta \leq b_i,\ i \in \mathcal{I}\}$ and let $\mathcal{B}$ be the set of constraints binding at θ': $\mathcal{B} = \{i \in \mathcal{I} : \beta_i\theta' = b_i\} \neq \emptyset$. Let $\beta = \sum_{i\in\mathcal{B}} \beta_i$ and $b = \sum_{i\in\mathcal{B}} b_i$. Obviously, $\beta\theta \leq b$ is valid for $T(G_u)$ and $\beta \geq 0$, so the first two claims immediately apply. To see that the latter two claims also apply, it is enough to show that $(\beta)_l > 0 \Leftrightarrow (\theta_0)_l \leq (\theta_0)_k$ and the rest follows from $\beta \geq 0$. Now, θ_0 is max-min fair $\Leftrightarrow$ increasing $(\theta_0)_k$ is only possible at the expense of decreasing some $(\theta_0)_l$ with $(\theta_0)_l \leq (\theta_0)_k \Leftrightarrow \forall l \in \mathcal{K}$ with $(\theta_0)_l \leq (\theta_0)_k : \theta' + \epsilon e_l \notin T(G_u)$ for each $\epsilon > 0 \Leftrightarrow \forall l \in \mathcal{K}$ with $(\theta_0)_l \leq (\theta_0)_k : (\beta)_l > 0$. The reverse direction of the proof comes similarly. ■

Proof of Theorem 2: We have already seen that some $\theta_0 \in T(G_u)$ is max-min fair, if and only if each $k \in \mathcal{K}$ has a bottleneck inequality $\beta\theta \leq wu$ conforming to (i)–(iv) in Theorem 1. Here, w can be thought of as edge weights and $(\beta)_k$ as the length of the shortest path from s_k to d_k over the weights w. Now, define the corresponding bottleneck separating edge set as

$$\mathcal{S}_k = \{(i,j) \in E : w_{ij} > 0\} \ . \tag{8}$$

This implies that $\mathcal{K}_{\mathcal{S}_k} = \{l \in \mathcal{K} : (\beta)_l > 0\} = \{l \in \mathcal{K} : (\theta_0)_l \leq (\theta_0)_k\}$, using item (iv) in Theorem 1. So $\mathcal{S}_k$ as defined by (8) immediately satisfies Property 1. To prove the theorem, we only need to show that it fulfills Property 2 too. For this, first we observe that vector $[\beta, w]$ taken from the bottleneck inequality of k solve the following linear program defined over the separating edge set $\mathcal{S}_k$ as defined by (8), with optimal objective function value zero:

$$\begin{aligned} 0 = \min wu - \beta\theta_0 & \\ w\Delta_l \geq \mathbf{1}(\beta)_l \quad & \forall l \in \mathcal{K} \\ \beta_l \geq 1 \quad & \forall l \in \mathcal{K}_{\mathcal{S}_k} \\ \beta_l = 0 \quad & \forall l \in \mathcal{K} \setminus \mathcal{K}_{\mathcal{S}_k} \\ w_{ij} \geq 1 \quad & \forall (i,j) \in \mathcal{S}_k \\ w_{ij} \geq 0 \quad & \forall (i,j) \in E \setminus \mathcal{S}_k \end{aligned}$$

Now, by the strong duality theorem of linear programming, the dual linear program below is also soluble and the optimal objective function value is zero:

$$\begin{aligned} 0 = \max \textstyle\sum_{(i,j)\in\mathcal{S}_k} \lambda_{ij} + \sum_{l\in\mathcal{K}_{\mathcal{S}_k}} \mu_l & \\ \textstyle\sum_{l\in\mathcal{K}} \Delta_l f_l + \lambda = u & \\ \mathbf{1} f_l - \mu = (\theta_0)_l \quad & \forall l \in \mathcal{K} \\ \mu_l \geq 0 \quad & \forall l \in \mathcal{K}_{\mathcal{S}_k} \\ f_l \geq 0, \lambda \geq 0 \quad & \forall l \in \mathcal{K} \end{aligned}$$

Let $[f, \lambda, \mu]$ be *any* optimal feasible solution. Now, the objective function value is zero, if and only if $\forall l \in \mathcal{K}_{\mathcal{S}_k} : \mu_l = 0$ and $\forall (i,j) \in \mathcal{S}_k : \lambda_{ij} = 0$, which exactly reproduces Property 2 and the proof is complete. ■