Multipath Routing Algorithms for Congestion Minimization

Ron Banner and Ariel Orda

Department of Electrical Engineering,
Technion – Israel Institute of Technology, Haifa 32000, Israel
banner@tx.technion.ac.il
ariel@ce.technion.ac.il

Abstract. Unlike traditional routing schemes that route all traffic along a single path, multipath routing strategies split the traffic among several paths in order to ease congestion. It has been widely recognized that multipath routing can be fundamentally more efficient than the traditional approach of routing along single paths. Yet, in contrast to the single-path routing approach, most studies in the context of multipath routing focused on heuristic methods. We demonstrate the significant advantage of optimal solutions. Hence, we investigate multipath routing adopting a rigorous (theoretical) approach. We formalize problems that incorporate two major requirements of multipath routing. Then, we establish the intractability of these problems in terms of computational complexity. Accordingly, we establish efficient solutions with proven performance guarantees.

Keywords: Routing, Congestion, Algorithms, Optimization, Combinatorics.

1 Introduction

Current routing schemes typically focus on discovering a single "optimal" path for routing, according to some desired metric. Accordingly, traffic is always routed over a single path, which often results in substantial waste of network resources. *Multipath Routing* is an alternative approach that distributes the traffic among several "good" paths instead of routing all traffic along a single "best" path.

Multipath routing can be fundamentally more efficient than the currently used single-path routing protocols. It can significantly reduce congestion in "hot spots", by deviating traffic to unused network resources, thus improving network utilization and providing load balancing [13]. Moreover, congested links usually result in poor performance and high variance. For such circumstances, multipath routing can offer steady and smooth data streams [6].

Previous studies and proposals on multipath routing have focused on *heuristic methods*. In [16], a multipath routing scheme, termed Equal Cost MultiPath (ECMP), has been proposed for balancing the load along multiple shortest paths using a simple round-robin distribution. By limiting itself to shortest paths, ECMP considerably

R. Boutaba et al. (Eds.): NETWORKING 2005, LNCS 3462, pp. 536 – 548, 2005.

reduces the load-balancing capabilities of multipath routing; moreover the *equal* partition of flows along the (shortest) paths (resulting from the round robin distribution) further limits the ability to decrease congestion through load balancing. OSPF-OMP [21] allows splitting traffic among paths *unevenly*; however, the traffic distribution mechanism is based on a heuristic scheme that often results in an inefficient flow distribution. Both [22] and [24] considered multipath routing as an optimization problem with an objective function that minimizes the congestion of the most utilized link in the network; however, they focused on heuristics and did not consider the quality of the selected paths. In [17], a scheme is presented to proportionally split traffic among several “widest” paths that are disjoint with respect to the *bottleneck links*. However, here too, the scheme is heuristic and evaluated by way of simulations.

Simulation results clearly indicate that multipath solutions obtained by *optimal* congestion reduction schemes are *fundamentally* more efficient than the solutions obtained by heuristics. For example, in Section 5, we show that if the traffic distribution mechanism in the ECMP scheme had been optimal, the network congestion would have decreased by more than *three times*; moreover, if paths other than shortest had been allowed, the optimal partition would have decreased the network congestion by more than *ten times*. Hence, the full potential of multipath routing is far from having been exploited.

Accordingly, in this study we investigate multipath routing adopting a rigorous approach, and formulate it as an optimization problem of minimizing network congestion. Under this framework, we consider two fundamental requirements. First, each of the chosen paths should usually be of satisfactory "quality". Indeed, while better load balancing is achieved by allowing the employment of paths other than shortest, paths that are substantially inferior (i.e., "longer") may be prohibited. Therefore, we consider the problem of congestion minimization through multipath routing subject to a restriction on the "quality" (i.e., length) of the chosen paths.

Another practical restriction is on the *number of routing paths per destination*, which is due to several reasons [17]: first, establishing, maintaining and tearing down paths pose considerable overhead; second, the complexity of a scheme that distributes traffic among multiple paths considerably increases with the number of paths; third, often there is a limit on the number of explicitly routing paths (such as label-switched paths in MPLS [19]) that can be set up between a pair of nodes. Therefore, in practice, it is desirable to use as few paths as possible while at the same time minimize the network congestion.

Our Results: Consider first the problem of minimizing the congestion under the requirement to route traffic along paths of "satisfactory" quality. We first show that the considered problem is NP-hard, yet admits a pseudo-polynomial solution. Accordingly, we design two algorithms. The first is an optimal algorithm with a pseudo-polynomial running time, and the second approximates the optimal solution to any desired degree of precision at the (proportional) cost of increasing its running time (i.e., an ε-optimal approximation scheme). In addition, we show that these algorithms can be extended to offer solutions to reliability-related problems.

Consider now the requirement of limiting the number of paths per destination. We show that minimizing the congestion under this restriction is NP-hard as well. Accordingly, we establish a computationally efficient 2-approximation scheme[1]. Then, we generalize the 2-approximation scheme into a bicriteria result and establish a (1+1/r)-approximation scheme that, for any given $r\geq 1$, violates the constraint on the number of routing paths by a factor of at most r. Finally, we broaden the scope of this problem and establish an efficient approximation scheme for the dual problem, which restricts the level of congestion while minimizing the number of paths per destination.

Due to space limits, several proofs and technical details are omitted from this version and can be found (online) in [4].

2 Model and Problem Formulation

A *network* is represented by a directed graph $G(V,E)$, where V is the set of nodes and E is the set of links. Let $N=|V|$ and $M=|E|$. A *path* is a finite sequence of nodes $p=(v_0,v_1,\ldots v_h)$, such that, for $0\leq n\leq h-1$, $(v_n,v_{n+1})\in E$. A path is *simple* if all its nodes are distinct. A *cycle* is a path $p=(v_0,v_1,\ldots,v_h)$ together with the link $(v_h,v_0)\in E$ i.e., $(v_0,v_1,\ldots,v_h,v_0)$.[2]

Given a source node $s\in V$ and a target node $t\in V$, $P^{(s,t)}$ is the set of (all) directed paths in $G(V,E)$ from s to t. For each path $p\in P^{(s,t)}$ and link $e\in E$, let $\Delta_e(p)$ count the number of occurrences of e in p. For example, given a non-simple path $p=(v_0,v_1,v_2,v_3,v_1,v_2,v_4)$ and a link $e=(v_1,v_2)$, we have $\Delta_e(p)=2$.

Each link $e\in E$ is assigned a *length* $l_e\in\mathbb{Z}^+$ and a *capacity* $c_e\in\mathbb{Z}^+$. We consider a *link state* routing environment, where each source node has an image of the entire network.

Definition 1: Given a (non-empty) path p, the *length* $L(p)$ of p is defined as the sum of lengths of its links, namely, $L(p)\triangleq\sum_{e\in p}l_e$.

Definition 2: Given a (non-empty) path p, the *capacity* $C(p)$ of p is defined as the capacity of its bottleneck link, namely, $C(p)\triangleq\underset{e\in p}{Min}\{c_e\}$.

Definition 3: Given are a network $G(V,E)$, two nodes $s,t\in V$ and a demand γ. A *path flow* is a real-valued function $f:P^{(s,t)}\rightarrow\mathrm{R}^+\cup\{0\}$ that satisfies the flow demand requirements, i.e., $\sum_{p\in P^{(s,t)}}f_p=\gamma$.

---

1 i.e., an algorithm that provides a solution that, in terms of congestion, is within a factor of at most 2 away from the optimum.

2 As shall be shown, all our solutions consist of *simple* paths exclusively. Cycles and non-simple paths are included in our terminology to simplify the presentation of the solution approach.

Definition 4: Given is a path flow $f{:}P^{(s,t)}\rightarrow\mathrm{R}^{+}\cup\{0\}$ over a network $G(V,E)$. A *link flow* is a real-valued function $f{:}E\rightarrow\mathrm{R}^{+}\cup\{0\}$ that satisfies, for each link $e\in E$, $f_e \triangleq \sum_{p\in P^{(s,t)}} \Delta_e(p)\cdot f_p$.

Definition 5: Given a network $G(V,E)$ and a link flow $\{f_e\}$, the value $\frac{f_e}{c_e}$ is the *link congestion factor* and the value $\max_{e\in E}\left\{\frac{f_e}{c_e}\right\}$ is the *network congestion factor*.

As noted in [3],[13],[22] the network congestion factor provides a good indication of congestion. In [4], we show that the problem of minimizing the network congestion factor is equivalent to the well-known Maximum Flow Problem [1]. Hence, when there are no restrictions on the paths (in terms of the number of paths or the length of each path), one can find a path flow that minimizes the network congestion factor in polynomial time through a standard max-flow algorithm.

We are ready to formulate the two problems considered in this study. The first problem aims at minimizing the network congestion factor subject to a restriction on the "quality" (i.e., length) of each of the chosen paths.

Problem RMP (Restricted Multipath) Given are a network $G(V,E)$, two nodes $s,t\in V$, a length $l_e>0$ and a capacity $c_e>0$ for each link $e\in E$, a demand $\gamma>0$ and a *length restriction* L for each routing path. Find a path flow that minimizes the network congestion factor such that, if $P\subseteq P^{(s,t)}$ is the set of paths in $P^{(s,t)}$ that are assigned a positive flow, then, for each $p\in P$, it holds that $L(p)\leq L$.

Remark 1: For convenience, and without loss of generality, we assume that the length l_e of each link $e\in E$ is not larger than the length restriction L. Clearly, links that are longer than L can be erased.

The next problem considers the requirement to limit the number of different paths over which a given demand is shipped while at the same time minimizing the network congestion factor.

Problem KPR (K-Path Routing) Given are a network $G(V,E)$, two nodes $s,t\in V$, a capacity $c_e>0$ for each link $e\in E$, a demand $\gamma>0$ and a restriction on the number of routing paths K. Find a path flow that minimizes the network congestion factor, such that, if $P\subseteq P^{(s,t)}$ is the set of paths in $P^{(s,t)}$ that are assigned a positive flow, then $|P|\leq K$.

Remark 2: In both problems, the source-destination pair (s,t) is assumed to be connected i.e., $\left|P^{(s,t)}\right|\geq 1$.

3 Minimizing Congestion Under Path Quality Constraints

In this section we investigate Problem RMP, i.e., the problem of minimizing congestion under path quality constraints. We begin by establishing its intractability.

Theorem 1: Problem RMP is NP-hard.

The proof [4] is based on a reduction to the *Partition Problem* [11].

3.1 Pseudo-Polynomial Algorithm for Problem RMP

The first step towards obtaining a solution to Problem RMP is to define it as a linear program. To that end, we need some additional notation.

Recall that we are given a network $G(V,E)$, two nodes $s,t\in V$, a length $l_e>0$ and a capacity $c_e>0$ for each link $e\in E$, a demand $\gamma>0$ and a length restriction L for each routing path. Let α be the network congestion factor. Denote by f_e^λ the total flow along $e=(u,v)\in E$ that has been routed from s to u through paths with a total length of λ. Finally, for each $v\in V$, denote by $O(v)$ the set of links that emanate from v, and by $I(v)$ the set of links that enter that node, namely $O(v)=\{(v,l)|(v,l)\in E\}$ and $I(v)=\{(w,v)|(w,v)\in E\}$. Then, Problem RMP can be formulated as a linear program over the variables $\left\{\left\{f_e^\lambda\right\},\alpha\right\}$, as specified in Fig 1.

Program RMP $\left(G(V,E),\{s,t\},\{l_e\},\{c_e\},\gamma,L\right)$

Minimize α (1)

Subject to:

$$\sum_{e\in O(v)} f_e^\lambda - \sum_{e\in I(v)} f_e^{\lambda-l_e} = 0 \qquad \forall v\in V\setminus\{s,t\},\forall\lambda\in[0,L] \qquad (2)$$

$$\sum_{e\in O(s)} f_e^\lambda - \sum_{e\in I(s)} f_e^{\lambda-l_e} = 0 \qquad \forall\lambda\in[1,L] \qquad (3)$$

$$\sum_{e\in O(s)} f_e^0 = \gamma \qquad (4)$$

$$\sum_{\lambda=0}^{L} f_e^\lambda \le c_e\cdot\alpha \qquad \forall e\in E \qquad (5)$$

$$f_e^\lambda = 0 \qquad \forall e\in E,\ \lambda\notin[0,L-l_e] \qquad (6)$$

$$f_e^\lambda \ge 0 \qquad \forall e\in E,\ \lambda\in[0,L] \qquad (7)$$

$$\alpha\ge 0 \qquad (8)$$

Fig. 1. Program RMP

The objective function (1) minimizes the network congestion factor. Constraints (2), (3) and (4) are nodal flow conservation constraints. Equation (2) states that the traffic flowing out of node v, which has traversed through paths $p\in P^{(s,v)}$ of length $L(p)=\lambda$, has to be equal to the traffic flowing into node v, through paths $p'\in P^{(s,u)}$ and links $e=(u,v)\in E$, such that $L(p')+l_e=\lambda$; since $\lambda\in[0,L]$, the length restriction is obeyed; finally, equation (2) must be satisfied for each node other than the source s and the target t. Equation (3) extends the validity of equation (2) to hold for traffic that encounters source s after it has already passed through paths with non-zero length. Informally, equation (3) states that "old" traffic that emanates from s *not* for the first time (through a directed cycle that contains the source s) must satisfy the nodal flow conservation constraint of equation (2), which *solely* focuses on nodes from $V\setminus\{s,t\}$. Equation (4) states that the total traffic flowing out of source s, which has traversed paths of length $L=0$, must be equal to the demand γ. Informally, equation (4) states that the total "new" traffic that emanates from the source s for the first time must satisfy the flow demand γ. Equation (5) is the link capacity utilization constraint. It states that the maximum link utilization is not larger than the value of the variable α i.e., the network congestion factor is at most α. Expression (6) rules out non-feasible flows and Expressions (7) and (8) restrict all variables to be non-negative.

We can solve Program RMP (Fig. 1) using any polynomial time algorithm for linear programming [15]. The solution to the problem is then achieved by decomposing the output of Program RMP (i.e., link flow $\{f_e^\lambda\}$) into a path flow that satisfies the length restriction L. This is done by modifying the *flow decomposition algorithm* [1] (that transforms link flows $\{f_e\}$ into path flows $\{f_p\}$) in order to consider length restrictions i.e., transform link flows with "lengths" $\{f_e^\lambda\}$ into path flows that obey the length restrictions. Due to space limits, the description of this algorithm is omitted and can be found in [4].

In the remainder of this subsection we consider the complexity of the overall solution (henceforth, Algorithm RMP), which is dominated by the complexity of Program RMP [4]. It follows from [15] that the complexity incurred by solving Program RMP is polynomial both in the number of variables $\{f_e^\lambda\}$ and in the number of constraints needed to formulate the linear program. Thus, since both of these numbers are in the order of $M\cdot L$, the complexity of Algorithm RMP is polynomial in $O(M\cdot L)$ i.e., Algorithm RMP is a *pseudo-polynomial algorithm* [11]. Thus, whenever the value of L is polynomial in the size of the network, Algorithm RMP is a *polynomial optimal* algorithm for Problem RMP. One such case is when the *hop count* metric is considered (i.e., $l_e\equiv 1$), since then $L\leq N-1$.

3.2 ε-Optimal Approximation Scheme for Problem RMP

In the previous subsection we established an optimal polynomial solution to Problem RMP for the case where the length restrictions are sufficiently small. In this subsection we turn to consider the solution to Problem RMP for *arbitrary* length restrictions. As Theorem 1 establishes that Problem RMP is NP-hard for this general case, we focus on the design of an efficient algorithm that approximates the optimal solution.

Our main result in this setting is the establishment of an ε–optimal approximation scheme, which is termed the *RMP Approximation Scheme*. This scheme is based on Algorithm RMP, specified in the previous subsection, which was shown to have a complexity that is polynomial in $M \cdot L$. Given an instance of Problem RMP and an approximation parameter ε, we reduce the complexity of Algorithm RMP by first scaling down the length restriction L by the factor $\Delta \triangleq \frac{L \cdot \varepsilon}{N}$ and then rounding it into an integer value. Obviously, as a result, we must also scale down the length of each link. However, in order to ensure that the optimal network congestion factor does not increase, we relax the constraints of the new instance with respect to the constraints of the original instance. Specifically, after we scale down the length restriction and the length of each link by the factor Δ, we round *up* the length restriction and round *down* the length of each link. Then, we invoke Algorithm RMP over the new instance, in order to construct a path flow that minimizes congestion while satisfying the relaxed length restrictions. Finally, we convert each non-simple path in the output of Algorithm RMP into a simple path by eliminating loops; this is essential, since the total error in the evaluation of the length of each path depends on the hop count. In Theorem 2, we establish that the resulting path flow violates the length restriction by a factor of at most $(1+\varepsilon)$ and has a network congestion factor that is not larger than the optimal network congestion factor. The proof can be found in [4].

Theorem 2: Given an instance $<G,\{s,t\},\{c_e\},\{l_e\},\gamma,L>$ of problem RMP and an approximation parameter ε, the *RMP Approximation Scheme* has a complexity that is polynomial in $1/\varepsilon$ and the size of the network; moreover, the output of the scheme is a path flow f that satisfies the following:

a. $\sum_{p\in P^{(s,t)}} f_p = \gamma$ i.e., the flow demand requirement is satisfied.

b. If α^* is the network congestion factor of the optimal solution, then, for each $e\in E$, it holds that $\sum_{p\in P^{(s,t)}} \Delta_e(p)\cdot f_p \leq \alpha^* \cdot c_e$, i.e., the network congestion factor is at most α^*.

c. For each path $p\in P^{(s,t)}$, if $f_p>0$ then p is *simple* and $L(p)\leq(1+\varepsilon)\cdot L$, i.e., the length restriction is violated by a factor of at most $(1+\varepsilon)$.

3.3 Further Results

In the following, we outline two important extensions to Problem RMP.

Multi-commodity Extensions: In [4], we consider a multi-commodity extension of Problem RMP, i.e., a problem with several source-destination pairs. Following basically the same lines as in Subsections 3.1 and 3.2, we present a pseudo-polynomial solution for this problem and establish an ε-optimal approximation.

End-to-End Reliability Constraints: In [4], we also consider the increased vulnerability to failures when multipath routing is employed in order to balance the network load. Indeed, when traffic is split among multiple paths, a failure in each routing path may result in the failure of the entire transmission. In [4] we formulate the problem and show that it is computationally intractable. However, we show there that the RMP Approximation Scheme can be modified in order to constitute an ε–optimal approximation scheme for the reliability problem.

4 Minimizing Congestion with K Routing Paths

In this section we solve Problem KPR, which minimizes congestion while routing traffic along at most K different paths. In [4], we show that Problem KPR admits a (straightforward) polynomial solution when the restriction on the number of paths is larger than the number of links M (i.e., $K \geq M$). However, we show in [4] that, in the more interesting case where $K<M$, the problem is NP-hard. Accordingly, in this section we present a 2-approximation scheme for $K<M$.

Our approximation scheme is based on solving an auxiliary problem that minimizes congestion while restricting the flow along each path to be *integral* in γ/K. In order to formulate the corresponding problem, consider first the following definition.

Definition 7: Given are a network $G(V,E)$, a capacity $c_e>0$ for each link $e \in E$, a demand γ and an integer K. A path flow $f{:}P \rightarrow \mathbb{R}^+ \cup \{0\}$ is said to be γ/K-*integral*, if for each path $p \in P^{(s,t)}$, it holds that f_p is a multiple of γ/K.

Problem Integral Routing: Given are a network $G(V,E)$, two nodes $s,t \in V$, a capacity $c_e>0$ for each link $e \in E$, a demand $\gamma>0$ and an integer K. Find a γ/K-integral path flow that minimizes the network congestion factor, such that the demand γ is satisfied.

4.1 Solving the Integral Routing Problem

The following observation shall be used in order to construct a polynomial solution to the Integral Routing Problem. The proof can be found in [4].

Lemma 1: Given an instance $<G,\{s,t\},\{c_e\},\gamma,K>$ of the Integral Routing Problem, the optimal network congestion factor is included in the set $\bar{\alpha} \triangleq \left\{ \frac{i \cdot \gamma}{K \cdot c_e} \middle| e \in E, i \in [0,K] \cap \mathbb{Z} \right\}$[1].

We now introduce *Procedure Test*, which is given an instance $<G,\{s,t\},\{c_e\},\gamma,K>$ of the Integral Routing Problem and a restriction α on the network congestion factor. Procedure Test performs three sequential steps. Initially, it multiplies all link capacities by a factor of α in order to impose the restriction on the network congestion factor; indeed, multiplying all capacities by α assures that the flow f_e along each link

[1] Observe that the size of $\bar{\alpha}$ is polynomial in the network size, namely: $\left|\bar{\alpha}\right| \leq M \cdot (K+1) = O\left(M^2\right)$.

$e \in E$ is at most $\alpha \cdot c_e$; therefore, for each $e \in E$, the link congestion factor f_e/c_e, and, in particular, the network congestion factor $\max_{e \in E}\{f_e/c_e\}$, are at most α. Next, the procedure rounds down the capacity of each link to the nearest multiple of γ/K; since the flow over each path in every solution to the Integral Routing Problem is γ/K-integral, such a rounding has no effect on the capability to transfer the flow demand γ. Finally, the procedure applies any standard maximum flow algorithm that returns an integral link flow when all capacities are integral. Since all capacities are γ/K-integral, the maximum flow algorithm determines a γ/K-integral link flow that transfers the maximum amount of flow without violating the restriction α on the network congestion factor. If this link flow succeeds in transferring at least γ flow units from s to t, then the procedure returns it. Otherwise, the procedure fails.

Theorem 3: Given is an instance $<G,\{s,t\},\{c_e\},\gamma,K>$ of the Integral Routing Problem. Denote by α^* the corresponding optimal network congestion factor. Then, Procedure Test succeeds for the input $<G,\{s,t\},\{c_e\},\gamma,K,\alpha>$ *iff* $\alpha \geq \alpha^*$.
The proof appears in [4].

Theorem 3 has two important implications that enable to construct an efficient solution to the Integral Routing Problem. First, the theorem establishes that the smallest α for which Procedure Test succeeds with the input $<G,\{s,t\},\{c_e\},\gamma,K,\alpha>$ is equal to α^*. Therefore, if S is a finite set that includes the optimal network congestion factor α^* and α is the smallest network congestion factor in S such that Procedure Test succeeds for the input $<G,\{s,t\},\{c_e\},\gamma,K,\alpha>$, then $\alpha=\alpha^*$. This fact, together with the fact that the set $\overline{\alpha}$ includes α^* (as per Lemma 1), imply that, for every instance $<G,\{s,t\},\{c_e\},\gamma,K>$ of Problem Integral Routing, the optimal network congestion factor α^* is the smallest $\alpha \in \overline{\alpha}$ such that Procedure Test succeeds for the input $<G,\{s,t\},\{c_e\},\gamma,K,\alpha>$. Moreover, since in case of a success Procedure Test returns the corresponding link flow, finding the smallest $\alpha \in \overline{\alpha}$ such that Procedure Test succeeds identifies a link flow with a network congestion factor of at most α^*.

The second implication of Theorem 3 enables to employ a *binary search* when we seek the smallest $\alpha \in \overline{\alpha}$ such that Procedure Test succeeds. Indeed, it follows from Theorem 3 that, when Procedure Test succeeds for $\alpha_1 \in \overline{\alpha}$, it succeeds for all $\alpha \in \overline{\alpha}$, $\alpha \geq \alpha_1$; and when it fails for $\alpha_2 \in \overline{\alpha}$, it fails for all $\alpha \in \overline{\alpha}$, $\alpha \leq \alpha_2$; thus, if Procedure Test succeeds for $\alpha_1 \in \overline{\alpha}$ (alternatively, fails for $\alpha_2 \in \overline{\alpha}$) it is possible to eliminate from further consideration all the elements of $\overline{\alpha}$ that are larger than α_1 (correspondingly, smaller than α_2).

Remark 3: Note that performing a binary search over $\overline{\alpha}$ requires *sorting* all the elements of $\overline{\alpha}$, which consumes $O\left(\left|\overline{\alpha}\right| \cdot \log\left|\overline{\alpha}\right|\right) \leq O\left(M^2 \cdot \log N\right)$ operations [10].

Thus, we conclude that the employment of a binary search so as to find the smallest $\alpha \in \overline{\alpha}$ for which Procedure Test succeeds, establishes a link flow that has the minimal network congestion factor. The optimal solution is then achieved by decomposing the resulting link flow into a path flow via the flow decomposition algorithm [1]. Due to space limits, the formal description of this algorithm, termed *Algorithm Integral Routing*, is omitted and can be found in [4]. Our discussion is summarized by the following theorem, which establishes that Algorithm Integral Routing solves Problem Integral Routing. Its proof appears in [4].

Theorem 4: Given is an instance $<G,\{s,t\},\{c_e\},\gamma,K>$ of Problem Integral Routing. If Algorithm Integral Routing returns Fail, then there is no feasible solution for the given instance; otherwise, the algorithm returns a γ/K-integral path flow that transfers at least γ flow units from s to t along simple paths, such that the network congestion factor is minimized.

Remark 4: It is easy to show [4] that the computational complexity of Algorithm Integral Routing is $O(M \cdot logN \cdot (M + N \cdot logN))$.

4.2 A 2-Approximation Scheme for Problem KPR

Finally, we are ready to establish a solution for Problem KPR. To that end, we show that the solution of the Integral Routing Problem can be used in order to establish a *constant approximation scheme* for Problem KPR. The approximation scheme is based on the following key observation, which links between the optimal solution of Problem Integral Routing and the optimal solution of Problem KPR.

Theorem 5: Given are a network $G(V,E)$ and a demand of γ flow units that has to be routed from s to t. If f_1 is a γ/K-integral path flow that has the minimum network congestion factor and f_2 is a path flow that minimizes its network congestion factor while routing along at most K paths, then the network congestion factor of f_1 is at most twice the network congestion factor of f_2.

Proof: Suppose that f_1 and f_2 satisfy the assumptions of the Theorem. Let α_1 and α_2 denote the network congestion factor of path flows f_1 and f_2, respectively. We have to show that $\alpha_1 \leq 2 \cdot \alpha_2$.

Out of the path flow f_2, we construct a γ/K-integral path flow that ships at least γ flow units from s to t and has a network congestion factor of at most $2 \cdot \alpha_2$. Clearly, such a construction implies that the network congestion factor of every *optimal* γ/K-integral path flow that ships γ flow units from s to t is at most $2 \cdot \alpha_2$; in particular, since f_1 is one such optimal γ/K-integral path flow, such a construction establishes that $\alpha_1 \leq 2 \cdot \alpha_2$.

With this goal in mind, define the following construction. First, double the flow along each routing path that f_2 employs; obviously, the resulting path flow transfers $2 \cdot \gamma$ flow units from s to t along at most K routing paths while yielding a network congestion factor of $2 \cdot \alpha_2$. Then, *round down* the (doubled) flow along each routing path to the nearest multiple of γ/K; in this process, the flow along each path is reduced by at most γ/K flow units. Hence, since there are no more than K routing paths, the total flow from s to t is reduced by at most γ units; therefore, since before the rounding operation exactly $2\cdot\gamma$ flow units were shipped from s to t, it follows that after rounding is performed, the resulting path flow transfers at least γ flow units from s to t.

Thus, we have identified a γ/K-integral path flow that transfers at least γ flow units from s to t. In addition, since prior to the rounding operation the network congestion factor is $2\cdot\alpha_2$ and the rounding can only reduce flow, the network congestion factor of the constructed path flow is at most $2\cdot\alpha_2$. ■

Note that, given a network $G(V,E)$ and a demand γ that needs to be routed over at most K paths, *every* γ/K-integral path flow satisfies the requirement to ship the demand γ on at most K different paths. On the other hand, it has been established in Theorem 5 that the network congestion factor obtained by an optimal γ/K-integral path flow is at most twice the network congestion factor of an optimal flow that admits at most K routing paths. Thus, computing a γ/K-integral path flow that has the minimum network congestion factor satisfies the restriction on the number of routing paths and obtains a network congestion factor that is at most twice larger than the optimum. We summarize the above discussion in the following corollary, which yields an approximation scheme for Problem KPR.

Corollary 1: Given are a network $G(V,E)$, a demand γ and a restriction on the number of routing paths K. The employment of Algorithm Integral Routing for the establishment of a γ/K-integral path flow that minimizes the network congestion factor provides a 2-approximation scheme for Problem KPR with a complexity of $O(M\cdot logN\cdot(M+N\cdot logN))$.

4.3 Further Results

In [4], we generalize the result of this section into a *bicriteria* result. Specifically, for any given $r\geq 1$, we establish a $(1+1/r)$-approximation scheme that violates the constraint on the number of paths by a factor of at most r. Note that, for $r = 1$, the corresponding scheme obtains the same performance guarantees as in Subsection 4.2 above. In addition, in [4] we consider the *dual* problem, which restricts the network congestion factor while minimizing the number of routing paths, and present a corresponding approximation scheme.

5 Simulation Results

In this section, we present a comparison between an optimal solution to multipath routing and that provided by a heuristic scheme such as the (popular) Equal Cost MultiPath (ECMP) routing scheme.

We generated 10,000 random topologies, following the lines of [23][1]. For each topology, we conducted the following measurements: (a) we measured the network congestion factor produced by invoking ECMP; (b) we measured the network

[1] Due to space limits, we omit the details of this construction, which can be found in [4].

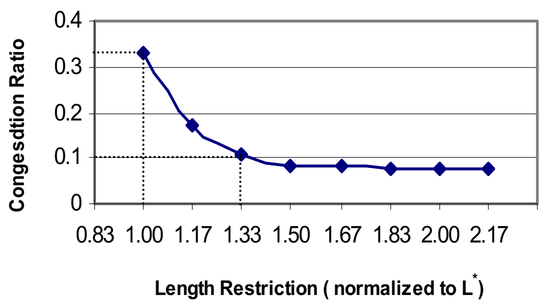

Fig. 2. The ratio between the network congestion produced by an optimal multipath routing assignment (for several length restrictions) and the network congestion produced by ECMP

congestion factor produced by an optimal assignment of traffic to shortest paths and to paths with a length that is equal to $1.17{\cdot}L^*$, $1.33{\cdot}L^*$, $1.5{\cdot}L^*$, $1.67{\cdot}L^*$, $1.83{\cdot}L^*$,$2{\cdot}L^*$ and $2.17{\cdot}L^*$, where L^* is the length of a shortest path. Our results are summarized in Fig. 2. Note that if the ECMP scheme had an optimal traffic distribution mechanism, the network congestion factor could be reduced by a *factor of 3*. Moreover by relaxing the requirement to route along shortest paths by 33%, the network congestion factor is *10 times smaller* than with the standard ECMP. Thus, by employing Algorithm RMP or its e-optimal approximation with $L \approx 1.33 \cdot L^*$, congestion can be reduced by a factor of *10* with respect to that produced by ECMP.

6 Conclusion

Previous multipath routing schemes for congestion avoidance focused on heuristic methods. Yet, our simulations indicate that optimal congestion reduction schemes are *significantly* more efficient. Accordingly, we investigated multipath routing as an optimization problem of minimizing network congestion, and considered two fundamental problems. Although both have been shown to be computationally intractable, they have been found to admit efficient approximation schemes. Indeed, for each problem, we have designed a *polynomial* time algorithm that approximates the optimal solution by a (small) *constant* approximation factor.

While this study has laid the algorithmic foundations for two fundamental multipath routing problems, there are still many challenges to overcome. One major challenge is to establish an efficient unifying scheme that combines the two problems. Furthermore, as in practice there may be a need for simpler solutions, another research challenge is the development of approximations with lower computational complexity. Finally, as discussed in [4], multipath routing offers reach ground for research also in other contexts, such as survivability, recovery, network security and energy efficiency. We are currently working on these issues and have obtained several results regarding survivability [5].

References

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows: Theory, Algorithm, and Applications", Prentice Hall, 1993.
[2] D. Awduche, J. Malcolm, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS" , Internet Draft, April, 1998.
[3] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for Traffic Engineering Over MPLS", IETF RFC 2702, September 1999.
[4] R. Banner and A. Orda, "Multipath Routing Algorithms for Congestion Minimization", CCIT Report No. 429, Department of Electrical Engineering, Technion, Haifa, Isreal, 2004. Available from: http://www.ee.technion.ac.il/people/ron/Congestion.pdf
[5] R. Banner and A. Orda, "The Power of Tuning: a Novel Approach for the Efficient Design of Survivable Networks", In Proc. IEEE ICNP 2004 (**Best Paper Award**).
[6] D. Bertsekas and R. Gallager, "Data networks", Prentice-Hall, 1992.
[7] Allan Borodin and Ran El-Yaniv, "Online Computation and Competative Analysis", Cambridge University Press, 1998.
[8] I. Cidon, R. Rom and Y. Shavitt, "Analysis of Multi-Path Routing", IEEE Transactions on Networking, v. 7, No. 6, pp. 885–896, 1999.
[9] S. Chen and K. Nahrstedt, "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions", IEEE Network, v. 12, No. 6, pp. 64-79, December 1998.
[10] T. Cormen, C. Leiserson, and R. Rivest, Introduction to Algorithms. Cambridge, MA: The MIT Press, 2001.
[11] M. R. Garey and D. S. Johnson, "Computers and Intractability", W.H. Freeman and Co., 1979.
[12] A. V. Goldberg and R. E. Tarjan, "A New Approach to The Maximum Flow Problem", Journal of ACM, v.35, No. 4, pp. 921-940, 1988.
[13] S. Iyer, S. Bhattacharyya, N. Taft, N. McKeoen, C. Diot, "A measurement Based Study of Load Balancing in an IP Backbone", Sprint ATL Technical Report, TR02-ATL-051027, May 2002.
[14] Y. Jia, Ioanis Nikolaidis and P. Gburzynski, "Multiple Path QoS Routing", Proceedings of ICC'01 Finland, pp. 2583-2587, June 2001.
[15] N. Karmarkar, "A New Polynomial-Time Algorithm For Linear Programming", Combinatorica, vol. 4, pp. 373-395, 1984.
[16] J.Moy, "OSPF Version 2", IETF RFC 2328, April 1998.
[17] S. Nelakuditi and Zhi-Li Zhang, "On Selection of Paths for Multipath Routing", In Proc. IWQoS, Karlsruhe, Germany, 2001.
[18] V. Paxson, "End-to-End Routing Behavior in the Internet", in proc. ACM SIGCOMM, 1996.
[19] E. Rosen, A. Viswanathan, and R. Callon. "Multiprotocol Label Switching Architecture". IETF RFC 3031, 2001.
[20] D. Thaler and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", IETF RFC 2991, 2000.
[21] C. Villamizar, OSPF Optimised Multipath (OSPF-OMP), Internet Draft, 1998.
[22] Y. Wang and Z. Wang, "Explicit Routing Algorithms For Internet Traffic Engineering", In Proc. ICCN'99, Boston, October 1999.
[23] B. M. Waxman, *Routing of Multipoint Connections*, IEEE Journal on Selected Areas in Communications, 6:1617-1622,1988.
[24] A. E. I. Widjaja, "Mate: MPLS Adaptive Traffic Engineering", Internet Draft, August, 1998.