How Bad is Single-Path Routing

Meng Wang
School of ECE
Cornell University, USA

Chee Wei Tan
EAS Division
Caltech, USA

Ao Tang
School of ECE
Cornell University, USA

Steven H. Low
EAS Division
Caltech, USA

***Abstract*—This paper investigates the network performance loss of using only single-path routing when multiple paths are available. The performance metric is the aggregate utility achieved by the joint optimization of congestion control and routing. As computing the exact loss for a general network topology is NP-hard, we develop analytical bounds on this "cost of not splitting". Our bound is independent of the number of source-destination pairs when the latter one is larger than the number of links in a network. We also propose a vertex projection method and combine it with branch-and-bound to provide progressively tighter bounds on the performance loss. Numerical examples are used to show the effectiveness of our approximation technique.**

I. Introduction

Routing is one of the key network protocols in communication networks such as the Internet. It selects the paths for traffic to flow from all the sources to their destinations. Even though there are proposals to allow flexible multipath routing in the Internet [1], the current routing protocol within an AS, e.g., the Open Shortest Path First (OSPF) protocol, primarily uses single-path routing where one user (source-desperation pair) uses only one path from the source to the destination, with the exception that traffic may split evenly among equal-cost paths.

Recently, there has been an interest to consider joint routing and congestion control, where routing and transmission rates are jointly optimized. To be more specific, assuming every user has a utility function that depends on its total transmission rate, one seeks to maximize the total utilities of all users subject to link capacity constraints.[1] This problem is analytically tractable if users can use all available paths because allowing source-based multipath routing makes the problem convex and admits an elegant optimality characterization. When each user optimizes its transmission rate over only one out of all available paths, i.e., using source-based single-path routing, this combinatorial problem is however nonconvex and known to be NP-hard [3].

This paper focuses on a key question: As compared with multipath routing, how is performance affected in terms of the aggregate utility by restricting to single-path routing? In other words, how bad is single-path routing? Or what is the "cost of not splitting"? It can guide the decision on whether to support multipath routing with flexible splitting (which is arguably expensive to support), since single-path routing has a smaller overhead in implementation. We first show that allowing users to send traffic on all the paths will not benefit the aggregate utility too much as the optimal multipath utility can be achieved even when the total number of paths in use is at most the sum of the number of links plus the number of users. We also provide analytical bounds as well as algorithms to estimate this performance gap.

Our problem belongs to the general multicommodity flow category [4]. Even though our formulation is closely related to that in [3], there are however other formulations in the multicommodity flow category. In particular, there has been recent study on "unsplittable flows" [5]–[8]. We remark that even though formulations may differ, the intrinsic difficulty is the same when they are considered as a class of combinatorial optimization problems. Therefore, we expect our work will also be useful to other related multicommodity problems.

The paper is organized as follows. After introducing the model and notation in Section II, we derive a general upper bound on the performance loss in Section III. The analysis suggests a vertex projection to find a near optimal single-path solution which is discussed in Section IV. In Section V, we further refine our estimation by combining vertex projection with a branch-and-bound technique. Numerical examples are provided in Section VI to demonstrate the effectiveness of our estimation. We conclude the paper in Section VII.

II. Model and Notation

A network consists of a set of L uni-directional links with finite capacities $c = (c_l, l = 1, ..., L)$, and supports a set of N source-destination pairs or users, indexed by i. There are K^i acyclic paths for user i and is represented by a $L \times K^i$ matrix R^i, where $R^i_{lk} = 1$ if path k of user i uses link l, and $R^i_{lk} = 0$ otherwise. The overall routing matrix is given by

$$R = [\ R^1 \quad R^2 \quad ... \quad R^N\ ].$$

For example, as shown in Fig.1, a seven-link network supports two users, each of which has two possible paths to choose from, and the corresponding routing matrices are

$$R^1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T,$$

$$R^2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}^T.$$

For every i, define a $K^i \times 1$ vector x^i with the rate x^i_k of path k of user i as the kth entry of x^i. The total rate of user i is $\|x^i\|_1 = \sum_{k=1}^{K^i} x^i_k$. Let a $\sum_i K^i \times 1$ vector x represent the complete bandwidth allocation

$$x = [\ (x^1)^T \quad (x^2)^T \quad ... \quad (x^N)^T\ ]^T.$$

[1]Other formulations are possible, e.g., minimizing the total link cost while satisfying the traffic demand between the sources and destinations [2].

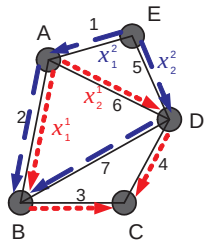

Fig. 1. A network of seven links supporting two users. User one: A to C (red dotted line), and user two: E to B (blue dashed line).

Each user i has a utility function U^i as a function of its total transmission rate $\|x^i\|_1$. We assume U^i to be strictly increasing and concave, which is the case for most TCP algorithms [9]. Let $U = (U^i, i = 1, ..., N)$. We call (c, R, U) a *network*.

The joint congestion control and multipath routing problem is to maximize the aggregate network utility by allocating rates for all users over all possible paths subject to link capacity constraints. We assume fairness in the sense that every user receives a positive rate, i.e., $\|x^i\|_1 > 0$ for any user i. It can be formulated as a convex optimization problem:

$$\begin{aligned} \max_{x \geq 0} \quad & \textstyle\sum_i U^i(\|x^i\|_1) \\ \text{s.t.} \quad & Rx \leq c. \end{aligned} \tag{1}$$

Compared with problem (1), each user in the joint flow control and single-path routing problem has additional constraints, namely each user only uses one path out of its finite set of possible paths. Let $\|x^i\|_0$ be the number of non-zero entries of x^i.[2] Then the single-path problem can be formulated as a non-convex optimization problem:

$$\begin{aligned} \max_{x \geq 0} \quad & \textstyle\sum_i U^i(\|x^i\|_1) \\ \text{s.t.} \quad & Rx \leq c, \\ & \|x^i\|_0 = 1 \quad \forall\, i. \end{aligned} \tag{2}$$

Let opt_M and opt_S denote the optimal value of (1) and (2) respectively. Then, $opt_M - opt_S$ can be interpreted as a measure of performance loss due to the additional single-path routing constraints, or the "cost of not splitting". Moreover, it was shown in [3] that the dual problem of (2) has the same optimal value as (1). Therefore, the duality gap of (2) is precisely the performance loss of utilizing only one out of a finite choice of paths by each user.

Throughout the paper, we assume $\sum_i K^i > N + L$, i.e. the cumulated number of paths that can be used by users is greater than the sum of the number of users and the number of links. This is the case when the network is large and therefore each user has many available paths. The key notation used is summarized in Table I.

[2]Our notation for the cardinality of x^i is the same as that used in the compressed sensing literature, and is commonly known as the L_0 norm .

TABLE I
SUMMARY OF KEY NOTATION

Notation	Meaning
$\tilde{x}$	A feasible multipath allocation
x^*	An optimal multipath allocation
$\hat{x}$	A vertex of the optimal solution set of the multipath problem
x'	A feasible single-path allocation projected from $\hat{x}$
$\bar{x}$	Optimizer of the fixed routing congestion control problem using the path configuration of x'

III. General Bound of Duality Gap

In this section, we obtain a general upper bound of the duality gap, and then show that the average cost of not splitting goes to zero asymptotically as the number of users goes to infinity. First, the following theorem indicates that for a network with N users, the optimal network utility can be achieved by using at most $L + N$ paths although the total available paths can be potentially exponential.

Theorem 1. *Given an L-link network supporting N users, for any multipath allocation $\tilde{x}$, there exists a multipath allocation x using at most $L + N$ paths such that they both achieve the same aggregate utility.*

Proof: Given a multipath allocation $\tilde{x}$, consider the following nonempty and bounded polyhedron:

$$P = \{x \in \mathcal{R}^{\sum_i K^i} \mid x \geq 0,\ \mathbf{1}^T x^i = \mathbf{1}^T \tilde{x}^i\ \forall\, i,\ \text{and}\ Rx \leq c\}.$$

Clearly, P contains at least one vertex [10], denoted by x. Since x is a vertex, there are $\sum_i K^i$ linearly independent constraints that are active at x. Note we call a constraint active if it holds with equality. Since we already have N active constraints from $\mathbf{1}^T x^i = \mathbf{1}^T \tilde{x}^i \quad \forall\ i$, and at most L active constraints from $Rx \leq c$, then there are at least $\sum_i K^i - N - L$ constraints from $x \geq 0$ are active. Therefore, at least $\sum_i K^i - N - L$ entries of x are zero, indicating that x contains at most $N + L$ positive entries. ■

Remark 1. We can obtain a similar result by applying the Shapley-Folkman theorem in [11] to (2) such that the optimal multipath utility can be achieved by a multipath allocation using at most $N + L + 1$ paths. Theorem 1 is thus a slightly stronger result than a direct application of [11].

Remark 2. Similar arguments hold for a more general class of delay-sensitive utility functions proposed in [12]. The objective function is $\sum_i U^i(\|x^i\|_1) - d^T Rx$ where $d^{L\times 1}$ stores the delay of each link. An additional constraint $d^T Rx = d^T R\tilde{x}$ exists in the set P, making the upper bound on the number of paths $N + L + 1$. All later results hold accordingly.

Theorem 1 implies that to achieve optimal network utility, at most L users (assuming $L < N$) need to use multiple paths, while others only need single-path routing. We next use this property to upper bound the duality gap.

Theorem 2. *Given an L-link network supporting N users, the duality gap of (2) is upper bounded by*

$$\min(L, N) \max_i \rho^i, \tag{3}$$

where

$$\rho^i = \max\{U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty) \mid x^i \geq 0, R^i x^i \leq c\}. \quad (4)$$

Proof: Let x^* be an optimal solution to (1). Following Theorem 1, there exists an optimal allocation $\hat{x}$ and a set S ($|S| \leq \min(L, N)$) of indices such that $\mathbf{1}^T \hat{x}^i = \mathbf{1}^T x^{*i} \;\; \forall\, i$, $R\hat{x} \leq c$, $\|\hat{x}^i\|_0 > 1 \;\forall\, i \in S$, and $\|\hat{x}^i\|_0 = 1 \;\forall\, i \notin S$.

Next, we project $\hat{x}$ to a feasible single-path solution x' by picking the maximum-rate path for each user. More formally, for any $i \in S$, let $d(i)$ be the index of the largest entry of $\hat{x}^i$, i.e.,

$$\hat{x}^i_{d(i)} = \|\hat{x}^i\|_\infty \;\forall\, i \in S.$$

Then, we define a $K^i \times 1$ vector x'^i such that

$$x'^i_{d(i)} = \hat{x}^i_{d(i)}, \text{ and } x'^i_k = 0, \;\forall\, k \neq d(i).$$

For any $i \notin S$, let $x'^i = \hat{x}^i$. Then, one can check that $\|x'^i\|_0 = 1, \;\forall\, i$, and $Rx' \leq R\hat{x} \leq c$, thus x' is a feasible solution to (2). Since $\hat{x}$ is an optimal solution to (1), we have

$$\begin{aligned} opt_M - opt_S \leq & \quad \textstyle\sum_i U^i(\|\hat{x}^i\|_1) - \sum_i U^i(\|x'^i\|_1) \\ = & \quad \textstyle\sum_{i\in S} U^i(\|\hat{x}^i\|_1) + \sum_{i\notin S} U^i(\|\hat{x}^i\|_1) \\ & - \textstyle\sum_{i\in S} U^i(\|x'^i\|_1) - \sum_{i\notin S} U^i(\|x'^i\|_1) \\ = & \quad \textstyle\sum_{i\in S} \left(U^i(\|\hat{x}^i\|_1) - U^i(\|\hat{x}^i\|_\infty)\right) \\ \leq & \quad \textstyle\sum_{i\in S} \rho^i \leq \min(L, N) \max_i \rho^i, \end{aligned}$$

where ρ^i is given by (4). ■

Note that ρ_i is a measure for the degree of nonconvexity of the ith utility function, and was first proposed in [13]. Moreover, it can be upper bounded as follows.

Theorem 3.

$$\rho_i \leq \max_{y\in[0,M^i]} (U^i(y) - U^i(y/K^i)), \quad (5)$$

where M^i is the maximum total transmission rate of user i that can be supported by a network (c, R, U), i.e.,

$$M^i = \max\{\|x^i\|_1 \;|x^i \geq 0, R^i x^i \leq c\}.$$

Proof: For any x^i such that $x^i \geq 0, R^i x^i \leq c$, we have $\|x^i\|_\infty \geq \|x^i\|_1/K^i$. Since U^i is strictly increasing, then

$$U(\|x^i\|_1) - U(\|x^i\|_\infty) \leq U(\|x^i\|_1) - U(\|x^i\|_1/K^i).$$

The inequality still holds if we maximize over all feasible x^i, thus

$$\begin{aligned} \rho^i \;\; \leq \;\; & \max\{U(\|x^i\|_1) - U(\|x^i\|_1/K^i) \;|x^i \geq 0, R^i x^i \leq c\} \\ = \;\; & \max_{y\in[0,M^i]} (U(y) - U(y/K^i)). \end{aligned}$$

■

Now, (5) is easy to calculate for a given network. Also, since M^i is always finite, it implies that ρ^i is finite for any user i.

Corollary 1. *If $U^i(\cdot) = \alpha^i \log(\cdot)$ where $\alpha^i > 0$ is some constant, then*

$$\rho_i = \alpha^i \log(K^i). \quad (6)$$

Proof: By Theorem 3, we have $\rho_i \leq \alpha^i \log(K^i)$. Also, there always exists some $\epsilon > 0$ such that $x^i = [\epsilon, \epsilon, ..., \epsilon]^T$ satisfies $R^i x^i \leq c$, thus, from (4), $\rho_i \geq \alpha^i \log(K^i)$. This proves (6). ■

Both the TCP Vegas protocol in [14] and the FAST TCP protocol in [15] implicitly solve (2) using the logarithm utility function. Thus, (6) gives a simple formula of ρ_i, which only depends on the number of paths of user i regardless of the network topology and c.

The significance of Theorem 2 is that when $N \geq L$, the upper bound of the duality gap depends on L and $\max_i \rho^i$, but not on N. We next use this property to show that the average "cost of not splitting" becomes zero asymptotically.

From Theorem 2, when $N \geq L$, we have $opt_M - opt_S \leq L \max_i \rho^i$, which yields

$$\frac{opt_M - opt_S}{N} \leq \frac{L}{N} \max_i \rho^i, \quad (7)$$

where $(opt_M - opt_S)/N$ can be interpreted as the average cost of not splitting. Since $\max_i \rho^i$ is bounded, the righthand side term of (7) goes to zero as $N \to \infty$. This result implies that, although the cost of not splitting is positive if we consider (2), the average cost of not splitting is close to zero if N is sufficiently large. In other words, by replacing the objective function in (2) with the average network utility, we get the following problem:

$$\begin{aligned} \max_{x\geq 0} \quad & \tfrac{1}{N} \textstyle\sum_i U^i(\|x^i\|_1) \\ \text{s.t.} \quad & Rx \leq c, \\ & \|x^i\|_0 = 1 \quad \forall\, i. \end{aligned} \quad (8)$$

Even though (8) is still nonconvex and has a positive duality gap, due to (7), its duality gap approaches zero asymptotically as N goes to infinity.

IV. Estimation of Duality Gap via Vertex Projection

In general, (3) may not be a tight upper bound of the duaity gap. However, for a given network, if we can find an optimal solution $\hat{x}$ of (1) and its corresponding index set S as defined in the proof of Theorem 2, a better upper bound to the duality gap than that given by (3) can be obtained as follows. From the proof of Theorem 2, we can find a feasible solution x' to (2) by keeping the largest path rate of each $\hat{x}^i$ and setting other path rates to zero. In this case, an improved upper bound to the duality gap is given by

$$opt_M - opt_S \leq \sum_{i\in S} \left(U^i(\|\hat{x}^i\|_1) - U^i(\|\hat{x}^i\|_\infty)\right).$$

However, x' may not achieve the maximum utility over this particular resultant single-path configuration. By solving a fixed routing congestion control problem using the path configuration of x', we obtain a single-path allocation $\bar{x}$ such that $\sum_i U^i(\|\bar{x}^i\|_1) \geq \sum_i U^i(\|x'^i\|_1)$. Thus, a better upper bound of the duality gap is

$$opt_M - opt_S \leq \sum_i U^i(\|\hat{x}^i\|_1) - \sum_i U^i(\|\bar{x}^i\|_1). \quad (9)$$

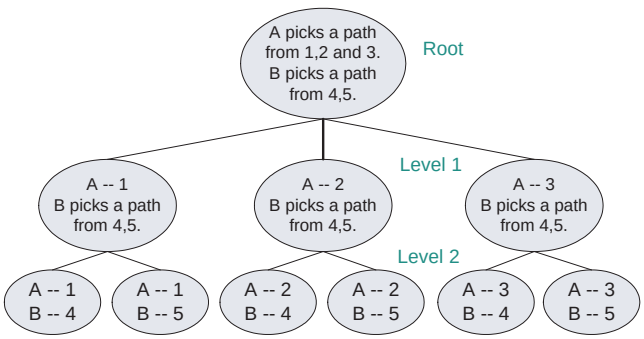

Fig. 2. The feasible path sets of two users (user A has three paths 1,2 and 3, user B has two paths 4 and 5)

To find such an $\hat{x}$ and S, we first solve (1) to obtain a multipath optimizer x^* (in polynomial time as (1) is convex). We then check whether x^* satisfies $\|x^{*i}\|_0 = 1$ for all $i = 1, ..., N$, or not. If so, (2) has the same value as the (1) and the duality gap is zero. This also means that the set S is empty in the first place. Otherwise, the problem is equivalent to finding a basic feasible solution $\hat{x}$ to a linear program over the polyhedron Q:

$$Q = \{x \in \mathcal{R}^{\sum_i K^i} \mid x \geq 0,\ \mathbf{1}^T x^i = \mathbf{1}^T x^{*i}\ \forall\, i,\ \text{and}\ Rx \leq c\}.$$

Clearly, Q is a convex polyhedron that is bounded, finite and pointed. Moreover, any point in Q is an optimal solution to the multi-path problem. Since U^i is strictly concave, one can check that Q also contains all the optimal solutions. Thus Q is the solution set of (1).

To find a vertex of Q, we can formulate an auxiliary maximization problem over Q with a different choice for the objective function. For example, let $k(i)$ be the path index of user i such that $x^{*i}_{k(i)} = \|x^{*i}\|_\infty$. Then, one possible objective function is $\sum_i x^i_{k(i)}$. We can also find all the vertices of the polyhedron Q, so as to find the best upper bound. This is a vertex enumeration problem and currently has no polynomial time algorithm. However, a practical pivot-based algorithm has been proposed in [16] to find v vertices of a polyhedron in $\mathbf{R}^d$ defined by a non-degenerate system of n inequalities in $O(ndv)$ time and $O(nd)$ space.

In summary, we first find a vertex of Q and then project it to a single-path configuration by picking the maximum-rate path for each user, then maximize network utility over this particular single-path configuration. We refer to this method as *vertex projection*. It gives a lower bound of opt_S and a single-path configuration that is feasible for (2). As the set S of users that use multiple paths is small for a vertex solution ($|S| \leq \min(L, N)$), we expect the cost of not splitting to be small after the multipath configuration is projected to a single-path configuration. Therefore the vertex projection method can give a relative "good" lower bound of opt_S, and also upper-bound the duality gap by (9).

V. Estimation of Duality Gap Via Branch-And-Bound

The vertex projection in Section IV gives a single-path allocation with its corresponding network aggregate utility serving as a lower bound of (2). In this section, we show how to integrate it with a branch-and-bound algorithm to give a better estimate of opt_S. An N-level tree is introduced to represent a progressively finer partition of the set of paths that each user considers. In particular, the tree has at its root node the original single-path problem (2). The intermediate nodes correspond to problems where some users fix their path choices while every other user can still choose its path from several paths. At each tree node, one user that has not fixed its path choice, say user n, partitions its set of paths producing K^n different subtree nodes. The tree has $\prod_{i=1}^N K^i$ leaf nodes, and each leaf node corresponds to a utility maximization problem over a specific single-path configuration. Figure 2 illustrates the feasible path sets in a two-user case.

We first state how to find opt_S via branch-and-bound. The algorithm starts from the root and branches from the current tree node into several subproblems at each step. For each new-found tree node, we find an upper bound and a lower bound of the maximum utility of (2) over the reduced feasible set for this subproblem. The upper bound is given by the value of its dual problem, which is equivalent to the value of its corresponding multipath problem, and the lower bound can be found by the vertex projection in Section IV. Let m_b be the maximum lower bound that has been found till the current step. If the upper bound for some tree node H is smaller than m_b, then H and all its offspring can be safely pruned from future considering. After pruning, we pick a tree node that has the maximum lower bound among the remaining tree nodes (not including leaf nodes) to branch from for the next step. The algorithm stops when there is no more tree nodes to branch from. It always finds opt_S, which is attained at the leaf node that has the maximum value. However, there is no guarantee that the algorithm will terminate in polynomial time.

To get a polynomial-time approximation algorithm, we propose to do greedy pruning at each step. Specifically, at each level of the tree, we only keep the node that has the maximum lower bound among all the nodes at the same level, and delete all its peers. Then, we branch from this node but also only keep one of its offspring, and so on. Thus, we get a path from the root to a leaf node. Though this leaf node may not solve (2) optimally as we greedily prune all but one node at each step, it gives a good lower bound of opt_S. This algorithm terminates in at most N steps as we fix the path choice for one user at each step.

Another issue is which user to branch from at each level. Let W be the set of users that have already fixed their single-path choices until the current step. Let x be a vertex of the optimal sets of the sub multipath problem in the last step. First, we find the user n that solves

$$n = \arg\max_{i \notin W} U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty),$$

and tentatively branch from n. We solve K^n subproblems

Initial: $W = \emptyset$ /* users that have fixed SP choice */
$I_{sp} = 0$, $I_{end} = 0$
Solve the MP problem, find a vertex x, project it to a SP configuration and find a lower bound m_b of opt_S
while $|W| < N$, $I_{sp} = 0$ and $I_{end} = 0$ **do**
  if $\|x^i\|_1 = \|x^i\|_\infty, \forall\, i$ **then**
    $I_{sp} = 1$ /* x is also a SP solution */
  else
    $B = \{1, 2, ...N\} \backslash W$, $I_{pt} = 0$
    while $I_{pt} = 0$, $I_{end} = 0$ **do**
      $n = \arg\max(U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty)), i \in B$
      /* pick a tentative user to branch */
      for each path k of user n **do**
        Solve a sub MP problem and find a vertex of the set of optimal solutions.
        Project the vertex to a SP configuration and find a lower bound m_k of the sub SP problem.
      end for
      if $\max_k m_k \geq m_b$ **then**
        $I_{pt} = 1$, /* make this branch permanent */
        $m_b = \max_k m_k$,
        $k^* = \arg\max m_k$, fix path k^* for user n
        $W \leftarrow W \cup \{n\}$,
        $x \leftarrow$ the vertex found in the $k^{*\text{th}}$ subproblem
      else
        $B \leftarrow B \backslash \{n\}$
        if $B = \emptyset$ **then**
          $I_{end} = 1$
        end if
      end if
    end while
  end if
end while
return x, m_b

Fig. 3. Algorithm 1: Finding a lower bound m_b of opt_S via greedy branch-and-bound (MP: multipath; SP: single-path)

and find a lower bound m_k of opt_S in the kth subproblem where user n only uses its kth path. Let $m_{k^*} = \max_k m_k$. If $m_{k^*} \geq m_b$, we update m_b with m_{k^*}, make this branch permanent and go to the next step. Otherwise, as m_b cannot be improved in this step by branching from user n, we discard this tentative branch, find the user that has the second largest value of $U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty)$ and make another tentative branch from it. The algorithm stops if all users have made their path choices or if we cannot make any tentative branch permanent at some step. Clearly, the algorithm gives a better estimate of opt_S in each step and thus produces a tighter lower bound of opt_S than the vertex projection in Section IV. Figure 3 summarizes this algorithm.

VI. Numerical Examples

In this section, we describe our numerical evaluation on a random network of L links supporting N users. Let the number of paths for each user be uniformly chosen from the set $\{1, 2, ..., M\}$ for some M. The link capacities are uniformly chosen from the interval [50,100]. A path uses a link with probability $p = 2\log(L)/L$. We use the same utility function $U(x) = \log(1 + x)$ for all the users, and vary N, L, and M in different simulation setup.

TABLE II
Vertices of the optimal solution set Q

	Vertex 1	Vertex 2	Vertex 3	Vertex 4
x^1	2.7352	0.0000	2.7352	0.0000
	0.0000	0.0000	0.0000	0.0000
	21.8216	***21.8216***	***21.8216***	***21.8216***
	0.0000	0.0000	0.0000	0.0000
	0.0000	2.7352	0.0000	2.7352
x^2	0.0000	0.0000	0.0000	0.0000
	14.3455	14.3455	9.2929	9.2929
	0.0000	0.0000	0.0000	0.0000
	5.0526	5.0526	0.0000	0.0000
	29.8307	***29.8307***	***39.9359***	***39.9359***
x^3	***28.2867***	***28.2867***	***28.2867***	***28.2867***
x^4	0.0000	0.0000	0.0000	0.0000
	32.4964	***32.4964***	22.3912	22.3912
	0.0000	0.0000	0.0000	0.0000
	28.1851	28.1851	***33.2378***	***33.2378***
	18.6865	18.6865	23.7392	23.7392
x^5	***22.7468***	***22.7468***	***22.7468***	***22.7468***
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	5.5335	5.5335	5.5335	5.5335

A. Vertices of the optimal solution set of the multipath problem

We fix $N = 5$, $L = 10$, and $M = 5$, and simulate one realization of the network. The five users use 5,5,1,5, and 4 paths respectively. opt_M is 18.2982 in this case, and we also calculate that opt_S is 17.8609 by exhaustive search. Using a vertex enumeration algorithm [16], we explicitly find all four vertices of the optimal polyhedron Q as listed in Table II.

Though lying in $\mathbf{R}^{20}$, the optimal set Q contains only 4 vertices. Moreover, after projecting the solutions to single-path configurations by choosing the maximum-rate paths, there are only two different configurations, and their maximum utilities are 16.6122 and 16.9176 respectively. Then, the upper bounds of the duality gap are 1.686 and 1.3806 respectively, while the actual duality gap is 0.4374. Thus, the vertex projection method can give a good bound on the duality gap, and any vertex of Q is a good candidate. So, in later simulations, we just pick one vertex to estimate the duality gap without enumerating all the vertices.

We observe that the number of paths used by these optimal solutions are 11,11,10 and 10 respectively, while $L+N$ equals 15 in this case. Therefore, the conclusion from Theorem 1 that optimal multipath routing can be achieved with at most $L+N$ paths is a relatively conservative estimate, while it is likely that an optimal multipath configuration requires far fewer paths.

B. Estimation of duality gap as N increases

We fix $L = 50$ and $M = 8$, and let N change from 1 to 60. All the results are averaged over 100 realizations. For each N, we calculate opt_M and find a lower bound of opt_S by vertex projection. Note that their difference is an upper bound of the duality gap. We also randomly choose one

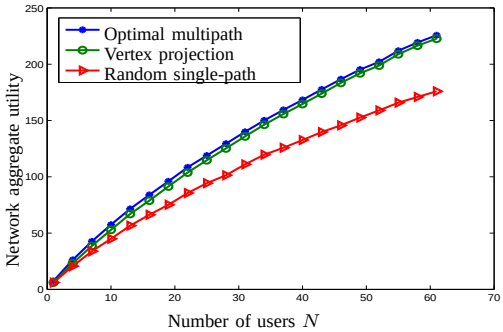

Fig. 4. Comparison of opt_M, a lower bound of opt_S and a random SP utility as N increases

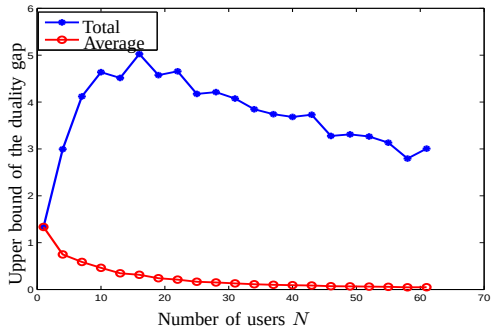

Fig. 5. Total duality and average duality gap as N increases

single-path configuration and calculate its maximum utility. As shown in Fig. 4, all three curves monotonically increase as the number of users increases. The lower bound of opt_S is always near opt_M, while the utility of a randomly chosen single-path routing gradually deviates from opt_M. Though opt_S is always near opt_M, and the duality gap or, equivalently, "the cost of not splitting" is not large, the utility of a randomly chosen single-path configuration can be significantly less than opt_M. Thus, an algorithm to find a near optimal single-path configuration can enhance the performance of the single-path routing. Fig. 5 shows an upper bound of the duality gap and an upper bound of the average cost of not splitting. Although both curves are just upper bounds, they give good estimates of the actual values. So we use them to study the trend of the total and the average cost of not splitting. When the number of users is small ($N < 15$) and the network is partially utilized, users can benefit from using multipaths, thus the cost of not splitting increases as the network supports more users. As the number of users increases, users begin to compete for link capacities with one another, and the benefit of multipath routing is no longer apparent. Thus, the cost of not splitting decreases after $N = 15$. Also, note that the average cost of not splitting monotonically decreases and is near 0 when $N = 60$.

C. Effectiveness of Algorithm 1

We fix $N = 10$, $L = 50$ and $M = 10$. In one realization of the network, the numbers of paths for different users are 6, 6, 2, 6, 5, 5, 2, 8, 10 and 5. opt_M is 38.5573, and the estimate of opt_S found by vertex projection is 34.9073. We can further improve it to 36.1646 in two steps by algorithm 1.

In another realization, the numbers of paths for different users are 8, 10, 8, 5, 6, 8, 10, 3, 3 and 4. opt_M is 38.475, and the estimate of opt_S by vertex projection is 34.2745. It can be further improved to 36.5473 in four steps by algorithm 1.

VII. Conclusion

We have examined the performance loss of joint congestion control and routing when routing is restricted to single-path routing as compared to the case where users can use multiple paths. We demonstrate that the total number of paths needed to achieve the optimal multipath utility is no greater than the sum of the number of links and the number of users. Furthermore, the average performance loss diminishes as the number of users tends to infinity. To estimate the cost of not splitting, we further propose a vertex projection method that can also be combined with a branch-and-bound technique to find a single-path configuration whose maximum network utility gives a good estimate of the optimal single-path utility.

Acknowledgments: The authors thank J. Rexford of Princeton for helpful comments. The research is supported by NSF under CCF-0835706.

References

[1] J. He and J. Rexford, "Toward internet-wide multipath routing," *IEEE Netw.*, vol. 22, no. 2, pp. 16–21, 2008.
[2] D. Xu, M. Chiang, and J. Rexford, "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 466–474.
[3] J. Wang, L. Li, S. Low, and J. Doyle, "Cross-layer optimization in TCP/IP networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 582–595, Jun. 2005.
[4] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *J. ACM*, vol. 46, no. 6, pp. 787–832, 1999.
[5] J. M. Kleinberg, "Single-source unsplittable flow," in *Proc. 37th IEEE Symp. Foundations of Computer Science*, 1996.
[6] ——, "Decision algorithms for unsplittable flow and the half-disjoint paths problem," in *Proc. 30th ACM Symp. Theory of Computing*, 1998.
[7] A. Bley, "On the approximability of the minimum congestion unsplittable shortest path routing problem," *Book Series Lecture Notes in Computer Science*, vol. 3509, pp. 97–110, 2005.
[8] R. K. Nikhil Bansal, Zachary Friggstad and M. R. Salavatipour, "A logarithmic approximation for unsplittable flow on line graphs," in *Proc. 19th Annual ACM-SIAM Symp. Discrete Algorithms*, 2009, pp. 702–709.
[9] S. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.
[10] D. Bertsimas and J. N. Tsitsiklis, Eds., *Introduction to Linear Programming*. Belmont, Massachusetts: Athena Scientific, 1997.
[11] D. P. Bertsekas and N. R. Sandell, "Estimates of the duality gap for large-scale separable nonconvex optimization problems," in *Proc. 21st IEEE Conf. Decision and Control*, vol. 21, Dec. 1982, pp. 782–785.
[12] J. Pongsajapan and S. Low, "Reverse engineering TCP/IP-like networks using delay-sensitive utility functions," in *Proc. IEEE INFOCOM*, May 2007, pp. 418–426.
[13] J. P. Aubin and I. Ekeland, "Estimates of the duality gap in nonconvex optimization," *Mathematics of Operations Research*, vol. 1, no. 3, pp. 225–245, 1976.
[14] S. H. Low, L. L. Peterson, and L. Wang, "Understanding TCP Vegas: a duality model," *J. ACM*, vol. 49, no. 2, pp. 207–235, 2002.
[15] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.
[16] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discrete Comput. Geom.*, vol. 8, no. 3, pp. 295–313, 1992.