Optimal Multi-Path Routing and Bandwidth Allocation under Utility Max-Min Fairness

Jerry Chou and Bill Lin
University of California San Diego, La Jolla, CA 92093

***Abstract*—An important goal of bandwidth allocation is to maximize the utilization of network resources while sharing the resources in a fair manner among network flows. To strike a balance between fairness and throughput, a widely studied criterion in the network community is the notion of max-min fairness. However, the majority of work on max-min fairness has been limited to the case where the routing of flows has already been defined and this routing is usually based on a single fixed routing path for each flow. In this paper, we consider the more general problem in which the routing of flows, possibly over multiple paths per flow, is an optimization parameter in the bandwidth allocation problem. Our goal is to determine a routing assignment for each flow so that the bandwidth allocation achieves optimal utility max-min fairness with respect to all feasible routings of flows. We present evaluations of our proposed multi-path utility max-min fair allocation algorithms on a statistical traffic engineering application to show that significantly higher minimum utility can be achieved when multi-path routing is considered simultaneously with bandwidth allocation under utility max-min fairness, and this higher minimum utility corresponds to significant application performance improvements.**

I. Introduction

Bandwidth allocation is a fundamental problem in various areas of networking. In this paper, we consider a general allocation problem in which a network consisting of links with fixed capacity is given along with a set of flows between source and destination pairs. The problem is to allocate a rate or bandwidth to each flow without exceeding link capacity. On one hand, we would like to improve the overall network utilization by maximizing the total throughput from all flows. On the other hand, fairness among flows must be maintained to guarantee the performance of individual flows. Therefore, an important goal of bandwidth allocation is to maximize the utilization of network resources while sharing the resources in a fair manner among network flows.

To strike a balance between fairness and throughput, a widely studied criterion in the network community is the notion of *max-min fairness* [6]. An allocation of bandwidths or rates is said to be max-min fair if it is not possible to increase the bandwidth of a flow without decreasing another already smaller flow. While max-min fair allocation treats all flows evenly and tends to allocate them with similar bandwidths, many variants [19], [8], [25] of max-min fair allocation have been proposed to differentiate the bandwidth requirements among flows. One such max-min fair variant is *weighted max-min fair allocation* [6], which assigns a weight to each flow. According to the weights, a flow would receive a bandwidth allocation proportional to its weight to gain the same fairness as others. Therefore, by giving varied weights to flows, the bandwidth requirement of flows can be differentiated. However, as shown in [8], those traditional max-min bandwidth allocations will often result in significant disparity in the actual throughput or performance of a flow, despite a "fair" allocation of bandwidth. Therefore, to further capture the general and possibly non-linear relationship between bandwidth allocation and the throughput of a flow, utility functions [27] were introduced as a general performance measure, and *utility max-min fair allocation* [8] was formulated to optimize for the max-min fairness of flows in terms of their utilities.

Nevertheless, the majority of work on max-min fairness has been limited to the case where the routing of flows has already been defined and this routing is usually based on a single fixed routing path for each flow. Although this setup simplifies the problem by decoupling the complicated flow routing problem from bandwidth allocation, the utilities that can be achieved by flows are unnecessarily hamstrung by routing decisions that have been fixed, ignoring potentially better allocations that could be achieved if optimal routing and bandwidth allocation were solved together simultaneously, and if the path diversity can be exploited by splitting traffic over multiple paths (e.g. by using MPLS tunnels). Therefore, in this paper, we consider the more general problem in which the routing of flows, possibly over multiple paths per flow, is an optimization parameter in the bandwidth allocation problem. Our goal is to determine a routing assignment for each flow so that the bandwidth allocation achieves optimal utility max-min fairness with respect to all feasible routings of flows. We call the resulting bandwidth allocation and optimal routing as a *multi-path utility max-min fair allocation*.

As we know, most max-min fair allocation formulations are based on some iterative water-filling algorithm [6]. In each iteration, the algorithm aims to maximize the allocation of all flows. The flows whose bandwidth cannot be further increased are then identified as saturated and are fixed for the remaining iterations. However, we face several new challenges when we consider the multi-path routing of flows as a part of the optimization problem. In particular, max-min fair allocation algorithms for the fixed single-path case generally rely on some notion of bottleneck link that determines the maximum common utility. However, under simultaneous multi-path routing optimization, the bandwidth allocation of a flow is not necessarily throttled by a certain link because it may be possible to re-route flows over different combinations of multiple paths to achieve higher utilities. To achieve global

978-1-4244-3876-1/09/$25.00 ©2009 IEEE

optimality, flows may need to be re-routed along different paths at each iteration. Such additional degrees of freedom make our more general problem significantly harder.

To the best of our knowledge, our combined optimal multi-path routing and bandwidth allocation problem under utility max-min fairness has not been solved previously. Specifically, the main contributions of this paper are as follows: First, we present a global optimization algorithm that is guaranteed to find an optimal routing and bandwidth allocation that can achieve optimal utility max-min fairness with respect to all feasible routings of flows, including multi-path routings. Second, we propose a fast fully polynomial iterative ϵ-approximation algorithm that can be efficiently implemented using a linear program solver. Finally, we apply these algorithms to a statistical traffic engineering application [10] in which historical traffic distributions are used as utility functions to model expected future traffic demands. We evaluated this statistical traffic engineering application on the actual network topology and traffic trace data of a public Internet backbone network, namely the Abilene network. Our evaluations show that significantly higher minimum utility and lower excess demand can be achieved when multi-path routing is considered simultaneously with bandwidth allocation.

The remainder of this paper is organized as follows. First, Section II reviews related work. We next briefly provide in Section III background material on max-min allocation, utility functions, and multi-path routing. After we give the formal definition of our multi-path utility max-min fair allocation problem in Section IV, we present new algorithms for solving this problem in Section V. Then Section VI evaluates our multi-path utility max-min allocation algorithms on a statistical traffic engineering application, with results demonstrating significant improvements in utilities that can be achieved. Finally, Section VII concludes the paper.

II. Related Work

Max-min fairness has been a widely-studied measure of fairness in the network community. However, the vast majority of work on max-min fairness has focused on the problem where routing decisions have already been fixed, often based on a single fixed routing path per flow. Several centralized solutions based on global knowledge of the network have been developed [1], [6], [16]. Distributed algorithms [2], [4], [5], [14] have also been proposed to achieve max-min fairness by adjusting flow rates based on limited link states and local flow information. In addition, several max-min fair variants have been studied, such as proportional max-min fairness [19] and utility max-min fairness [8]. In particular, utility max-min has been applied to several application-oriented allocation problems, such as flow control [21], link resource [24], etc.

As discussed in Section I, the simultaneous optimal multi-path routing and bandwidth allocation problem under the general setting of utility max-min fairness has not been solved before. Even the routing problem in the context of traditional (weighted) max-min fair bandwidth allocation is rarely discussed in the previous literature. The fair bandwidth allocation for single source flows was first studied by [23]. [20] and [11] proposed approximation algorithms to find unsplittable flow routings. The fair bandwidth allocation problem has also been studied in the online setting where a route is assigned to each flow when it arrives. [9] proposed a heuristic routing algorithm which selects the best single route for a new flow based on link congestions, such that the max-min bandwidth allocation is maximized after the flow is added. [12] developed an approximation algorithm that could achieve a max-min fair allocation with $O(\log^2 n \log^{1+\epsilon} U/\epsilon)$-competitive ratio. This bound was further improved in [7]. Finally, multi-path routing under fair bandwidth allocation has been studied [3], [13], [17], [22] as well. But majority of works [17], [22] consider routing as input rather than an optimization parameter. While we consider [3] and [13] have the closest problem formulation to us, both works only considered the weighted max-min case instead of the more general utility max-min problem with arbitrary utility functions. As shown in [8] and Figure 1, utility functions simply cannot be capture as a linear line, and it is non-trivial to extend the solution from weights to nonlinear functions.

III. Background

In this paper, we consider a network consists of N nodes connected by M links $\ell = (\ell_0, \ell_1, \ldots, \ell_{M-1})$ with link capacity $c(\ell_i)$ for any link ℓ_i. Given n commodities $\Gamma = (C_0, C_1, \ldots, C_{n-1})$ where C_i represents the flow from node s_i to t_i, our objective is to decide an allocation vector r whose component r_i is the rate for commodity C_i. Notice, the terms of flow and commodity are used interchangeably in the paper.

A. Max-Min Fair Bandwidth Allocation

Max-min fair is one of the most widely-used fairness criteria in bandwidth allocation. Its general definition is as follows:

***Definition** 1 (Max-min fair bandwidth allocation):* An allocation vector $r = (r_0, r_1, \ldots, r_{n-1})$ is max-min fair when any component r_i of r cannot be increased without decreasing some already smaller or equal component r_k ($r_k \leq r_i$).

In previous works, commodities were restricted to use a given routing path. Thus, the set of feasible bandwidth allocations is defined as follows:

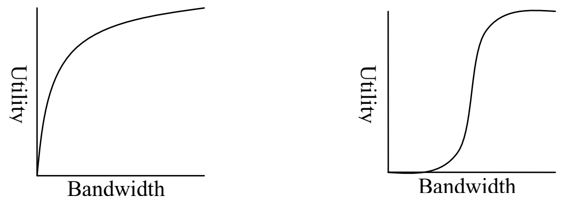
***Definition** 2 (Feasible bandwidth allocation):* A feasible bandwidth allocation $r = (r_0, r_1, \ldots, r_{n-1})$ assigns rate r_i to commodity C_i such that no link in the network is congested:

$$\sum_{\forall C_i \text{ uses } \ell_j} r_i \leq c(\ell_j), \forall \ell_j \in \ell \qquad (1)$$

Section III-C provides the definition of feasible allocation under the more general setting where the routing of flows is not known and a flow can be routed over multiple paths.

B. Utility Functions

Utility functions were first introduced into the bandwidth allocation problem by [27] to capture the performance of application flows. For example, elastic applications, including traditional data applications like emails and file transfers, are



(a) Elastic utility (b) Real-time utility

Fig. 1. Utility functions for different application classes.

best described by a convex utility function. In contrast, a real-time application could be modeled by a nearly single-step function because of its sensitivity to bandwidth requirement. As shown in Figure 1, from those application flows, their performance simply cannot be capture by a weight or linear function.

Therefore, utility max-min fairness based on a set of utility functions $\phi = \{\phi_0, \phi_1, \ldots, \phi_{n-1}\}$ was introduced as an alternative performance measure. Each function $\phi_i \in \phi$ computes the utility for commodity C_i delivered under its allocated bandwidth r_i as

$$\mu_i = \phi_i(r_i), \forall i \in n \tag{2}$$

In this paper, we assume utility functions are strictly increasing function over the domain range $[0, 1]$; that is, $\phi_i(k) < \phi_i(k'), \forall k' > k$ and $0 < \mu_i < 1, \forall i$. Thus, the inverse of a utility function is also well-defined as

$$r_i = \phi_i^{-1}(\mu_i), \forall i \in n \tag{3}$$

Accordingly, a utility max-min allocation corresponding to a given set of utility functions is defined as follows:

***Definition** 3 (Utility max-min bandwidth allocation):* A utility max-min allocation is a feasible bandwidth allocation vector $r = (r_0, r_1, \ldots, r_{n-1})$ where any component r_i of r cannot be increased without decreasing some component r_k with equal or smaller utility ($\phi_k(r_k) \le \phi_i(r_i)$).

C. Multi-Path Routing

In this paper, we consider the bandwidth allocation problem under multi-path routing where a commodity can use arbitrary routing, and its traffic can be split over multiple paths. A general formulation for an arbitrary routing assignment is $\mathbf{R}$ where $\mathbf{R}_{ij}$ is the fraction of traffic from commodity C_i routed on link ℓ_j, and we say a routing assignment $\mathbf{R}_{ij}$ is feasible if and only if it satisfies the following constraint.

***Definition** 4 (Feasible multi-path allocation):* A feasible multi-path bandwidth allocation vector $r = (r_0, r_1, \ldots, r_{n-1})$ assigns rate r_i to commodity C_i where r can be realized by a feasible routing assignment $\mathbf{R}$ without violating the flow conservation and/or overloading the network, such that

$$\sum_{\ell_j \in E^+(k)} r_i R_{ij} - \sum_{\ell_j \in E^-(k)} r_i R_{ij} = \begin{cases} r_i & \text{if } k = s_i \\ -r_i & \text{if } k = t_i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in n \text{ and } k \in N \tag{4}$$

$$\sum_{\forall C_i} R_{ij} \cdot r_i \le c(\ell_j) \qquad \forall j \in M \tag{5}$$

$$R_{ij} \ge 0 \qquad \forall i \in n \text{ and } j \in M \tag{6}$$

where $E^+(k)$ and $E^-(k)$ represent the set of incoming and outgoing links at node k.

IV. Multi-path Utility Max-min Allocation

A. Motivation

We start with an example to illustrate the difference in bandwidth allocation when considering utility functions and multi-path routing. Figure 2 shows a network with four nodes interconnected by 10-units bandwidth links. The network has three commodities, (A, D), (B, D) and (C, D), and their utility functions corresponding to a given bandwidth allocation r are $\phi_1(r) = r^2/100$, $\phi_2(r) = (r^2 + 12r)/100$ and $\phi_3(r) = (3r + 40)/100$, respectively. Notice that some of the utility functions given are non-linear. In this example, both commodities (B, D) and (C, D) have only one possible routing path each. However, commodity (A, D) has two possible routing paths, $A \to B \to D$ and $A \to C \to D$.

First, we consider the traditional max-min allocation problem where commodities are routed over a single path, and we assume the path specified for commodity (A, D) is $A \to B \to D$. Under the single-path max-min allocation defined by Definition 1 and Definition 2, the max-min fair allocation vector is $(5, 5, 10)$. This arises by assigning a common 5-units of bandwidth to all three commodities in the first iteration, which would saturate both commodities, (A, D) and (B, D), respectively. The third commodity (C, D) is increased to a full 10-units of bandwidth in the second iteration. Corresponding to this max-min allocation, the resulting utilities for commodities (A, D), (B, D), and (C, D) are $\phi_1(5) = 0.25$, $\phi_2(5) = 0.84$, and $\phi_3(10) = 0.70$, respectively.

On the other hand, under the single-path *utility* max-min allocation defined by Definition 3, the utility max-min fair allocation vector is $(6.8, 3.2, 10)$, and the corresponding utilities achieved are $\phi_1(6.8) = 0.47$, $\phi_2(3.2) = 0.47$, and $\phi_3(10) = 0.70$, respectively. This arises by allocating bandwidth to achieve the maximum common utility for all three commodities in the first iteration, which in this example is 0.47. However, to achieve this maximum common utility, more bandwidth for example needs to be allocated to the first commodity (A, D) than to the second commodity (B, D). Again, the bandwidth allocation can be further increased for the third commodity (C, D) to achieve a higher utility. Comparing with the traditional max-min allocation, clearly the utility max-min allocation achieves better fairness with respect to the given utility functions since utility max-min allocation specifically aims to do so. In particular, the minimum utility is increased from $\phi_1(5) = 0.25$ to $\phi_1(6.8) = 0.47$.

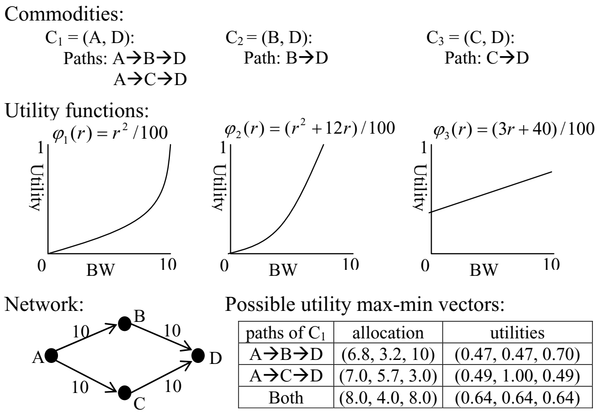


Fig. 2. Multi-path utility max-min example.

When we further consider utility max-min allocation under *multi-path routing*, there could be three possible allocation results, depending on the choice of routing paths available to commodity (A, D). If we choose the path $A \rightarrow B \rightarrow D$ to route commodity (A, D), then the allocation would be the same as the previous utility max-min allocation. However, if we choose the path $A \rightarrow C \rightarrow D$ to route commodity (A, D), then the utility max-min fair allocation would be $(7, 5.7, 3)$, and the corresponding utilities would be $(0.49, 1.00, 0.48)$. Finally, if we choose to use both paths to route commodity (A, D), then the utility max-min allocation would be $(8, 4, 8)$, where commodity (A, D) would be allocated 6 units of bandwidth along the path $A \rightarrow B \rightarrow D$ and 2 units of bandwidth along the path $A \rightarrow C \rightarrow D$. As a result, the corresponding utilities to the allocation would be $(0.64, 0.64, 0.64)$.

Comparing the three utility allocation results, not only does the last allocation fully utilize available capacities along all links, it also achieves better fairness than the allocation results using a single routing path because no one commodity can increase its utility by decreasing another one. In fact, by considering multi-path routing, it guarantees to achieve equal or better fairness than the traditional single-path bandwidth allocation because the solution space with single-path routing is only a subset of the solution space with multi-path routing. Therefore, utility max-min allocation under multi-path routing provides a more powerful and general fair allocation framework, but it is also a much more complicated problem because it couples flow routing and bandwidth allocation together.

B. Definitions

As shown by the example in Figure 2, a utility max-min allocation exists for each given routing assignment. We define a local multi-path utility max-min allocation as the utility max-min allocation with respect to a given routing assignment $\mathbf{R}$ in Definition 5. Then according to the utility order defined in Definition 7, we define the optimal multi-path utility max-min allocation as the largest one among all local allocations in Definition 8. Finally, we say an allocation is an ϵ-approximation to the optimal solution if it satisfies the requirement defined in Definition 9.

TABLE I

VARIABLES USED IN MULTI-PATH UTILITY MAX-MIN ALGORITHMS.

ℓ	a set of links $\ell_0, \ell_1, \dots, \ell_{M-1}$ with capacity $c(\ell)$
Γ	set of commodities $C_0, C_1, \dots, C_{n-1}$ where C_i is a flow from node s_i to t_i
ϕ_i	utility function for commodity C_i
r_i	bandwidth allocated to commodity C_i
$\mathbf{R}$	a feasible routing assignment where $\mathbf{R}_{ij}$ is the fraction of traffic from commodity C_i routes on link ℓ_j
μ_i	utility achieved by commodity C_i
d_i	a temporary rate assignment for C_i
π	a iteration counter starts from 1
Γ^{π}_{SAT}	set of commodities identified as saturated at iteration π
Γ^{π}_{UNSAT}	$\Gamma^{\pi}_{UNSAT} = \Gamma \backslash \bigcup_{k=0}^{\pi-1} \Gamma^{k}_{SAT}$
μ^{π}_{max}	the maximum common utility achieved at iteration π

***Definition** 5 (Local multi-path utility max-min allocation):* A local multi-path utility max-min allocation is a feasible multi-path allocation vector $r = (r_0, r_1, \dots, r_{n-1})$ where any component r_i of r cannot be increased without decreasing some component r_k with equal or smaller utility ($\phi_k(r_k) \leq \phi_i(r_i)$) under some routing assignment $\mathbf{R}$.

***Definition** 6 (Utility-ordered allocation vector):* Given a allocation $r = (r_0, r_1, \dots r_{n-1})$, we define a corresponding utility-ordered allocation vector $\overline{r} = (r_{i_0}, r_{i_1}, \dots, r_{i_{n-1}})$, such that $\phi_{i_k}(r_{i_k}) \leq \phi_{i_{k+1}}(r_{i_{k+1}})$ for $k = 0 \dots n-2$, where f_i is the utility function of commodity C_i.

***Definition** 7 (Utility order ($>_u$)):* Given allocation vectors a, b and their utility-ordered allocation $\overline{a} = (a_{i_0}, a_{i_1}, \dots, a_{i_{n-1}})$ and $\overline{b} = (b_{j_0}, b_{j_1}, \dots, b_{j_{n-1}})$, we say $a >_u b$ if only if there is some m such that $\phi_{i_k}(a_{i_k}) = \phi_{j_k}(b_{j_k})$ for $0 \leq k < m$ and $\phi_{i_m}(a_{i_m}) > \phi_{j_m}(b_{j_m})$.

***Definition** 8 (Optimal multi-path utility max-min allocation):* The optimal multi-path utility max-min fair allocation is a feasible multi-path allocation vector that is the largest among all local multi-path utility max-min allocations under the ordering defined by $>_u$. In other words, $r = (r_0, r_1, \dots, r_{n-1})$ is an optimal multi-path utility max-min vector if any commodity C_i cannot be increased without decreasing the rate of another C_j, such that $\mu_j \leq \mu_i$ where $\mu_j = \phi_j(r_j)$ and $\mu_i = \phi_i(r_i)$ under any routing assignment.

***Definition** 9 (ϵ-approximation to the optimal allocation):* Let $r = (r_0, r_1, \dots, r_{n-1})$ be the optimal multi-path utility max-min vector, and $\overline{r} = (r_{i_0}, r_{i_1}, \dots, r_{i_{n-1}})$ is its utility-ordered vector. We say another utility max-min allocation vector $r' = (r'_0, r'_1, \dots, r'_{n-1})$ with its utility-ordered vector $\overline{r'} = (r'_{j_0}, r'_{j_1}, \dots, r'_{j_{n-1}})$ is ϵ-approximation to the optimal multi-path utility max-min allocation r if there is some m such that $\phi_{j_k}(r'_{j_k})(1+\epsilon) \geq \phi_{i_k}(r_{i_k})$ for $0 \leq k < m$ and $\phi_{j_m}(r'_{j_m}) > \phi_{i_m}(r_{i_m})$.

V. Optimal Multi-Path Utility Max-Min

In this section, we first provide a general problem formulation to achieve the optimal multi-path utility max-min allocation. We then provide a linear programming (LP) formulation solution to achieve an approximate optimal solution. Table V summarizes all the variables used in these algorithms.

A. OPT_MP_UMMF

The basic idea to solving a utility max-min allocation problem is to iteratively increase the utility of all commodities and determine the maximum common utility that can be achieved in each iteration. Commodities that reached their maximum utilities are tagged as saturated and removed from the water-filling process by fixing their corresponding utility.

To formulate our problem into an iterative form, we have an iterative optimization algorithm, OPT_MP_UMMF, which consists of the following steps. The algorithm starts with initializing $\pi = 1$, $\Gamma^0_{SAT} = \emptyset$, and stops when all commodities are identified as saturated (i.e., $\Gamma^\pi_{UNSAT} = \emptyset$).

Step 1: Find the maximum common utility μ^π_{max} that can be achieved by all unsaturated commodities.

$$\text{maximize } \mu^\pi_{max} \tag{7}$$
$$\text{subject to} \tag{8}$$
$$d_i = \phi_i^{-1}(\mu_i), \forall C_i \in \bigcup_{k=0}^{\pi-1} \Gamma^k_{SAT} \tag{9}$$
$$d_i = \phi_i^{-1}(\mu^\pi_{max}), \forall C_i \in \Gamma^k_{UNSAT} \tag{10}$$
$$\sum_{\forall C_i} \mathbf{R}_{ij} \cdot d_i \leq c(\ell_j), \forall \ell_j \in \ell \tag{11}$$

In the formulation, the first two constraints give the bandwidth requirement of each commodity. In particular, Equation 9 sets the bandwidth for the saturated commodities as their previously assigned utility, while Equation 10 sets the bandwidth of the unsaturated commodities to the current maximum common utility. Finally, there must be a feasible routing assignment $\mathbf{R}$ that can carry the bandwidth requirement d by satisfying the constraint in Equation 11.

Step 2: Identify newly saturated commodities, Γ^π_{SAT}, by testing each unsaturated commodity $C_i \in \Gamma^\pi_{UNSAT}$ with the following optimization problem such that commodity C_i is saturated if its utility cannot be increased by any routing.

$$\text{maximize } \tau \tag{12}$$
$$\text{subject to} \tag{13}$$
$$d_j = \phi_j^{-1}(\mu_j), \forall C_j \in \bigcup_{k=0}^{\pi-1} \Gamma^k_{SAT} \tag{14}$$
$$d_j = \phi_j^{-1}(\mu^\pi_{max}), \forall C_j \in \Gamma^\pi_{UNSAT} \backslash C_i \tag{15}$$
$$d_i = \phi_i^{-1}(\mu^\pi_{max} + \tau) \tag{16}$$
$$\sum_{\forall C_i} \mathbf{R}_{ij} \cdot d_i \leq c(\ell_j), \forall \ell_j \in \ell \tag{17}$$

The above optimization problem fixes the rates of all commodities in Equations 14 and 15, except the commodity C_i being tested. It then finds the maximum utility that can be increased for commodity C_i in Equation 16 while there still exists some feasible routing $\mathbf{R}$ to carry all commodities. Therefore, if $\tau < 0$, we cannot increase the utility of commodity C_i by any routing and $\Gamma^\pi_{SAT} = \Gamma^\pi_{SAT} \bigcup C_i$.

Step 3: Assign the utility and bandwidth for each newly saturated commodity $C_i \in \Gamma^\pi_{SAT}$.

$$\mu_i = \mu^\pi_{max}, r_i = \phi_i^{-1}(\mu^\pi_{max})$$

The last step is to assign the bandwidth allocation for the newly saturated commodities and move them into the saturated set Γ^π_{SAT}. Once commodities are in the saturated set, their bandwidth allocations and utilities will not be changed, but their routing paths still could be altered to better utilize residual capacities and achieve higher utilities for the remaining unsaturated commodities in later iterations.

Next, we prove the correctness of the above optimal algorithm.

Theorem 1: The allocation vector r returned by OPT_MP_UMMF is optimal multi-path utility max-min.

Proof: According to Definition 8, we have to prove that the rate of any commodity C_i cannot be increased without decreasing the rate of another C_j, such that $\mu_j \leq \mu_i$ where $\mu_j = \phi_j(r_j)$ and $\mu_i = \phi_i(r_i)$ under any routing $\mathbf{R}$.

To prove by contradiction, we assume commodity $C_i \in \Gamma^\pi_{SAT}$ is identified as saturated at iteration π and its bandwidth can be increased without decreasing the bandwidth of any commodity C_j which has less or equal utility than C_i.

First of all, according to the OPT_MP_UMMF algorithm, if the utility of some commodity $C_j \in \Gamma^{\pi'}_{SAT}$ is less than or equal to the utility of commodity C_i, then $C_j \in \bigcup_{k=0}^{\pi} \Gamma^k_{SAT}$. This is because their utilities are assigned to be the maximum common utility of the iteration when they are identified as a saturated commodity, and μ_{max} is a non-decreasing vector. Thus, if $\mu_j \leq \mu_i$, then $\pi' \leq \pi$ and $C_j \in \bigcup_{k=0}^{\pi} \Gamma^k_{SAT}$.

Then in Step 2 of the OPT_MP_UMMF algorithm, at iteration π, when we test for commodity C_i, we set the rate of any commodity $C_j \in \bigcup_{k=0}^{\pi} \Gamma^k_{SAT}$ to be their final allocation r_j and the rate of any commodity $C_j \notin \bigcup_{k=0}^{\pi} \Gamma^k_{SAT}$ to be the maximum common utility at iteration π, μ^π_{max}, which is also the same as r_i because $r_i = \mu^\pi_{max}$. However, we still cannot increase the bandwidth of commodity C_i by any feasible routing. Therefore, we cannot increase the utility of commodity C_i by decreasing the bandwidth of any commodity C_j with greater utility C_i. In other words, we have to increase the utility of commodity C_i by decreasing the bandwidth of some commodity C_j with less or equal utility than C_i, and that is in contradiction to our assumption. ■

B. ϵ-OPT_MP_UMMF

The OPT_MP_UMMF algorithm is a non-linear optimization problem because the utility functions used in Step 1 can be non-linear. Therefore, we propose a fast fully polynomial approximation algorithm, ϵ-OPT_MP_UMMF, which uses a binary search formulated as linear programming to find the maximum common utility. In addition, we re-formulate the optimization problems in ϵ-OPT_MP_UMMF such that they can all be solved as a well-defined Maximum Concurrent Flow (MCF) [18] routing problem. Given set of commodity C with its demand function, $d(C)$, and a set of links ℓ with capacities, $c(\ell)$, a MCF solver finds a routing $\mathbf{R}$ to maximize λ, which is the common fraction of demand for each commodity that can be routed with the given link capacities.

The modified Step 1 of the ϵ-OPT_MP_UMMF algorithm is as follows.

Step 1: Binary search the utility domain to achieve the maximum common utility μ^{π}_{max} for unsaturated commodities. The initial values of variables are $\mu_{high} = 1, \mu_{low} = 0, \lambda = 0$.

while $\mu_{high} - \mu_{low} \geq \delta$ and $\lambda < 1$
  $\mu^{\pi}_{max} = (\mu_{high} + \mu_{low})/2$
  $d_i = \phi_i^{-1}(\mu_i), \forall C_i \in \bigcup_{k=0}^{\pi-1} \Gamma^k_{SAT}$
  $d_i = \phi_i^{-1}(\mu^{\pi}_{max}), \forall C_i \in \Gamma^{\pi}_{UNSAT}$
  $(\lambda, \mathbf{R}) = MCF(C, d, \ell, c)$
  if $\lambda < 1$
    $\mu_{high} = \mu^{\pi}_{max}$
  else
    $\mu_{low} = \mu^{\pi}_{max}$
  endif
endwhile

The binary search procedure starts with guessing the maximum common utility as a value in the utility domain range 0 to 1. Then with a given common utility μ, we verify if μ can be achieved by finding a feasible routing to carry the corresponding bandwidth allocation; that is, the required bandwidth to reach common utility μ for unsaturated commodities and to satisfy previously assigned utilities for saturated commodities. According to the Maximum Concurrent Flow (MCF) problem, by assigning the demand of commodities as the required bandwidth for the common utility μ, there exists a feasible routing to achieve the common utility μ if the λ returned by the MCF solver is ≥ 1.0. Because utility functions are strictly increasing, we find μ as the maximum common utility when there is no feasible routing to achieve an even greater common utility $\mu+\delta$, where δ can be an arbitrarily small value depending on the ϵ selected by the approximation algorithm.

Since the maximum common utility in found Step 1 is approximated, the utility of a saturated commodity could be further increased. Thus, we must change Step 2 accordingly to guarantee the utility of a saturated commodity can only be increased by at most a fraction of ϵ to its current utility.

Step 2: Identify newly saturated commodities Γ^{π}_{SAT} by verifying if each unsaturated commodity C_i meets the following saturation test in which $C_i \in \Gamma^{\pi}_{SAT}$ if and only if the utility of C_i cannot be further increased by ϵ% of its current utility.

$d_j = \phi_j^{-1}(\mu_j), \forall C_j \in \bigcup_{k=0}^{\pi-1} \Gamma^k_{SAT}$
$d_j = \phi_j^{-1}(\mu^{\pi}_{max}), \forall C_j \in \Gamma^{\pi}_{SAT} \backslash C_i$
$d_i = \phi_i^{-1}(\mu^{\pi}_{max} \cdot (1+\epsilon))$
$(\lambda, \mathbf{R}) = MCF(C, d, \ell, c)$
if $\lambda < 1$
  $\Gamma^{\pi}_{SAT} = \Gamma^{\pi}_{SAT} \bigcup C_i$
endif

In the above Step 2, we determine a commodity as saturated if its utility cannot be increased by a fraction of ϵ to the current utility under any feasible routing assignment. Again, we use a MCF solver to verify if there exists a feasible routing to increase the utility of a commodity by assigning its demand to the required bandwidth for achieving an additional ϵ fraction of its current utility. If the λ returned from a MCF solver is less than 1, then we know that the utility of the commodity cannot be increased by more than ϵ with any feasible routing, and therefore the commodity should be identified as saturated.

Notice, the difference between δ and ϵ is that δ means increasing limit for common utility of unsaturated commodities, while ϵ represents the increasing limit of the utility of individual unsaturated commodities. Thus, if we can increase the common utility by δ, it does not mean the utility of each unsaturated commodity cannot be increased by δ. Therefore, δ has to be chosen carefully with respect to ϵ to guarantee that there is at least one commodity whose utility cannot be increased by ϵ when the common utility cannot be increased by δ. We construct such δ based on ϵ in Lemma 1.

Lemma 1: For a given ϵ, there exists some δ such that there is at least one commodity identified as saturated at every iteration in the ϵ-OPT_MP_UMMF algorithm.

Proof: The basic idea is to construct δ based on ϵ. The detailed proof is given in the appendix. ∎

Finally, we show that the ϵ-OPT_MP_UMMF algorithm achieves an ϵ-approximation allocation to the optimal solution, and the algorithm eventually terminates as following.

Theorem 2: ϵ-OPT_MP_UMMF achieves ϵ-approximation to the optimal multi-path utility max-min allocation.

Proof: Let $\mu^{*\pi}_{max}$ and μ^{π}_{max} be the maximum common utility achieved by the optimal and ϵ-OPT_MP_UMMF algorithm,respectively. We say ϵ-OPT_MP_UMMF algorithm achieves ϵ approximation to the optimal if there exist some m such that $\mu^{\pi}_{max} \cdot (1+\epsilon) \geq \mu^{*\pi}_{max}, \forall \pi < m$ and $\mu^{m}_{max} > \mu^{*m}_{max}$.

We prove by induction. After first iteration, clearly $\mu^{1}_{max} \cdot (1+\epsilon) \geq \mu^{*1}_{max}$, because otherwise it would contradict the condition of saturation test. After second iteration, let μ'^{2}_{max} be the optimal maximum common utility can be achieved when the utility of commodity $C_i \in \Gamma^1_{SAT}$ is assigned to be μ^1_{max}. Then $\mu'^{2}_{max} > \mu^{*2}_{max}$ because $\mu^{1}_{max} < \mu^{*1}_{max}$. Since $\mu^{2}_{max} \cdot (1+\epsilon) > \mu'^{2}_{max}$ and $\mu'^{2}_{max} > \mu^{*2}_{max}$, $\mu^{2}_{max} \cdot (1+\epsilon) > \mu^{*2}_{max}$.

After π iterations, if $\mu^{\pi-1}_{max} > \mu^{*\pi-1}_{max}$, we find $m = \pi$. Otherwise, we know $\mu^{k}_{max} < \mu^{*k}_{max}, \forall k < \pi$. Let μ'^{π}_{max} be the optimal maximum common utility can be achieved when the utility of commodity $C_i \in \Gamma^k_{SAT}$ is assigned to be $\mu^k_{max}, \forall k < \pi$. Then $\mu'^{\pi}_{max} > \mu^{*\pi}_{max}$, because the utility assignment for any saturated commodity $C_i \in \bigcup_{k=1}^{\pi} \Gamma^k_{SAT}$ is less than its optimal utility μ^{*k}_{max}. Again, since the saturation test guarantees $\mu^{\pi}_{max} \cdot (1+\epsilon) > \mu'^{\pi}_{max}$ and we already know $\mu'^{\pi}_{max} > \mu^{*\pi}_{max}$, $\mu^{\pi}_{max} \cdot (1+\epsilon) > \mu^{*\pi}_{max}$ holds for at any iteration π as long as $\mu^{k}_{max} < \mu^{*k}_{max}, \forall k < \pi$. ∎

Theorem 3: ϵ-OPT_MP_UMMF algorithm terminates after at most n iterations, where n is the number of commodities.

Proof: According to Lemma 1, at least one commodity is identified as saturated after each iteration. Since the algorithm terminates after all commodities are saturated, it has at most n iterations, where n is the number of commodities. ∎

Conclude all, the overall ϵ-OPT_MP_UMMF algorithm is also polynomial time solvable with respect to the size of network because (1) there can be at most n iterations, (2) the number of binary searchers at each iteration is a constant with respect to ϵ, and (3) each binary search step involves a MCF problem that can be solved in polynomial time.

VI. Evaluation

To demonstrate the improvements possible when path selection is considered as part of the optimization problem, we use a specific statistical traffic engineering problem, COPLAR [10], as an example application. We note that the focus of this work is on the multi-path utility max-min fair allocation algorithms and on their optimality that can be analytically proven, not on this example traffic engineering problem per se. We emphasize that the proposed optimization framework can be applied to a number of other network applications as well.

In the following, we first briefly introduce the example statistical traffic engineering application, COPLAR [10], and our evaluation setup. We then compare our multi-path utility max-min allocation results with existing max-min allocation solutions to demonstrate the improvements of our approach.

A. COPLAR - A Statistical Traffic Engineering Application

COPLAR [10], which stands for [c]oarse [op]tica[l] circuit switching with [a]daptive [r]e-routing, is a new network paradigm for optical networks. As Internet traffic continues to grow unabated at an exponential rate, it is unclear whether or not the existing packet routing network architecture based on electronic routers will continue to scale at the necessary pace. To reduce work load at electronic routers and take advantage of the abundance of optical fiber capacity, the central idea of COPLAR is to route traffic on coarse optical circuit switching by default and only adaptively re-route exceeding over spare circuit capacity when necessary. Since circuit switching doesn't require the participation of any intermediate electronic routers, COPLAR relies on a bandwidth allocation algorithm to find a set of static or coarse grind provisioned circuits that can cover majority of the network traffic.

To provision a circuit configuration that can maximize the likelihood of having enough bandwidth, COPLAR uses historical traffic distributions as utility functions to model expected future traffic demands. As observed in [26], historical traffic demands during a particular time of day (e.g. 11am-Noon on a weekday) are a good indicator of expected future traffic demands over the same time of day. The flows considered are at the level of OD (Origin-Destination) pairs where traffic between each pair of ingress and egress nodes of the network is considered as a commodity.

In particular, historical traffic demands can be explicitly captured by means of demand distribution functions

$$F = (f_i(x)),$$

with each $f_i(x)$ corresponding to the probability distribution of traffic demands for commodity C_i. For each $f_i(x)$, we have a corresponding cumulative distribution function (CDF) that describes the probability distribution of a random variable X that represents that actual traffic demand. Let x be the bandwidth allocation. Then the CDF of X is given by

$$\phi_i(x) = Pr[X \leq x],$$

Fig. 3. Abilene network topology.

which corresponds to the probability the bandwidth allocation x is sufficient to satisfy the actual traffic demand X. To maximize the acceptance probability that the bandwidth allocations can satisfy the actual traffic demands for all commodities in a max-min fair manner, the problem become exactly the same as finding a a multi-path utility max-min fair bandwidth allocation. Therefore, we could apply our algorithms to find a set of routing paths or circuit configurations for the COPLAR static traffic engineering application to minimize excess demand.

B. Setup

We evaluated our bandwidth allocation solutions on the Abilene network using actual network topology and traffic trace data. As shown in Figure 3, the Abilene network has 11 nodes interconnected by 10 Gb/s links. The traffic trace data can be found in [28] as a set of traffic matrices. Each traffic matrix contains the demand rate between an OD pair at a 5-minute time interval, and it is computed based on the actual packet sampling information collected from network routers.

In the evaluations, we compare our multi-path utility max-min allocation results with the traditional weighted max-min and utility max-min allocations under single-path problem formulation. For simplicity, we use MP_UMMF to denote the results of multi-path utility max-min allocation, while UMMF and WMMF represent the single-path allocation of utility max-min and weighted max-min, respectively. Specifically, for the two single-path max-min allocation scenarios, we fixed the routing path of each commodity to its shortest path between the source and destination node. We determine the weight and utility function of each commodity based on the historical traffic measurements over 2 months period from 3/1/04 to 4/21/04. In the weighted max-min allocation scenario, the weight of a commodity is the average demand over its historical traffic measurements. On the other hand, as mentioned in Section VI-A, we determine the utility of a commodity as the empirical cumulative distribution function of its historical demands, so that the utility value is directly corresponding to the *acceptance probability* (i.e., the probability of having sufficient bandwidth allocation for a commodity). The allocation of both single-path max-min allocations can be solver by the traditional water-filling algorithm [6], while our multi-path utility max-min allocation is computed by solving the ϵ-OPT_MP_UMMF algorithm with the linear programming optimization tool CPLEX [15].

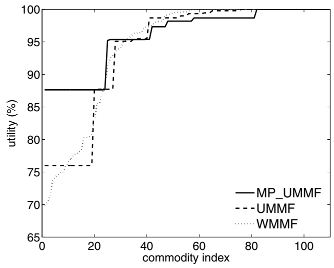


Fig. 4. The utility of individual commodities under different max-min allocations when link capacity is 1 Gb/s.

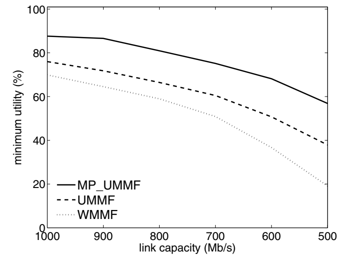


Fig. 5. The minimum utility of max-min allocations when link capacity (10 Gb/s) is scaled down by a factor of 10 to 20 with every 100 Mb/s apart.

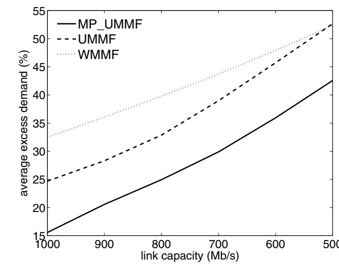


Fig. 6. The excess demand of max-min allocations when link capacity (10 Gb/s) is scaled down by a factor of 10 to 20 with every 100 Mb/s apart.

C. Max-Min Fair Allocations Comparison

Here, we compare the allocation results of MP_UMMF, UMMF and WMMF. As we know, current backbone networks are over-built to accommodate fluctuations in traffic. To emphasize the importance of bandwidth allocation when link capacity is relatively scarce, we adjusted the degree of resource contention in our evaluations by scaling down the link capacity (10 Gb/s) by a factor of 10 to 20. In the following, we first show the utilities of individual commodities achieved by each of the allocations when link capacity is 1 Gb/s. Then we compare the minimum utility and excess demand under varied degrees of resource contention.

1) Utility of Individual Commodities: We take the allocation results when link capacity is 1 Gb/s as an example to show the utility achieved for each commodity. The results under other degrees of resource contention are discussed later. Since the Abilene network has 11 nodes, there are 110 OD pairs in total. We plot the utility of each commodity in Figure 4 from the commodity with the lowest utility to the highest utility, and we have the following observations.

First, the results of utility max-min allocations appear as step functions because all commodities identified as saturated in the same iteration have the same utility/acceptance probability. But, in weighted max-min allocation, the commodities saturated at the same iteration could have different utilities because its linear utility function is only an approximation to the actual non-linear utility function of acceptance probability.

Second, we found MP_UMMF achieves much greater utility than UMMF and WMMF for most of the commodities, especially for the ones with smaller utility. For example, the minimum utility of MP_UMMF, UMMF and WMMF are 87.63%, 75.99% and 69.89%, respectively. In other words, MP_UMMF increases the minimum utility of UMMF and WMMF by a factor of 1.15x (87.63/75.99) and 1.25x (87.63/69.89), respectively. In addition, the result of UMMF is also higher than WMMF by a factor of 1.10x (87.63/69.89).

As defined by the utility order in Definition 7, the allocation results with higher minimum utility are fairer. Therefore, our multi-path utility multi-path allocation appears to improve both the fairness and total utility of bandwidth allocation. Accordingly, we further investigate the minimum utility and actual excess demand under varied resource contention in Section VI-C2 and VI-C3, respectively.

2) Minimum Utility: Figure 5 plots the minimum utility under varied degrees of resource contention when we scale down the link capacity by a factor of 10 to 20 with every 100 Mb/s apart. For all allocations, the minimum utility decreases as the degree of resource contention increased. However, MP_UMMF consistently achieves the highest utility among all allocations, especially when the degree of resource contention is higher. This is because the allocation becomes crucial as the link capacity is limited. For example, when link capacity is 500 Mb/s, the minimum utilities of MP_UMMF, UMMF and WMMF are 56.84%, 37.87% and 19.12%, respectively. In other words, the MP_UMMF is able to further increase the minimum utility of UMMF and WMMF allocations by a factor of 1.50x (56.84/37.87) and 2.97x (56.84/19.12), respectively. As shown from the figure, although UMMF optimizes for the same utility functions as MP_UMMF, it is not able to efficiently utilize bandwidth by adjusting routing. As a result, it clearly achieves much less utility than the multi-path allocation. Furthermore, the minimum utility of WMMF is ever smaller than UMMF because it is difficult to use a single weight value to capture or approximate the actual non-linear utility function. Therefore, only MP_UMMF can consistently achieve the best results by considering both multi-path routing and nonlinear utility functions.

3) Application Performance: Finally, we compare the performance of our statistic engineering application by showing the excess demand under varied link capacity. Under a given link capacity, we determine the excess demand of the traffic offering during 5-days period from 4/22/04 to 4/26/04 which is different from the traffic datasets for computing our bandwidth allocations. Figure 6 plots the average excess demand over the results at all time intervals within the 5-days period. As shown in the figure, we have lower excess demand with greater link capacity because each commodity can be allocated more bandwidth and less traffic demand would excess the bandwidth allocation. But, again MP_UMMF is still able to achieve the lowest excess demand among the three. For example, when link capacity is 1,000 Mb/s, the excess demand of MP_UMMF, UMMF and WMMF is 15.56%, 24.69% and 32.48%, respectively. In other words, MP_UMMF substantially reduces the excess demand of UMMF and WMMF by 36.98% and 52.09%, respectively. Therefore, our improvement on the utilities shown previously also transfers to better performance for our statistic traffic engineering application.

VII. Conclusion

In this paper, we considered routing as an optimization parameter in the bandwidth allocation problem. Our goal is to determine a routing assignment for each flow and allocate bandwidth to them such that the allocation of flows achieves optimal utility max-min fairness with respect to all feasible routings of flows. We presented, for the first time, an algorithm that finds the global optimal utility max-min fair allocation where the multi-path routing of flows is simultaneously decided. We also presented a fast approximation algorithm that can be efficiently implemented using a linear program solver. Finally, we apply these algorithms to a statistical traffic engineering application to show that significantly higher minimum utility can achieved when multi-path routing is considered simultaneously with utility max-min fair allocation, and this higher minimum utility translates to significant performance improvements of our statistical traffic engineering application.

Appendix

Proof of Lemma 1: For a given ϵ, there exists some δ such that there is at least one commodity identified as saturated at every iteration in the ϵ-OPT_MP_UMMF algorithm.

Proof: The basic idea is to construct δ based on ϵ. If we can find a routing to increase the utility for each previous unsaturated commodity by ϵ, there must exist a routing which can increase the utility of all previous unsaturated commodities by δ. Thus, we prove the lemma by contradiction because if all previous unsaturated commodities in Step 2 can increase its utility by more than ϵ and remain as unsaturated, then there must exist a routing to increase the utility of all previous unsaturated commodities by more then δ, which clearly contradicts to the termination condition of the binary search in Step 1.

Now we construct such a δ as the following. At any iteration π, we know the current maximum common utility is μ^{π}_{max}. For each commodity $C_k \in \Gamma^{\pi}_{UNSAT}$, if it is identified as unsaturated, there must exist a routing $\mathbf{R}^k$ to carry the addition bandwidth $\triangle d_i$ for commodity C_i where

$$\triangle d_k = \phi_k^{-1}(\mu^{\pi}_{max}(1+\epsilon)) - \phi_i^{-1}(\mu^{\pi}_{max})$$

Therefore, if all $C_k \in \Gamma^{\pi}_{UNSAT}$ are identified as unsaturated commodities, we can construct a feasible routing $\mathbf{R}'$ to carry an additional bandwidth $\triangle d'_k$ for each commodity $C_k \in \Gamma^{\pi}_{UNSAT}$ by combining all routings $\mathbf{R}^k$ where

$$\triangle d'_k = \frac{\triangle d_k}{\mid \Gamma^{\pi}_{UNSAT} \mid} \text{ and } R'_{ij} = \sum_{\forall k \in \Gamma^{\pi}_{UNSAT}} \frac{R^k_{ij}}{\mid \Gamma^{\pi}_{UNSAT} \mid}$$

Accordingly, let $\triangle \delta_k$ be the utility can be increased corresponding to the additional bandwidth $\triangle d'_k$ $\forall C_k \in \Gamma^{\pi}_{UNSAT}$.

$$\triangle \delta_k = \phi_k(\phi_k^{-1}(\mu^{\pi}_{max}) + \triangle d'_k) - \mu^{\pi}_{max}.$$

Then we choose δ as the minimum value of all $\triangle \delta_k$, so

$$\delta = \min(\triangle \delta_k, \forall C_k \in \Gamma^{\pi}_{UNSAT}).$$

As a result, if we can increase the utility of each $C_i \in \Gamma^{\pi}_{UNSAT}$ by ϵ, there must exist a feasible routing to increase the utility of all $C_i \in \Gamma^{\pi}_{UNSAT}$ by δ. Therefore, at least one commodity is identified as saturated by the δ we found. ■

References

[1] Y. Afek, Y. Mansour, and Z. Ostfeld. Convergence complexity of optimistic rate based flow control algorithms (extended abstract). In *ACM Symposium on Theory of Computing*, pages 89–98, 1996.

[2] Y. Afek, Y. Mansour, and Z. Ostfeld. Phantom: a simple and effective flow control scheme. *Computer Networks*, 32(3):277–305, 2000.

[3] M. Allalouf and Y. Shavitt. Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation. *IEEE/ACM Transaction on Networks*, 2008.

[4] B. Awerbuch and Y. Shavitt. Converging to approximated max-min flow fairness in logarithmic time. In *IEEE INFOCOM*, 1998.

[5] Y. Bartal, M. Farach-Colton, S. Yooseph, and L. Zhang. Fast, fair, and frugal bandwidth allocation in atm networks. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, 1999.

[6] D. Bertsekas and R. Gallager. Data networks, 1987. Prentice-Hall.

[7] N. Buchbinder and J. Naor. Fair online load balancing. In *SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 291–298, 2006.

[8] Z. Cao and E. W. Zegura. Utility max-min: An application-oriented bandwidth allocation scheme. In *IEEE INFOCOM*, 1999.

[9] S. Chen and K. Nahrstedt. Maxmin fair routing in connection-oriented networks. In *Euro-Parallel and Distributed Systems Conference*, 1998.

[10] J. Chou and B. Lin. Coarse optical circuit switching by default, rerouting over circuit for adaptation. *Journal of Optical Networking*, 18(1):33–50, January 2009.

[11] Y. Dinitz, N. Garg, and M. X. Goemans. On the single-source unsplittable flow problem. In *IEEE FOSC*, pages 290–299, 1998.

[12] A. Goel, A. Meyerson, and S. Plotkin. Combining fairness with throughput: online routing with multiple objectives. In *ACM Symposium on Theory of Computing*, pages 670–679, 2000.

[13] Y. T. Hou, Y. Shi, and H. D. Sherali. Rate allocation in wireless sensor networks with network lifetime requirement. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 67–77. ACM, 2004.

[14] Y. T. Hou, H. H.-Y. Tzeng, and S. S. Panwar. A generalized max-min rate allocation policy and its distributed implementation using ABR flow control mechanism. In *IEEE INFOCOM*, pages 1366–1375, 1998.

[15] ILOG. Cplex: optimization software. www.ilog.com/products/cplex/.

[16] J. M. Jaffe. Bottleneck flow control. *IEEE/ACM Transaction on Communications*, pages 954–962, 1988.

[17] E. Jin, W. Wang, and M. Palaniswami. Utility max-min fair flow control for multipath communication networks. In *ICSPCS*, December 2007.

[18] G. Karakostas. Faster approximation schemes for fractional multicommodity flow problems. In *Symposium on Discrete Algorithms*, 2002.

[19] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[20] J. M. Kleinberg, Y. Rabani, and E. Tardos. Fairness in routing and load balancing. In *IEEE FOSC*, pages 568–578, 1999.

[21] H.-W. Lee and S. Chong. A distributed utility max-min flow control algorithm. *The International Journal of Computer and Telecommunications Networking*, 50(11):1816–1830, 2006.

[22] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. *SIGCOMM Comput. Commun. Rev.*, 26(4):206–217, 1996.

[23] N. Megiddo. Optimal flows in networks with sources and sinks, 1974. Math Programming.

[24] S. Mohanty, C. Liu, B. Liu, and L. Bhuyan. Max-min utility fairness in link aggregated systems. In *HPSR workshop*, pages 1–7, 2007.

[25] B. Radunović and J.-Y. L. Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transaction on Networks*, 15(5):1073–1083, 2007.

[26] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in measuring internet backbone traffic variability:models, metrics, measurements and meaning. In *International Teletraffic Congress (ITC)*, November 2003.

[27] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

[28] Y. Zhang. Abilene traffic matrices. www.cs.utexas.edu/~yzhang/research/AbileneTM.