TECHNICAL RESEARCH REPORT

Single-path routing of time-varying traffic

*by Abhishek Kashyap, Bobby Bhattacharjee, Richard La,
Mark Shayman, Vahdi Tabatabaee*

TR 2006-5

Single-Path Routing of Time-varying Traffic

Abhishek Kashyap*, Bobby Bhattacharjee†, Richard La*, Mark Shayman* and Vahid Tabatabaee†
*Department of Electrical and Computer Engineering
University of Maryland, College Park, USA
Email: {kashyap, hyongla, shayman}@glue.umd.edu
†Department of Computer Science
University of Maryland, College Park, USA
Email: {bobby, vahid}@cs.umd.edu

***Abstract*— We consider the problem of finding a single-path intra-domain routing for time-varying traffic. We characterize the traffic variations by a finite set of traffic profiles with given non-zero probabilities. Our goal is to optimize the average performance over all of these traffic profiles. We solve the optimal multi-path version of this problem using linear programming and develop heuristic single-path solutions using randomized rounding and iterated rounding. We show through simulations that the single-path routings produced by proposed algorithms are good oblivious routing, i.e., the routing performs well in the worst case as well.**

We analyze our single-path heuristic (finding the optimal single-path routing is NP-Hard), and prove that the randomized rounding algorithm has a worst case performance bound of $O(\log(KN)/\log(\log(KN)))$ compared to the optimal multi-path routing with a high probability, where K is the number of traffic profiles, and N number of nodes in the network. Further, our simulations show the iterated rounding heuristics perform close to the optimal multi-path routing on a wide range of topologies, including synthetic graphs, and measured ISP topologies, in both the average and the worst-case. We also show that the naive shortest-path routing on these same topologies and traffic profiles would lead to much higher congestion. Overall, these results are extremely positive since they show that in a wide-range of practical situations, it is not necessary to deploy multi-path routing; instead, an appropriately computed single-path routing is sufficient to provide both good average and worst-case performance.

Keywords: Traffic engineering, Oblivious routing, Unsplittable flow, Randomized rounding, Mathematical programming/optimization.

I. Introduction

One of the main techniques used to manage network resources and ensure reliable performance in IP networks is intra-domain traffic engineering. Intra-domain traffic engineering uses information about the network traffic profile (traffic matrix) to manage and possibly optimize the network performance. A traffic matrix specifies the expected traffic rate between every ingress-egress pair in the network. The output of traffic engineering is a routing policy f, which is a set of paths and their corresponding relative rate vector. The relative rate vector specifies the fraction of traffic assigned to each path. For optimal traffic engineering, we need to (i) change the routing parameters to adapt to traffic profile variations, which leads to disruption of traffic in the network, along with signaling overhead for forwarding the new routing information [1], and (ii) significantly change the IP forwarding mechanism to support arbitrary traffic distribution among multiple paths between every ingress-egress pair in the network.

In this paper, we propose an approach that uses a *fixed single-path* routing that works well for a given set of traffic profiles. Since the routing is fixed we do not need to change the routing parameters, and since it is single-path we do not need to distribute the load among multiple paths. Specifically, we address the problem of finding a fixed single-path routing for time-varying traffic, characterized by a set of traffic profiles with known probability distribution. In other words, multiple traffic profiles, and the probability distribution on these profiles are given, and our goal is to find a fixed single-path routing policy for all of these profiles.

Let $T_1, T_2, \cdots, T_K$, be the traffic profiles, occurring with non-zero probabilities $p_1, p_2, \cdots, p_K$, respectively. For any given routing policy f and traffic profile T_k, let $\text{Util}_l(f, T_k)$ be the utilization of link l. We want to find a single-path routing policy f that minimizes the average maximum link utilization (with average taken with respect to the traffic profiles). Therefore, the routing policy f^* we seek is,

$$f^* = \arg\min_f \sum_{k=1}^{K} p_k \max_l \text{Util}_l(f, T_k). \tag{1}$$

The traffic profile within a domain can either be predicted by observing the traffic in the network ([2], [3], [4], [5]), or can be inferred from the Service Level Agreements (SLAs). It has been shown that the traffic profile has a pseudo-periodic behavior on different time-scales (like day, week, etc.), which is predictable given past history of the traffic [2], [6]. Thus, the traffic profiles can be estimated based on the previous observations and we can assume the existence of a few traffic profiles sufficient to characterize the traffic over a time period (eg. over a day). Depending on the traffic profiles frequency over the observed history, we can assign a probability distribution on their occurrence.

Given a traffic profile, an optimal multi-path routing can be formulated as a multi-commodity flow (MCF). If the cost function is linear, as is the case in our problem, the MCF problem can be solved by a linear program [7]. In this paper, we extend the MCF formulation to find a routing that minimizes the average cost function of multiple traffic profiles. The problem of routing a single traffic profile using single path per demand is known as the unsplittable flow problem and is NP-Hard [8]. The case of multiple traffic profiles, which we consider, is a generalization of the problem and is thus NP-Hard as well.

We propose two sets of heuristic algorithms for fixed single-path routing. The first algorithm is based on randomized rounding [9], [10], and the second set consists of iterated rounding schemes for the problem. We provide analytical and simulation results which show that the performance of our proposed fixed single-path routing algorithms is very close to the optimal multi-path routing. The main contributions of this paper are:

- Extensive simulation results on the NSF-Net [11] and Rocketfuel [12] topologies show that the randomized rounding algorithm is within 10%, and the iterated rounding algorithm is within 5% of the optimal fixed multi-path routing.
- Simulations show that the iterated rounding schemes work well irrespective of the number of traffic profiles.
- We show that, for probability at least p for any $p \in (0, 1)$, the performance of the randomized rounding algorithm is an $O(\log(KN)/\log(\log(KN)))$-approximation of the optimal multiple path routing. K is the number of traffic profiles and N is the number of nodes in the network.
- For Sprintlink and Tiscali networks [12], which have the properties of small-world topology networks [13], [14], the simulation results show that the randomized rounding and the iterated rounding performance are respectively within 0.4% and 0.2% of the optimal multi-path routing performance.
- Simulation results on small-world topology networks show that for each one of the given traffic profiles $T_1, T_2, \cdots, T_K$, the performance of the iterated rounding algorithm is within 10% of the respective optimal multi-path routings $f_1, f_2, \cdots, f_K$.

The first and third items confirm that for typical network graphs, routings produced by rounding algorithms perform close to optimal. The second item shows that the routing produced by rounding algorithm has scalable performance with respect to the network size and the number of traffic profiles. The last item goes one step further, and for each one of the traffic profiles, compares the iterative rounding algorithm routing to the multi-path routing produced by the optimal algorithm for the corresponding traffic profile. In fact, in this way, we are comparing our algorithm to an idealistic multiple-path routing policy that can simultaneously adapt to the traffic profile variations.

A. Related Work

The idea of having a fixed routing for multiple traffic profiles in the OSPF/IS-IS framework was proposed in [1]. The authors consider multiple traffic profiles, and provide a set of OSPF/IS-IS link weights which works well for all the traffic profiles. They give algorithms based on local search, after starting from an initial set of link weights. Then, OSPF or IS-IS routing uses the weights for routing the traffic in the network. We consider the problem of finding optimal routes directly rather than finding OSPF/IS-IS weights. Another work that considers multiple traffic profiles is that of joint logical topology configuration and routing of traffic on lightpaths in MPLS over WDM networks [15]. They formulate the problem as an Integer Linear Program (ILP) and use space-reduction heuristics to find a feasible solution. Then, they use the static routing inside the domain. Optimal source-destination multi-path routing and destination multi-path routing algorithms for multiple traffic matrices have been proposed in [16]. The objective considered in [16] is the average performance over the traffic matrices, as in our problem. Another performance metric has been proposed for multi-path routing in [17] that takes a weighted average of average and worst case performance of the routing.

Oblivious routing has recently been proposed as a static routing good for the space of all traffic matrices. The objective of oblivious routing is to find a routing f which minimizes the objective function (called *oblivious ratio*) of Equation 2, i.e., it minimizes the maximum of the ratio of the maximum link utilization of routing f for traffic profile t to the maximum link utilization of optimal routing OPT_t for traffic profile t, with t being in the space of all possible traffic profiles T.

$$O(f) = \max_{t \in T} \frac{\max_l \text{Util}_l(f, t)}{\max_l \text{Util}_l(OPT_t, t)} \quad (2)$$

Optimal oblivious multi-path routing algorithms have been proposed in [18] and [19]. The first optimal polynomial-time algorithm for finding an oblivious multi-path routing was given in [18], but it uses Ellipsoid algorithm [20], which is extremely slow. A single linear program (LP) for finding the optimal routing was given in [19]. We consider our problem (with objective of Equation 1), that is different from the oblivious routing problem, due to the following reasons: First, oblivious routing looks at performance relative to the optimal routing for each traffic matrix. As an example to illustrate why this can lead to a suboptimal routing, consider a situation in which there are a few low-demand traffic profiles that have a low maximum link utilization for a routing that is good (relative to optimal) for other traffic profiles. But, the ratio between the maximum utilization of this routing to the optimal for these low-demand profiles may be very high. Thus, even though the congestion caused for these profiles for a routing good for other profiles is low, these profiles may affect the determination of optimal oblivious routing and lead to a routing that is not as good for traffic profiles which have a high load on the network. Second, oblivious routing considers the worst case performance among the traffic profiles. There may be traffic profiles with very low probability of occurrence, and considering the worst case performance among all traffic profiles may give a routing with a worse performance most of the time compared to the routing given by our algorithms that consider the average performance over the traffic profiles. Third, the traffic profiles are pseudo-periodic and not usually totally unpredictable, and can be assumed to be from among a discrete set of traffic profiles [1], [15]. Thus, considering only a discrete set of traffic profiles is sufficient, and the complexity introduced by considering the whole traffic profile space can be avoided. Fourth, the problem we consider is easier to extend to finding a single-path routing, which is simpler and enables efficient fair queueing, whereas the LP formulation of [19] is too complex for any analysis when extended to single-path routing. We also show by simulations that for the given set of traffic profiles, the oblivious ratio of the optimal multi-path routing for our objective function is very low. Thus, our routing strategy is good in the oblivious sense too.

For unsplittable routing of a single profile ($K = 1$), an $O(\log(N)/\log(\log(N)))$-approximation randomized algorithm was proposed in [21]. For the case when the maximum path length for each demand is restricted to d in the MCF formulation, an $O(\log(d)/\log(\log(d)))$ randomized algorithm is provided in [22], and deterministic algorithms of the same approximation factor using the method of pessimistic estimators [23] have been proposed in [24].

The organization of this paper is as follows: Section 2 gives the network model and a formal statement of the problem. Section 3 gives the MCF formulation of the multi-path routing problem, and gives simulation results illustrating the performance in terms of the oblivious ratio. Section 4 gives the single-path routing algorithms. Section 5 gives the simulation results, and Section 6 concludes the paper. The proof of approximation ratio for the randomized rounding algorithm is given in the Appendix.

II. Network Model and Problem Statement

The network consists of routers, and bidirectional links between pairs of routers (nodes) forming a topology. We denote the set of links between the nodes as E_l. We model the network by a graph $G = (V, E)$, where the vertices in V are nodes in the network, and E is the set of unidirectional *edges* between pairs of vertices, with two anti-parallel edges for each bidirectional link in E_l. We use the term $link$ when we refer to bidirectional links and $edge$ when we refer to one of the corresponding unidirectional edges. We assume all the links have the same capacity, thus the traffic rate on the links represents the utilization of all links. The algorithms work for non-uniform link capacity as well.

We are given a collection of traffic matrices (profiles) $\{T_1, .., T_K\}$ with probabilities of occurrence $\{p_1, .., p_K\}, \sum_{k=1}^{K} p_k = 1$. Each traffic matrix is a set of traffic demands between ingress-egress node pairs, and each traffic demand is between 0 and 1. The set of ingress-egress pairs is assumed to be the same in all traffic profiles. The objective is to find a routing which minimizes the mean maximum link load (total load on a bidirectional link) in the network[1]. Equation 3 represents the cost function for routing f, and our objective is to find a routing that minimizes it. Here, $t^k_{i,j}$ represents the traffic demand between source i and destination j in traffic profile T_k. $f^l_{i,j}$ represents the fraction of flow between source i and destination j routed through (bidirectional) link l, i.e., it is the sum of total traffic flowing through the two anti-parallel edges corresponding to link l in E. For multi-path routing, $f^l_{i,j} \in [0, 1]$ while for single-path routing, $f^l_{i,j} \in \{0, 1\}$. The formulation can be easily extended to work with multiple classes of traffic between each ingress-egress pair by indexing the traffic demands as (source,destination,class).

[1]The formulation and algorithms also work with the objective of minimizing mean maximum edge load with a slight modification.

TABLE I
NOTATION

Symbol	Definition
T_k	Traffic profile k
p_k	Probability of occurrence of profile T_k
$t^k_{i,j}$	Demand between ingress-egress pairs i, j in profile T_k
σ_k	Maximum edge utilization for profile T_k
$f^e_{i,j}$	Fraction of demand between i, j routed on edge e
e^1_l, e^2_l	Anti-parallel edges for link l
out_n, in_n	Set of outgoing and incoming edges at node n

$$cost(f) = \sum_{k=1}^{K} (p_k \max_l \sum_{(i,j)} t^k_{i,j} f^l_{i,j}) \tag{3}$$

III. Linear Program for Splittable Traffic

For routing unsplittable demands, we propose some algorithms which find an optimal multi-path routing and then select a single path for each traffic demand from the set of paths in the optimal multi-path solution. In this section, we present the algorithm to compute an optimal multi-path routing. We formulate the problem as a multi-commodity flow problem with a linear objective function, which can be formulated as a single linear program (LP). We call each entry of a traffic profile as a (traffic) demand between an ingress-egress pair. The linear program is as given in Equation 4. The notation is given in Table I, and explained below. The sets of outgoing and incoming edges at vertex n are denoted by out_n and in_n respectively. The fraction of traffic demand for an ingress-egress pair (i, j) through edge e is represented by $f^e_{i,j}$. The anti-parallel edges corresponding to link l are denoted by e^1_l and e^2_l. The first constraint along with the objective function minimizes the mean maximum link load. The second, third and fourth constraints are flow conversation laws for the routing f. The third constraint ensures the total flow fraction going out of a source is one, while the second constraint ensures the total outgoing and incoming flows for a traffic demand are equal at the nodes which are not a source or destination for the traffic demand. The fourth constraint ensures the total traffic going out of the destination of a demand is zero. The output of the LP is a routing f, which is used to route all the traffic profiles. The last constraint bounds the routing variables between 0 and 1.

$$
\begin{aligned}
&\text{Minimize } \sum_{k=1}^{K} p_k \sigma_k \\
&\text{s.t.} \\
&\sum_{(i,j)} t_{i,j}^{k} (f_{i,j}^{e_l^1} + f_{i,j}^{e_l^2}) \ \leq \ \sigma_k \ , \forall l \in E_l, k \in \{1,..,K\} \\
&\sum_{e \in in_n} f_{i,j}^{e} = \sum_{e \in out_n} f_{i,j}^{e}, \forall n \in \{1,..,N\} - \{i,j\}, \forall (i,j) \\
&\sum_{e \in out_i} f_{i,j}^{e} \ - \sum_{e \in in_i} f_{i,j}^{e} = 1, \ \forall (i,j) \\
&\sum_{e \in out_j} f_{i,j}^{e} \ = 0, \ \forall i,j \\
&f_{i,j}^{e} \ \geq \ 0 \ \forall i,j,e
\end{aligned} \tag{4}
$$

The output of the LP may give a routing with loops if the load on the links in the loop is less than the maximum link load in the network. The loops are removed by subtracting the flows of each demand between the anti-parallel edges of each link.

A. Oblivious Ratio Performance

We show via simulations that the routing computed by the LP formulation to minimize the mean maximum link load works well according to the oblivious ratio. The oblivious ratio of a routing f is the maximum over the traffic profiles (T_k) of the ratio between the maximum link load of f for the traffic profile and the maximum link load of the optimal routing for the traffic profile (Equation 5). $\text{Util}_l(f, T_k)$ represents the load on link l when routing f is used to route the traffic profile T_k. OPT_k represents the optimal routing for minimizing the maximum link load for traffic matrix T_k.

$$
O(f) = \max_k \frac{\max_l \text{Util}_l(f, T_k)}{\max_l \text{Util}_l(OPT_k, T_k)} \tag{5}
$$

We evaluate the routing given by the LP of Equation 4 for oblivious ratio on the NSF-Net topology (Figure 1) and the Exodus network topology (Figure 2) given by Rocketfuel [12]. We assume each node in the network can be a source and a destination. The traffic profiles are generated randomly, with each traffic demand being an independent uniform random variable between $[0, 1]$. The traffic profiles are assumed to have equal probabilities $p_k = 1/K$. The number of traffic profiles is varied from 5 to 50, and as the number of traffic profiles is increased in each simulation, the set of profiles includes the previous traffic profiles as well. The simulations are run 20 times, and the oblivious ratio of the routing is noted. Figures 3 and 4 show the average and maximum of oblivious ratio of the routings output by the LP of Equation 4 over the 20 simulations for the NSF-Net and Exodus topologies respectively. The oblivious ratio is very small (within 1.2 in the worst case for NSF-Net and 1.14 for Exodus), and thus the algorithm works well in terms of oblivious ratio. Note that even though the traffic profiles in all simulations include the traffic profiles in corresponding simulation with less number of traffic profiles, the maximum oblivious ratio of our algorithm

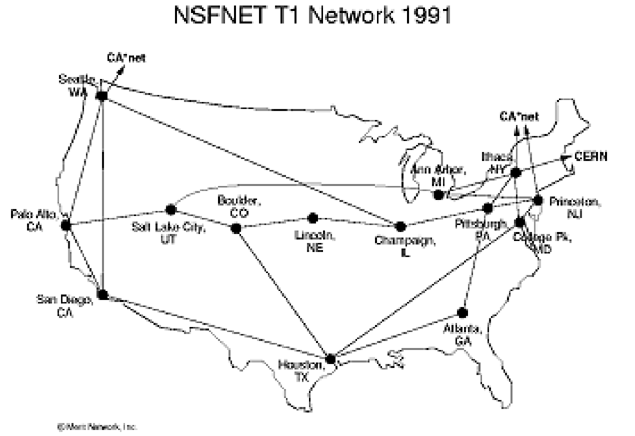

Fig. 1. NSF-Net Backbone Topology

Fig. 2. Exodus US Backbone Topology

can decrease with increase in number of traffic profiles in some instances. This is because our algorithm is not an optimal oblivious routing, and so addition of traffic profiles to a set of traffic profiles may give a routing which works better in terms of the oblivious ratio than the routing found on the original set of profiles for some instances. The average performance in terms of oblivious ratio gets worse with increasing number of traffic profiles, as is expected.

IV. Single Path Routing of Traffic Flows

We now discuss the algorithms for finding a single-path routing. The problem of routing traffic demands on single paths to minimize congestion is NP-Hard [8]. Thus, we resort to heuristic algorithms to compute the routing. Changing the bounds on the routing variables in the LP of Equation 4 from $[0, 1]$ to $\{0, 1\}$ would make sure only a single path is selected for each demand, but this is an Integer Linear Program, and solving it is NP-Hard as well. We solve the LP to get an optimal multi-path solution and round the solution to get a single path from among the corresponding multiple paths for each traffic demand[2]. We propose a few deterministic rounding schemes and randomized rounding schemes. We prove an

[2]We show by simulations that single-path routing performs nearly as well as optimal multi-path routing, so single-path routing is sufficient.

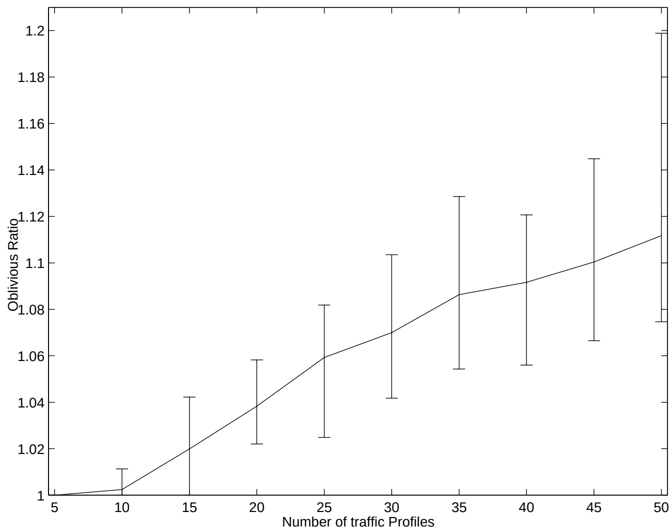

Fig. 3. Oblivious ratio of our optimal algorithm on the NSF-Net topology

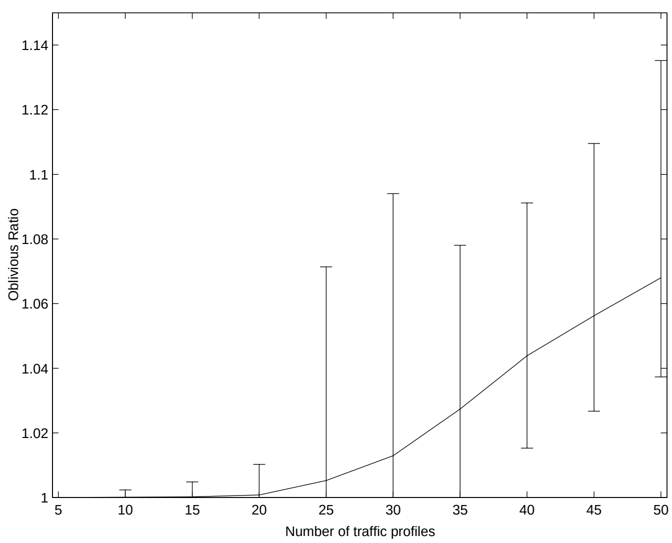

Fig. 4. Oblivious ratio of our optimal algorithm on the Exodus topology

approximation ratio bound for one of the randomized rounding algorithms.

After solving the LP, we perform path decomposition [25] on the routing variables. This gives a set of paths for each traffic demand, each path having a value assigned to it that represents the fraction of the traffic demand being routed through the path. Then, we perform rounding on the fractional path assignments to get an integer solution, i.e., we select one path from the set of paths corresponding to each traffic demand for routing. We propose different rounding algorithms, each following the same procedure, but doing the rounding (selecting the paths) according to a different criteria. The outline of the algorithms is as follows:

1) Solve the LP of Equation 4.
2) Use path decomposition to get l_i paths for each traffic demand i. Let $x_{i,j}$ denote the fraction of traffic carried by path j of demand i. For each i, $x_{i,j}$'s sum to 1.
3) For each traffic demand i, round one of the $x_{i,j}$'s to 1, and the rest to zero, i.e., select path j according to some criteria.

A. Shortest Path Rounding

In shortest path rounding, the shortest path (chosen arbitrarily if multiple exist) among the paths given by the LP is selected for each traffic demand. As selecting the shortest path from the candidate paths utilizes minimum network resources, so this strategy is a natural strategy for rounding.

B. Maximum Utilization Rounding

Among the set of paths for a traffic demand i, the one with the maximum fraction ($x_{i,j}$ on path j) routed through it can be viewed as the most favored by the LP. A path more favored in the LP solution is expected to lead to the lowest value of the objective function. Thus, in this rounding strategy, we pick the path with the maximum fraction assigned to it, and the shortest path if multiple such paths exist. We call this path as the maximum utilization path for a traffic demand. The path decomposition is not unique, so this path may be different under different decompositions, but once a decomposition is done, the path with the maximum fraction assigned is the most favored.

C. Randomized Rounding

In randomized rounding, for each traffic demand i, we treat the fraction of demand routed on each path as the probability of its occurrence and round one of the $x_{i,j}$'s to 1 with probability $x_{i,j}$, round the remaining to 0. We treat this rounding as rolling an l_i-face dice for each traffic demand i with face probabilities equal to $x_{i,j}$'s. The resulting paths form the solution. The whole rounding procedure is repeated a fixed number of times and then until the ratio of the standard deviation of the objective value (over the repetitions) and average value of the objective falls below a certain threshold ϵ. We take the threshold to be 0.1 for simulations, and the minimum number of repetitions is taken to be 10. The best solution from the repetitions is taken as the output. We prove that the randomized rounding described above has a worst case performance bound of $O(\log(KN)/\log(\log(KN)))$ relative to the optimal with probability at least p for any $p \in (0,1)$. Here, N is the number of nodes in the network and K is the number of traffic profiles. Thus, in the worst case, the algorithm will produce a solution within $O(\log(KN)/\log(\log(KN)))$ times the optimal in a finite number of repetitions. The bound is stated in Theorem 4.1.

Theorem 4.1: Randomized rounding produces a solution with approximation ratio of $O(\log(KN)/\log(\log(KN)))$ with probability at least p for any $p \in (0,1)$.

Proof: See Appendix.

D. Iterated Rounding

We propose another set of heuristics, in which we perform *iterated rounding*, i.e., we round the paths for a few demands, fix the paths for these demands in the solution, resolve the LP and repeat the procedure. The criteria for selecting the traffic demands to be rounded in an iteration is by selecting the demands with maximum sum of the squares of the path utilizations. The reason for using this measure is that the demands with higher value of this measure will have a path with high fraction routed through it (indicating a strong preference for this path by the LP), and thus will incur less penalty while rounding. A demand with a lower value of this measure has a more even distribution of traffic among the paths and thus incurs more penalty on the objective value by rounding. So, at each step, we round only a few demands which are expected to increase the objective value the least among all the traffic demands. The algorithm is given in Algorithm 1.

Algorithm 1 Iterated Rounding

1: Define the set of demands as T. Set $U = T$.

2: While $U \neq \emptyset$, repeat:

- (a). Solve the LP of equation 4.
- (b). Do path decomposition for traffic demands in U to get the *candidate paths* for each demand. Let l_i denote the number of candidate paths for demand i. Let the fraction routed on each path $j \in \{1, .., l_i\}$ for demand i be $x_{i,j}$.
- (c). Find the set $R \subseteq U$ of demands according to Equation 6. Here, c is a constant used to include demands for which the measure is close to the maximum. We take c to be 0.9 in the simulations.

$$R = \{i | i \in U, \sum_{j \in \{1,..,l_i\}} x_{i,j}^2 \geq c u_{max}\}$$
$$u_{max} = \max_{i \in U} \sum_{j \in \{1,..,l_i\}} x_{i,j}^2 \qquad (6)$$

- (d). Select a path from the set of candidate paths for each demand in R according to some rounding scheme (like maximum utilization or randomized rounding).
- (e). Set $U = U \backslash R$, fix the paths of demands in R and repeat Step 2 if $U \neq \emptyset$.

3: Output the resulting paths.

We can use any rounding scheme to round the selected demands at each step of the algorithm. We propose the use of three rounding schemes to do the rounding: **Iterated Maximum Utilization Rounding**, **Iterated Randomized Rounding** and **Iterated Hybrid Rounding**. Iterated maximum utilization rounding uses maximum utilization rounding and iterated randomized rounding uses randomized rounding. In iterated randomized rounding, the whole iterated rounding procedure is repeated multiple times according to the same criteria as in randomized rounding. There are no repetitions within a run of the iterated rounding procedure.

Fig. 5. Sprintlink US Backbone Topology

Simulations show that the two iterated rounding schemes have similar performance. Maximum utilization rounding of a demand incurs a higher penalty in an iteration when the fractions are evenly split for the demands selected in set R. Thus, doing randomized rounding for such demands is expected to give a solution at least as good as maximum utilization rounding of these demands, if the randomized rounding is repeated sufficient number of times. For such demands i, the sum of the squares of the $x_{i,j}$'s over j (Equation 6) is low.

In iterated hybrid rounding, we modify the iterated maximum utilization rounding to do randomized rounding whenever the maximum value of the measure in Equation 6 is less than a threshold. We call this algorithm iterated hybrid rounding. We set the threshold at 0.55 for the simulations, after observing the threshold below which the rounding penalty is high for maximum utilization rounding. As there is a randomization component for some traffic demands in this algorithm, so as in iterated randomized rounding, we repeat the whole iterated rounding procedure multiple times (according to the same criteria as before).

V. Simulation Results and Discussion

We implemented the algorithms in C, and used CPLEX [26] for solving the linear programs. The experiments are done on the NSF-Net topology (Figure 1: 14 nodes, 21 bidirectional links), the Exodus topology (shown in Figure 2: 15 nodes, 33 bidirectional links), the Sprintlink topology (Figure 5: 27 nodes, 69 bidirectional links) and the Tiscali topology (51 nodes, 129 bidirectional links). Exodus, Sprintlink and Tiscali topologies are taken from Rocketfuel [12]. The traffic profiles are taken to be equiprobable in all simulations.

A. Comparison of rounding schemes

The first set of experiments are done with 20 randomly generated traffic profiles, with each traffic demand in each traffic profile being a uniform r.v. between 0 and 1. The traffic profiles are assumed to be equiprobable. All nodes are assumed to be sources and destinations, and traffic demands are generated between all pairs. The simulations are run 20 times, and the objective value for the solution of single-path routing algorithms is compared to the LP optimal value (optimal multi-path solution given by Equation 4). For each simulation, the randomized rounding algorithms are run at least 10 times (and at most until the conditions, given in algorithm description, on the mean and standard deviation values of the results are met). We note that 10 runs of the randomized algorithms are always sufficient, as the solutions of each run are very close to each other. The optimal single-path routing has the objective value at least as high as the LP optimal, as the LP feasible solution set contains the ILP feasible solution set. Thus, the performance relative to LP optimal is an upper bound for the performance relative to the single-path optimal routing.

Figures 6, 7 and 8 show the ratio of the objective values given by the single-path algorithms and the LP optimal value for the NSF-Net, Exodus and Sprintlink topologies. Table II gives the average and the maximum value of this ratio (over the 20 runs) for the topologies. For the Exodus and NSF-Net topologies, all the iterated rounding algorithms give a solution with the objective value within 5% of the LP optimal value, and the iterated hybrid rounding works slightly better than other iterated rounding schemes. Also, the solutions returned by the iterated rounding algorithms and the randomized algorithm are all in a very small performance range, whereas the solution returned by the maximum utilization rounding and shortest path (SP) rounding deviate from the average much more, and lead to a very high rounding penalty in the worst case. For the Sprintlink topology, all the rounding schemes work very close to the LP optimal (the worst performance for SP rounding being 1.5%, and the worst performance for iterated rounding schemes being 0.14%). Simulations show that the solution returned by the LP has very few traffic demands being split among multiple paths, i.e., the routing consists of most demands being routed on single paths. Thus, even SP rounding, which led to a 51% penalty in the objective value over the LP solution in the worst case in simulations on Exodus, gives only a 1.5% worst case penalty for Sprintlink. Also, iterated randomized rounding gives a solution within 0.1% of the LP optimal for all simulations on Sprintlink. This suggests that the optimal single-path routing is very close to the optimal routing that allows multi-path.

The results show that all the iterated rounding schemes have similar performance on all the topologies. Thus, we use iterated maximum utilization rounding scheme as a representative iterated rounding scheme for rest of the simulations. The iterated maximum utilization rounding algorithm is evaluated on the Tiscali network for 10 randomly generated traffic profiles and the simulations are done 20 times (with a different set of equiprobable profiles in each). Figure 9 gives the average and maximum value of the performance of the single-path routing algorithm with respect to the LP optimal objective value. Simulations show that the computed single-path routing has the same performance as optimal multi-path routing in half the simulations, and the rounding penalty is only 0.15% in the worst case. Thus, single-path routing is near-optimal (optimal multi-path) for the Tiscali network too.

The near-optimal performance of single-path routing on Sprintlink and Tiscali networks shows that single-path routing is sufficient on some topologies. The topologies of Sprintlink and Tiscali have a **few** very high-degree nodes and other nodes have a much lower degree (i.e., there is some sort of clustering and hierarchy in the backbone network). This leads to a few short hop paths between all nodes in the network, and most paths go through the set of high-degree nodes and thus the optimal multi-path routing selects mostly single paths using the high-degree nodes for each source-destination pair. In NSF-Net and Exodus topologies, all the nodes have similar degrees and thus the optimal routing gains over single-path routing by splitting the traffic among multiple paths. We will show by simulations on random graphs that increasing the degree at all nodes actually increases the performance gap between multi-path and single-path routing. Thus the key idea is to have a few nodes having a very high degree compared

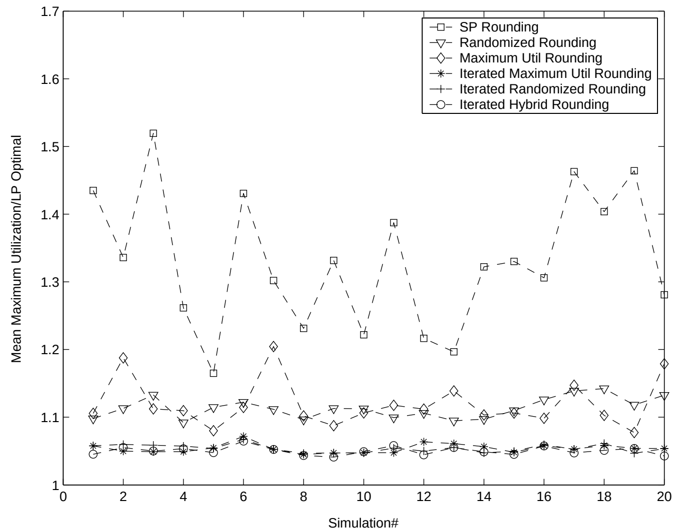

Fig. 6. Performance relative to LP optimal on the Exodus topology

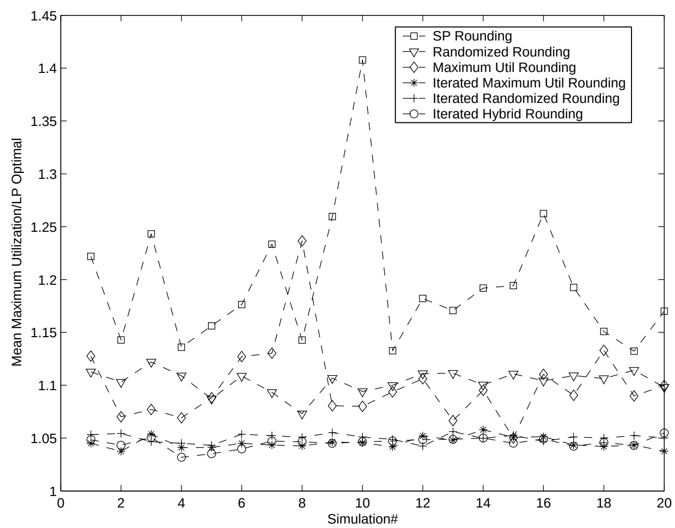

Fig. 7. Performance relative to LP optimal on the NSF-Net topology

TABLE II

AVERAGE AND WORST CASE PERFORMANCE ON DIFFERENT TOPOLOGIES

Topology		SP Round.	Rand. Round	Max. Util. Round.	Iter. Max. Util. Round.	Iter. Rand. Round.	Iter. Hybrid Round.
NSF-Net	Mean	1.1950	1.1037	1.1011	1.0455	1.0503	1.0454
	Maximum	1.4077	1.1220	1.2365	1.0579	1.0560	1.0549
Exodus	Mean	1.3302	1.1135	1.1195	1.0538	1.0538	1.0502
	Maximum	1.5194	1.1422	1.2044	1.0717	1.0671	1.0647
Sprintlink	Mean	1.0051	1.0010	1.0034	1.0003	1.0002	1.0003
	Maximum	1.01433	1.0031	1.0080	1.0014	1.0010	1.0014

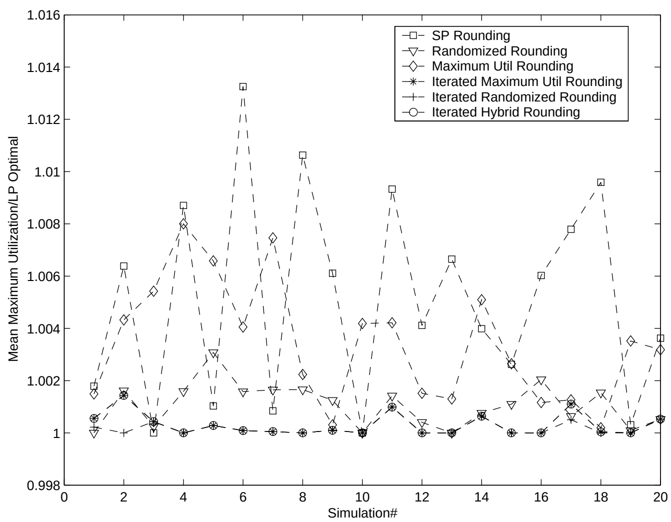

Fig. 8. Performance relative to LP optimal on the Sprintlink topology

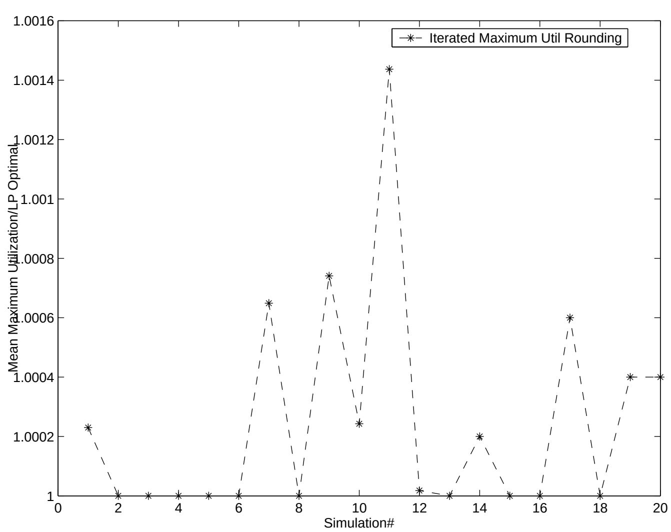

Fig. 9. Performance of iterated maximum util. rounding relative to LP optimal on the Tiscali topology

to the other nodes, leading to clustering, as is the case with Sprintlink and Tiscali networks. Networks with this property are called small-world networks [13], [14], and it has been shown that the Internet network is a small-world network at both router level and inter-domain level [14].

B. Simulations with varying number of traffic profiles

We next illustrate the variation in performance of the schemes with the number of traffic profiles. The number of traffic profiles is varied from 1 to 50, and the simulations are done on the NSF-Net topology. The traffic profiles are assumed equiprobable in each simulation. The performance measure is again the ratio of objective value for the rounding scheme to LP optimal. The simulations are run 20 times, and the algorithms compared are the randomized rounding, maximum utilization rounding and iterated maximum utilization rounding. Figure 10 shows the average and maximum of the performance measure over the 20 simulations for different number of traffic profiles. Results show that the performance of the randomized and maximum utilization rounding schemes degrades as the number of traffic profiles is increased. The maximum utilization rounding gives much worse worst case performance than randomized rounding and iterated maximum utilization rounding, as the previously discussed simulations also showed. The performance is expected to degrade, as is suggested by Theorem 4.1. The degradation in performance with increasing number of traffic profiles is due to the increase in the number of correlated link load constraints in the LP (there is a set of link load constraints for each traffic profile). Thus, rounding a demand affects more constraints if there are more traffic profiles and leads to a higher penalty. This correlation between constraints is the cause of the introduction of the number of traffic profiles (K) in the bound of Theorem 4.1.

For the iterated rounding scheme, there is hardly any degradation with the increase in number of traffic profiles. Also, the worst case performance is very close to the average case performance for the iterated rounding scheme. Thus, the single-path routing computed using the iterated algorithm works well irrespective of the number of traffic profiles. Note that the performance of single-path routing relative to optimal multi-path routing can improve with increase in the number of profiles in some simulation instances. This is because the optimal multi-path routing changes with addition of traffic profiles, and thus the rounding penalty can be less for this routing than for the optimal multi-path routing for a smaller

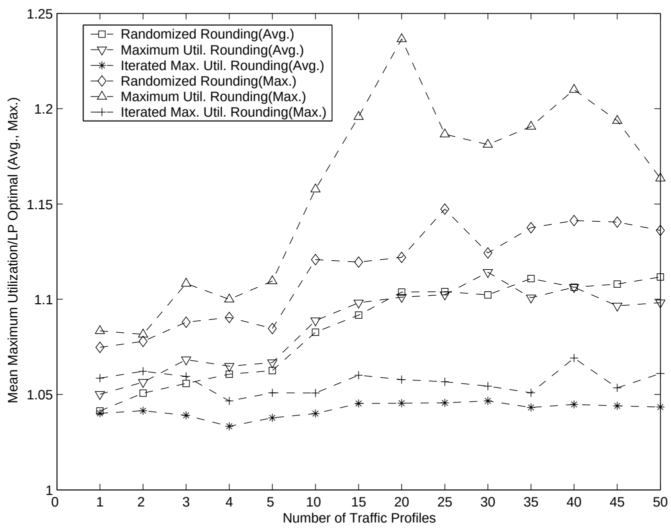

Fig. 10. Average and worst case performance relative to LP optimal for different number of traffic profiles on the NSF-Net topology

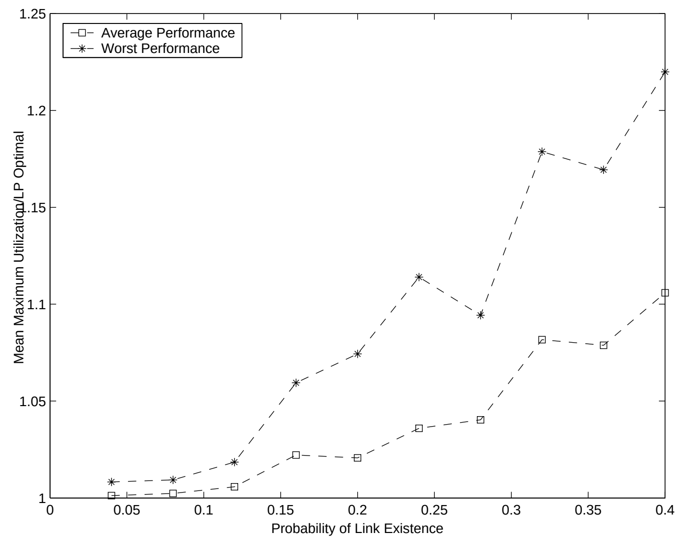

Fig. 11. Average and worst case performance relative to LP optimal of Iterated Max. Util. Rounding on random graphs

set of traffic profiles.

C. Results on random graphs

We next evaluate the algorithms on random graphs. We generate random graphs of type $\mathcal{G}(n, p)$, where n is the number of nodes in the network, and p is the probability of forming a link between a pair of nodes [27]. For the simulations, n is taken to be 20, a set of 20 traffic profiles is generated at random (and used in all simulations), p is varied between 0.04 and 0.4, and the algorithms are evaluated on 10 randomly generated graphs for each value of p. Figure 11 shows the performance of iterated maximum utilization rounding in terms of the ratio of objective value to the LP optimal objective value. It shows both the average and maximum over the 10 randomly generated graphs for each value of p. The performance of the algorithm degrades with the increase in the value of p as the number of alternate paths between each source-destination pair increases, and so do the gains achieved by routing over multiple paths. In the routing given by the LP (optimal multi-path routing), more routes are expected to split over paths, and over more paths for a higher value of p, and thus the rounding leads to a higher penalty. However, as worst case performance shows, the algorithm may have a better performance on a random graph generated with a higher value of p than a random graph generated with a lower value of p in some instances, as the graph generation is random.

Thus, increasing the degree of nodes increases the performance gap between multi-path and single-path routing, and multi-path routing is more desirable for networks with many high-degree nodes.

D. Comparison with min-hop routing for link loads

We also compute a single-path routing using minimum number of hops for each source-destination pair. We evaluate the min-hop routing with respect to the routing computed by iterated maximum utilization rounding for 20 simulations with 20 randomly generated traffic profiles in each simulation. For each simulation, we find the mean of the link utilization (taken with respect to the traffic profile distribution, which is taken to be uniform) of each link, and normalize the mean utilization at each link for both min-hop and our algorithm by the maximum among the links of mean link utilization achieved by our algorithm in that simulation. Figures 12 and 13 show the distribution of the average number of links (over 20 simulations) for min-hop algorithm having mean link loads greater than the maximum of mean link loads for our algorithm, for the NSF-Net and Exodus topologies. Each bar represents the average number of links which had the normalized mean utilization using min-hop routing in a particular range. The min-hop algorithm gives an mean link-load value as high as 2.2 times the maximum of mean link load given by our algorithm on the Exodus topology, and 1.3 times on the NSF-Net topology. Also, the average number of links having mean link loads for min-hop algorithm greater than our algorithm is 10 links (out of total 33 links) for the Exodus topology and 6 links (out of 20 links) for the NSF-Net topology. Thus, min-hop algorithm would cause more congestion than the maximum congestion for our algorithm on about 30% of the links on both the topologies.

E. Evaluation of oblivious ratio

We also evaluate the routing given by iterated maximum utilization rounding algorithm for the oblivious ratio, i.e., the worst case performance over the traffic profiles. We compute the single-path routing for 20 simulations, each having a set of 20 randomly generated equiprobable traffic profiles. For each simulation, we compute the optimal multi-path routing for each traffic profile, and take the ratio of the maximum link utilization by the single-path routing to the optimal multi-

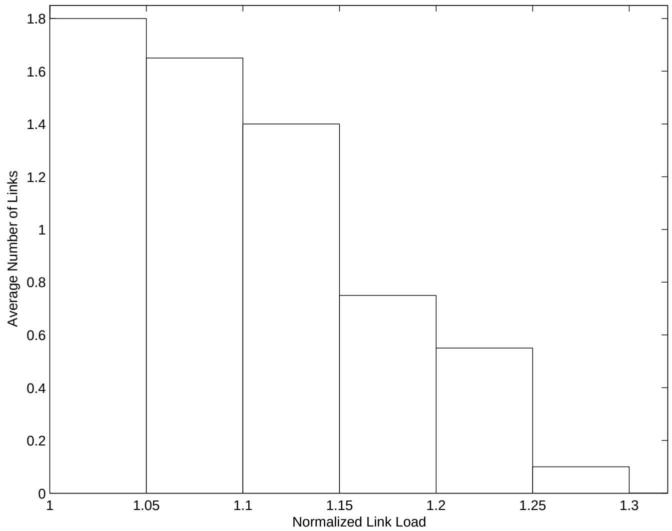

Fig. 12. Distribution of links with mean load (using min-hop routing) greater than maximum load for iterated max. util. rounding on the NSF-Net topology

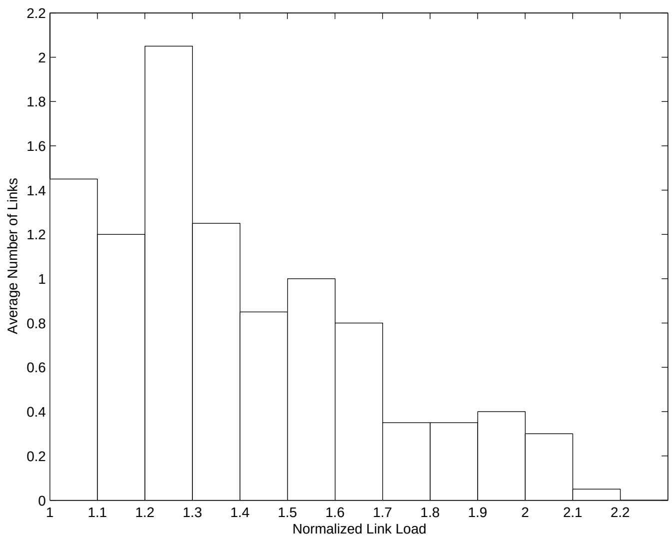

Fig. 13. Distribution of links with mean load (using min-hop routing) greater than maximum load for iterated max. util. rounding on the Exodus topology

path routing for each traffic profile. The maximum of this ratio over the traffic profiles gives an upper bound on the oblivious ratio of our single-path routing as the optimal single-path routing cannot perform better than the optimal multi-path routing for each traffic profile. Table III gives the average and maximum of the oblivious ratio over the 20 simulations for the NSF-Net, Exodus, Sprintlink and Tiscali topologies. The average oblivious ratio bound is around 1.16 for NSF-Net and Exodus, and maximum is 1.36 for NSF-Net and 1.25 for Exodus. Thus, the oblivious ratio (worst case performance over the traffic profiles) for the single-path routing is reasonable, considering that they give close to optimal performance for our objective function, which optimizes the average performance

TABLE III
AVERAGE AND WORST CASE OBLIVIOUS RATIO UPPER BOUNDS FOR SINGLE PATH ROUTING USING ITERATED MAX. UTIL. ROUNDING ON DIFFERENT TOPOLOGIES

Topology	Avg. Oblivious Ratio	Max. Oblivious Ratio
NSF-Net	1.1602	1.3606
Exodus	1.1556	1.2485
Sprintlink	1.0043	1.0162
Tiscali	1.0163	1.0878

over the traffic profiles. For Sprintlink and Tiscali, the upper bound on oblivious ratio is within 1.02 and 1.09, which can be considered quite close to optimal. Thus, the small-world property leads to an average case optimal routing performing close to optimal in the oblivious (worst-case) sense too.

VI. Conclusion

We consider the problem of single-path routing of time-varying traffic. We model the traffic as a discrete set of traffic profiles, and propose LP rounding-based heuristics. The simulations show the multi-path LP formulation of the problem is a good routing according to the oblivious ratio measure. For the single-path routing formulation, we propose the use of a randomized rounding algorithm and prove it to be an $O(\log(KN)/\log(\log(KN)))$-approximation algorithm with probability at least p for any fixed $p \in (0,1)$, with K being the number of traffic profiles, and N being the number of nodes in the network. We propose some iterated rounding schemes which give a performance within 5% for the Rocketfuel [12] topologies considered. For the Sprintlink and Tiscali topologies, the optimal multi-path algorithm gives a routing consisting of mostly single-paths, and thus rounding the multi-path solution to single-path routing leads to an insignificant increase in the objective value. The single-path routing computed by iterated algorithms are shown to be within 0.2% of the optimal on these ISP topologies. The characterizing property of these topologies is the existence of some very high-degree nodes and other nodes having low degree; such networks are called small-world networks. Also, on small-world networks, the single-path routing is shown to be close to optimal (oblivious ratio) in the worst case as well.

Acknowledgement

The authors would like to thank Aravind Srinivasan for his guidance on the existing randomized rounding approximation ratio proofs, and suggestions on the rounding heuristics.

Appendix

Proof of Theorem 4.1

We now prove the approximation ratio of Theorem 4.1 for the randomized rounding algorithm. The problem of finding single-path routing for multiple traffic profiles can be written as an integer linear program (ILP). The linear relaxation of the ILP is given by the LP of Equation 4. In randomized rounding, the LP is solved, and the paths are selected from the paths given by the LP randomly. Let the number of nodes in the network be N, the number of links in the network be m, the number of traffic demands in each traffic profile (assumed to be same, some can be zero in some profiles) be n, and the number of traffic profiles be K. Let the number of paths after path decomposition of the LP solution be l_i for each traffic demand i. We will write a sequence $\{1,..,k\}$ as $[k]$ from now on. Let the variables $x_{i,j}$ represent the fraction of demand i routed on path $j \in [l_i]$. Let p_k denote the probability of occurrence of traffic profile k. Also, let y^* be the objective value for the solution returned by the LP, and y^*_k be the maximum link load in the network when the routing given by the LP is used on traffic profile T_k. Thus, $y^* = \sum_k p_k y^*_k$.

If there is just one traffic profile ($K = 1$), the problem of rounding the routing given by the LP to a routing using single path per demand can be formulated as a minimax integer program (MIP) [9]. An MIP for the problem can be written as in Equation 7. Here, $\overrightarrow{W}$ is a m-dimensional vector with all entries as W and $A \in [0,1]^{m\times L}$, where $L = \sum_{i\in[n]} l_i$. The first and last constraints ensure that exactly one path is chosen for each traffic demand. The second set of m inequality constraints (one for each link) and the objective function minimizes the maximum link load (W) in the network.

$$\begin{aligned}&\text{Minimize } W\\ &\text{s.t.} \sum_{j\in[l_i]} x_{i,j} = 1 \ \forall i \in [n]\\ &Ax \le \overrightarrow{W}\\ &x_{i,j} \in \{0,1\} \ \forall i,j \end{aligned} \tag{7}$$

For multiple traffic profiles, we modify the MIP. The problem of minimizing the mean of the maximum link load over the traffic profiles can be written as in Equation 8. In this formulation, there are mK inequality constraints which represent the maximum link load constraints in the network for each traffic profile. Here, $\overrightarrow{W_k}$ is a m-dimensional vector with all entries as W_k and $A_k \in [0,1]^{m\times L}$ for $k \in [K]$. We will call this formulation as the MIP in what follows.

$$\begin{aligned}&\text{Minimize } \sum_{k\in K} p_k W_k\\ &\text{s.t.} \sum_{j\in[l_i]} x_{i,j} = 1 \ \forall i \in [n]\\ &A_k x \le \overrightarrow{W_k} \ \forall k \in [K]\\ &x_{i,j} \in \{0,1\} \ \forall i,j \end{aligned} \tag{8}$$

For analyzing the randomized rounding algorithm, we give some useful large deviation bounds on sums of independent random variables. Let $X_1, X_2, ...X_n$ be random variables distributed between 0 and 1. Let $X = \sum_i X_i$ with $E[X] = \mu$. The inequalities of Equations 9 and 10 are useful for minimization and maximization problems respectively [28], [29], [30], [10]. We will use the first bound for our optimization problem.

$$Pr[X \ge \mu(1+\delta)] \le G(\mu,\delta) = \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu \ \forall \delta \ge 0 \tag{9}$$

$$Pr[X \le \mu(1-\delta)] \le H(\mu,\delta) = e^{-\mu\delta^2/2} \ \forall \delta \in [0,1] \tag{10}$$

Another very commonly used bound is the union bound (Boole's inequality) (Equation 11). Here, E_i is any set of events.

$$Pr[\bigvee_{i\in[m]} E_i] \le \sum_{i\in[m]} Pr[E_i] \tag{11}$$

We now present a lemma [9] that will be useful in the analysis of the problem.

Lemma 1.1: (a). $\forall \mu \ge 0, \forall p \in (0,1), \exists \delta = D(\mu,p)$ such that: (i) $G(\mu,\delta) \le p$, and (ii) $D(\mu,p) = \Theta\left(\frac{\log(p^{-1})}{\mu\log(\log(p^{-1})/\mu)}\right)$ if $\mu \le \log(p^{-1})/2$, and is $\Theta(\sqrt{\frac{\log(p^{-1})}{\mu}})$ otherwise.

(b). If $0 < \mu_1 \le \mu_2$, then for any $\delta > 0$, $G(\mu_1, \mu_2\delta/\mu_1) \le G(\mu_2,\delta)$.

We now come back to the MIP we defined in Equation 8. We index the inequality constraints as (profile number,link number). We can rewrite the (k,l)th inequality constraint (for kth profile and lth link) of the MIP as in Equation 12, where index i indicates a traffic demand and j indicates a path for i.

$$\sum_{i\in[n]} X^k_{l,i} \le W_k \text{ where } X^k_{l,i} = \sum_{j\in[l_i]} A^k_{l,(i,j)} x_{i,j} \tag{12}$$

We define the randomly rounded variables corresponding to each $x_{i,j}$ as $z_{i,j}$, $i \in [n], j \in [l_i]$. In the rounding scheme, for each i, there is exactly one $z_{i,j}$ equal to one with probability $x_{i,j}$, and rest are zero. This leads to $E[(A_k z)_l] = b^k_l \le y^*_k$ due to linearity of expectation. As there are maximum m paths per traffic demand and each demand lies in $[0,1]$, we assume $y^*_k \le m$. This assumption leads to a term of $min\{y^*, m\}$ in the results, so the results are valid even when this assumption does not hold. Define $c_k = y^*_k D(min\{y^*_k, m\}, 1/(\Delta mK))$ for any fixed $\Delta > 1$, and bad events $\{E^k_l : l \in [m], k \in [K]\}$ by $E^k_l \equiv$"$(A_k z)_i \ge b^k_l + c_k$". Define $\delta^k_l = c_k/b^k_l$ for each l and k. Define the r.v. $Z^k_{l,i} = \sum_{j\in[l_i]} A^k_{l,(i,j)} z_{i,j}$. Note that for each (k,l), the r.v.s $\{Z^k_{l,i} : i \in [n]\}$ lie in $[0,1]$ and are independent. Also, the events can be written as $E^k_l \equiv$"$\sum_{i\in[n]} Z^k_{l,i} \ge b^k_l + c_k$". Since $b^k_l \le y^*_k$, for each $l \in [m]$ and $k \in [K]$:

$$\begin{aligned} Pr[E^k_l] &\le G(b^k_l, \delta^k_l)\\ &\le G(y^*_k, c_k/y^*_k) = G(y^*_k, D(y^*_k, 1/(\Delta mK)))\\ &\le 1/(\Delta mK) \end{aligned} \tag{13}$$

The first inequality comes from Equation 9, the second inequality comes from part (b) of Lemma 1.1 and the last inequality comes from the definition of D. Here, $\mu =$

y_k^* and $p = 1/(\Delta m K)$. Applying the union bound on the events $E_l^k, l \in [m], k \in [K]$, we see that the probability that at least one bad event happens is less than $1/\Delta$. Thus, the probability that no bad event happens is greater than $1 - 1/\Delta$. Thus, for each traffic profile k, the resulting maximum link utilization, W_k is within $y_k^* + min\{y_k^*, m\} D(min\{y_k^*, m\}, 1/(\Delta m K))$ with probability at least $1 - 1/\Delta$, which is $y_k^* O((\log(Km)/\log(\log(Km)))$. Thus the objective value of the rounded solution is given by Equation 14.

$$\begin{aligned} W &= \sum_{k \in K} p_k W_k \\ &= \sum_{k \in K} p_k y_k^* O((\log(Km)/\log(\log(Km))) \\ &= y^* O((\log(Km)/\log(\log(Km))) \\ &= y^* O((\log(KN)/\log(\log(KN))) \end{aligned} \tag{14}$$

The last inequality follows from $m \leq N^2$. Note that optimal objective value for single-path routing cannot be less than y^* as the LP covers the ILP instance and thus its solution is always better than the optimal ILP solution. Thus, the randomized rounding algorithm produces an $O((\log(KN)/\log(\log(KN)))$-approximate solution with probability at least p, with $p = 1 - 1/\Delta$. $\square$

REFERENCES

[1] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20(4), pp. 756–767, 2002.

[2] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. on Networking*, vol. 9(3), pp. 265–279, 2001.

[3] N. G. Duffield and M. Glossglauser, "Trajectory sampling for direct traffic observation," *IEEE/ACM Trans. on Networking*, vol. 9(3), pp. 280–292, 2001.

[4] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: Router link data," *American Statistical Association Journal*, vol. 95, pp. 1063–1075, 2000.

[5] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diott, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Computer Communication Review*, pp. 161–174, 2002.

[6] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying web traffic over flow-switched networks," *IEEE/ACM Trans. on Networking*, vol. 6(6), pp. 673–685, 1998.

[7] S. Suri, M. Waldvogel, D. Bauer, and P. R. Warkhede, "Profile-based routing and traffic engineering," *Computer Communications*, vol. 24(4), pp. 351–365, 2003.

[8] M. Garey and D. Johnson, *Computers and Intractability: A guide to the theory of NP-Completeness*. Freeman and Company, 1979.

[9] A. Srinivasan, "Approximation algorithms via randomized rounding: A survey," *Lectures on Approximation and Randomized Algorithms (M. Karonski and H. J. Promel, editors), Series in Advanced Topics in Mathematics, Polish Scientific Publishers PWN*, pp. 9–71, 1999.

[10] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[11] NSFNET - The National Science Foundation Network, *http://moat.nlanr.net/INFRA/NSFNET.html*.

[12] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," *ACM SIGCOMM*, 2002.

[13] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, p. 440442, 1998.

[14] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, pp. 47–97, 2002.

[15] F. Ricciato, S. Salsano, A. Belmonte, and M. Listanti, "Off-line configuration of a MPLS over WDM network under time-varying offered traffic," *IEEE INFOCOM*, pp. 57–65, 2003.

[16] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley, "On optimal routing with multiple traffic matrices," *IEEE INFOCOM*, 2005.

[17] C. Zhang, Z. Ge, J. Kurose, Y. Liu, and D. Towsley, "Optimal routing with multiple traffic matrices tradeoff between average case and worst case performance," *UMASS CMPSCI Technical Report 05-32*, 2005.

[18] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke, "Optimal oblivious routing in polynomial time," *ACM Symposium on Theory of Computing (STOC)*, pp. 383–388, 2003.

[19] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs," *ACM SIGCOMM*, pp. 313–324, 2003.

[20] B. Grötschel, L. Lovász, and A. Shrijver, *Geometric algorithms and combinatorial optimization*. Springer-Verlag, New York, 1988.

[21] P. Raghavan and C. D. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7(4), pp. 365–374, 1987.

[22] A. Srinivasan, "An extension of the Lovász local lemma, and its applications to integer programming," *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 6–15, 1996.

[23] P. Raghavan and C. D. Thompson, "Probabilistic construction of deterministic algorithms: Approximating packing integer programs," *Journal of Computer and System Sciences*, vol. 37, pp. 130–143, 1988.

[24] T. Leighton, C.-J. Lu, S. Rao, and A. Srinivasan, "New algorithmic aspects of the local lemma with applications to routing and partitioning," *SIAM Journal on Computing*, vol. 31(2), pp. 626–641, 2001.

[25] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Application*. Prentice-Hall, 1993.

[26] CPLEX, *http://www.cplex.com*.

[27] B. Bollobás, *Modern Graph Theory*. Springer, 1993.

[28] W. Hoeffding, "Probability inequalities for sums of random variables," *American Statistical Association Journal*, vol. 58, pp. 13–30, 1963.

[29] D. Angluin and L. G. Valiant, "Fast probabilistic algorithms for hamiltonian circuits and matchings," *Journal of Computer and System Sciences*, vol. 18, pp. 155–193, 1979.

[30] N. Alon and J. H. Spencer, *The Probabilistic Method*. Wiley-Interscience Series, John Wiley and Sons, Inc., New York, 1992.