A Simple Method for Balancing Network Utilization and Quality of Routing

Yuxi Li Janelle Harms Robert Holte
Department of Computing Science, University of Alberta, Edmonton, AB, Canada T6G 2E8
{yuxi, harms, holte}@cs.ualberta.ca

***Abstract*— Applegate and Cohen [3] design demand oblivious routing schemes that achieve low oblivious ratio with no or approximate knowledge of traffic demands. We investigate the quality of oblivious routing with respect to path dispersion, which is concerned with the number of paths; and path variation, which is concerned with how far the paths are from the shortest paths and the variation of path lengths. The results show the dispersion and the variation are high in general. We propose a penalty method to improve the quality of oblivious routing. The penalty method strikes a good balance between the conflicting objectives of minimizing the oblivious ratio and optimizing the quality of oblivious routing.**

Moreover, we apply the simple penalty method to the problem of minimizing the maximum link utilization given a traffic matrix. With the penalty method, we can achieve almost the same maximum link utilization, and improve the quality of routing to almost perfect, i.e. one or two paths that are very close to the shortest paths between each pair of nodes.

I. Introduction

The goal of traffic engineering is to optimize the performance of operational networks [4]. Traffic engineering has drawn much attention in recent years. Two important components of traffic engineering are traffic estimation and routing. A good understanding of the interplay between these two inter-related components will make significant contribution to network management and performance.

A routing specifies how to route the traffic between each Origin-Destination pair across a network. The objective in designing a routing is to provide good quality of service and to optimize the utilization of network resources. Load balancing is an important approach to addressing network congestion problems resulting from inefficient resource allocation [4]. Load balancing can be achieved by minimizing the maximum resource utilization.

Measuring or estimating traffic demands accurately is non-trivial. Designing a routing robust to changing and uncertain traffic is desirable. Applegate and Cohen [3] establish a polynomial time Linear Programming (LP) model, which can achieve a low oblivious ratio with no knowledge or only approximate knowledge of traffic demands. The oblivious ratio measures how far a routing is from the optimal routing with respect to the worse-case link utilization, as discussed in Section II-A. Such a routing is referred to as a demand oblivious routing.

In designing a routing, there is usually a single objective to optimize, such as the link utilization in [3] or the network revenue in [12]. Bertsekas and Gallager indicate in their textbook [6] (page 434) that the optimization objectives of minimax link utilization and minimum average delay are essentially equivalent. Besides a major objective, like link utilization, there are other factors to consider, such as path dispersion and path variation (for the definitions, see Section III). We call these two factors the *quality of routing*. Path dispersion is concerned with the number of paths. Path variation is concerned with how far the paths are from the shortest paths and the variation of path lengths. Applegate and Cohen [3] suggest potential augmentations to their work by considering other constraints like MPLS label stack size and number of paths. A measure of the number of paths, i.e., path dispersion, was studied in [13] when the objective is a convex function; while in [3] and our study, the objective is linear. We observe that the path dispersion and path variation in oblivious routing are high in general. We propose a simple penalty method to improve the quality of oblivious routing. Experimental results show the penalty approach strikes a good balance between the conflicting objectives of minimizing oblivious ratio and optimizing the quality of oblivious routing.

We further apply the penalty method to the problem of minimizing the maximum link utilization given a traffic demand. We can achieve almost the same maximum link utilization, and improve the quality of routing to almost perfect.

We provide background in Section II. In Section III, we study the quality of oblivious routing. We describe and evaluate the penalty method in Section IV. In Section V, we apply the penalty method to the routing problem of minimizing the maximum link utilization given a traffic demand. Then we draw conclusions.

II. Background

Given a network, a routing specifies how to route the traffic between each Origin-Destination (OD) pair across the network. Open Shortest Path Protocol (OSPF), a popular Internet routing protocol, follows a destination-based evenly-distributed approach. The MultiProtocol Label Switching (MPLS) [15] architecture allows for more flexible routing. In this paper, the routes we find can be applied to MPLS.

A traffic matrix (TM) provides the amount of traffic between each OD pair over a certain time interval. Given a TM,

The first author receives Alberta Research Council (ARC) Karl A Clark memorial scholarship, honorary Izaak Walton Killam memorial scholarship and Informatics Circle of Research Excellence (iCore) graduate student scholarship. This research was partially supported by the Natural Science and Engineering Research Council of Canada (NSERC).

the optimal routing is solvable as a Linear Programming multi-commodity flow problem [2], [12]. The MPLS routing protocol can be configured with such an LP solution. A solution close to the optimal is also achievable under the OSPF routing model for a given TM by tuning link weights [9]. Unfortunately, it is non-trivial to estimate traffic demands accurately. This is still an active research area, e.g., [8], [11] and [16]. Consequently, an optimal routing is hard to obtain. The dynamic nature of network traffic aggravates the problem. The demand oblivious routing designed in [3] provides an approach to tackle the issue of designing a good routing with no or approximate knowledge of traffic demands. The oblivious routing designs a static routing that is as close as possible to the optimal under all possible traffic patterns.

A. Routing and Performance Metrics

In the following, we describe the routing and the performance metrics of the demand oblivious routing methodology [3]. We are interested in what fraction of traffic for each OD pair is routed along each link. Thus, a *routing* $f_{ij}(l)$ specifies the fraction of traffic for the OD pair $i \to j$ on link l. When the demand for the OD pair $i \to j$ is d_{ij}, the traffic on link l is $d_{ij} f_{ij}(l)$. Throughout the paper, we denote a routing as $\mathbf{f}$, a link as l, and the capacity of a link l as $cap(l)$.

In Figure 1, we present two example routings for illustration. The vector $(f_{i_1,j_1}(l),\ f_{i_2,j_2}(l))$ on each link l specifies the routing. For example, in Figure 1(a), (1, 0) on link (i_1, A) specifies that 100% of the traffic for OD pair $i_1 \to j_1$ travels link (i_1, A); while no traffic of $i_2 \to j_2$ on (i_1, A). The vector (.5, .5) on link (A, B) specifies 50% of the traffic of $i_1 \to j_1$ (as well as $i_2 \to j_2$) travels link (A, B). In Figure 1(a), there are two paths for $i_1 \to j_1$, i_1ABDj_1 and i_1ACDj_1. There is only one path for $i_1 \to j_1$ in Figure 1(b), i_1ABDj_1. Similarly, OD pair $i_2 \to j_2$ has two paths in Figure 1(a), while there is only one path in Figure 1(b).

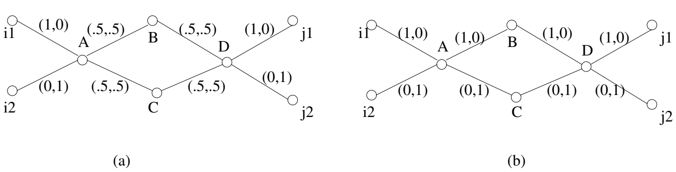

Fig. 1. An Example

A routing $\mathbf{f}$ is defined as:

$$\begin{cases} \forall i, \forall j \neq i : \sum_{e \in OUT(i)} f_{ij}(e) - \sum_{e \in IN(i)} f_{ij}(e) = 1; \\ \forall k, \forall i \neq k, \forall j \neq k, i : \\ \qquad \sum_{e \in OUT(k)} f_{ij}(e) - \sum_{e \in IN(k)} f_{ij}(e) = 0; \\ \forall \text{ edge } e, \forall i, j \neq i : f_{ij}(e) \geq 0; \end{cases} \tag{1}$$

In the above, $IN(i)$ and $OUT(i)$ denote the sets of edges "into" and "out of" node i respectively.

For a given routing $\mathbf{f}$ and a given traffic demand $\mathbf{D}$, the maximum link utilization measures the goodness of the routing, i.e., the lower the maximum link utilization, the better the routing:

$$max_l \frac{\sum_{i,j} d_{ij} f_{ij}(l)}{cap(l)}$$

Given a TM $\mathbf{D}$, the *optimal routing* minimizes the maximum link utilization:

$$\text{OPTU}(\mathbf{D}) = min_f max_l \frac{\sum_{i,j} d_{ij} f_{ij}(l)}{cap(l)}$$

For a given routing $\mathbf{f}$ and a given TM $\mathbf{D}$, the *performance ratio* is defined as the ratio of the maximum link utilization of the routing $\mathbf{f}$ on the TM $\mathbf{D}$ to the maximum link utilization of the optimal routing on the TM $\mathbf{D}$. The performance ratio measures how far routing $\mathbf{f}$ is from the optimal routing on the TM $\mathbf{D}$. Formally,

$$\text{PERF}(f, \{\mathbf{D}\}) = \frac{max_l \sum_{i,j} d_{ij} f_{ij}(l)/cap(l)}{\text{OPTU}(\mathbf{D})}$$

The performance ratio is usually greater than 1. It is equal to 1 only when the routing $\mathbf{f}$ is an optimal routing.

For a set of TMs D, the performance ratio of a routing $\mathbf{f}$ is defined as

$$\text{PERF}(\mathbf{f}, D) = max_{\mathbf{D} \in D} \text{PERF}(\mathbf{f}, \mathbf{D})$$

The performance ratio with respect to a set of traffic matrices is usually strictly greater than 1, since usually a single routing $\mathbf{f}$ can not optimize the link utilization over the set of traffic matrices.

When the set D includes all possible TMs, $\text{PERF}(\mathbf{f}, D)$ is referred to as the *oblivious performance ratio* of the routing $\mathbf{f}$. This is the worst performance ratio the routing $\mathbf{f}$ achieves with respect to all TMs. An *optimal oblivious routing* is the routing that minimizes the oblivious performance ratio. Its oblivious ratio is the *optimal oblivious ratio* of the network.

B. LP Model

Based on [14], [5], Applegate and Cohen develop LP models to solve the oblivious routing problem in polynomial time [3]. The following LP[1] can compute the oblivious ratio of a network with $O(n^2m)$ variables and $O(nm^2)$ constraints [3], where m and n are the numbers of edges and nodes in the network respectively. In the following, an edge is directed and a link is undirected. $f_{ij}(l) = \sum_{e:\text{link-of}(e)=l} f_{ij}(e)$, where l is a link, and link-of(e) is the link corresponding to edge e. After applying LP duality theory, the variables $d_{ij}(e)$ disappear and new variables $\pi(l, m)$ and $p_l(i, j)$ are introduced. Refer to [3] for details.

$$\begin{array}{l} \min r \\ f_{ij}(e) \text{ is a routing} \\ \forall \text{ links } l : \sum_m cap(m)\pi(l, m) \leq r \\ \forall \text{ links } l, \forall \text{ pairs } i \to j : \\ \quad f_{ij}(l)/cap(l) \leq p_l(i, j) \\ \forall \text{ links } l, \forall \text{ nodes } i, \forall \text{ edges } e = j \to k : \\ \quad \pi(l, link-of(e)) + p_l(i, j) - p_l(i, k) \geq 0 \\ \forall \text{links } l, m : \pi(l, m) \geq 0 \\ \forall \text{links } l, \forall \text{nodes } i : p_l(i, i) = 0 \\ \forall \text{links } l, \forall \text{nodes } i, j : p_l(i, j) \geq 0 \end{array} \tag{2}$$

[1]They develop another LP for the case when approximate knowledge of traffic demands is available. In this paper, we concentrate on the case of no knowledge of traffic demands.

III. STUDY OF THE QUALITY OF OBLIVIOUS ROUTING

In this section, we study the quality of oblivious routing found using LP (2), and motivate the penalty method to improve the quality of routing without much degradation of oblivious ratio.

The objective of LP (2) is to minimize the oblivious ratio. The lower the oblivious ratio, the closer the solution is to the optimal solution. The routing calculated by LP (2) achieves low oblivious ratios [3] with no knowledge of traffic demands. However, some factors like *path dispersion* and *path variation* are not considered. Path dispersion is concerned with how many paths exist between each OD pair as specified by the routing. Path variation is concerned with how much the paths between each OD pair differ from the shortest path of the OD pair and how much they differ from each other. It is non-trivial to manage a routing with large path dispersion and large path variation. With large path variation, it may be difficult to achieve fairness between flows. Long paths are usually not preferred. The delay jitter may be high if a flow takes multiple paths with high variation in lengths. Path dispersion and path variation are represented by two metrics: *number of paths* and *path length difference*. We count the number of paths between each OD pair of the routing computed by the LP. For each OD pair, we calculate the path length difference as the difference between the length of the shortest path by Dijkstra's algorithm [2] and the average of the lengths of all paths for the OD pair calculated by the LP. With a small length difference to the shortest paths, we expect small variations between the path lengths.

We use "realistic" Internet topologies from the Rocketfuel project [17]. We use AS1221 (Telstra, Australia), AS1755 (Ebone, Europe), AS3967 (Exodus, Europe) and AS6461 (Abovenet, US). The OSPF weights on the links (inferred weight and latency) are also provided [10]. The capacities of links are assigned according to the CISCO heuristics, that is, the link weight is inversely proportional to the link capacity. We study at the level of PoPs (Point of Presence). We use CPLEX [1] to solve the LP programs.

Figure 2 and Figure 3 present the histograms for number of paths and path length difference respectively for the studied topologies. The results show that there may be tens of or up to more than one hundred paths for an OD pair. Path length differences between the average and the shortest path vary mostly from 1 to 6 hops, or even more. The large path dispersion and path variation won't be good news for network providers.

It would be desirable to restrict the number of paths and/or the length difference of paths. To explicitly model these restrictions seems non-trivial: the LP model may become a mixed Integer Programming problem [2], which in general is hard to solve. Preliminary investigation of modeling this problem as a mixed Integer Programming problem reveals that, even on a small random network (10 nodes), the problem is hard to solve. In the next section, we propose a penalty method to construct a polynomial LP for this interesting yet challenging issue.

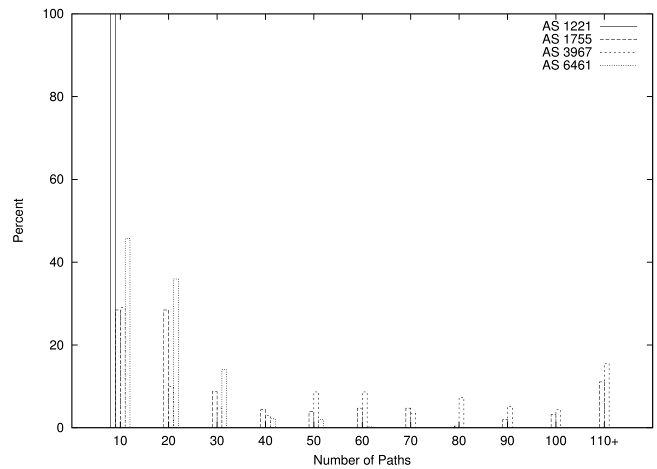

Fig. 2. Histogram for Number of Paths

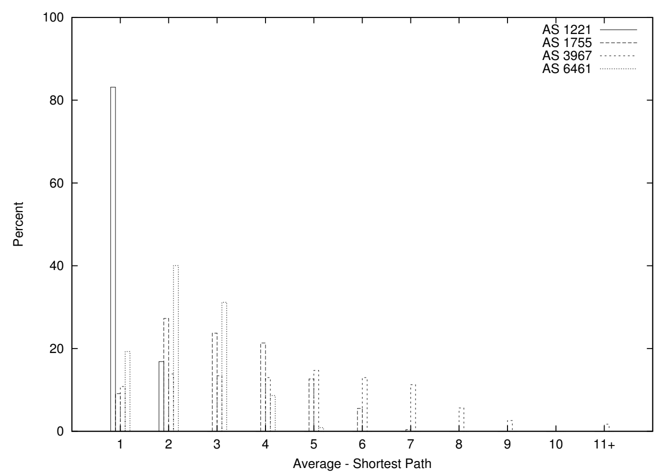

Fig. 3. Histogram for Path Length Difference

IV. A PENALTY METHOD: BALANCING NETWORK UTILIZATION AND QUALITY OF ROUTING

We propose a penalty method to balance the oblivious ratio and the quality of oblivious routing. We can still model this problem as an LP: instead of solely minimizing the oblivious ratio r, we add a penalty component t to the objective function in LP (2). That is, the objective of LP (2) becomes $r+t$. Note, r in the objective is still the oblivious ratio as in LP (2). As well, there will be new constraints for the introduction of the penalty component t. In deciding the penalty term, we attempt to choose one that is tailored to the topology. As shown in the following, the penalty term is related to LP (2) (the LP without a penalty term) and the characteristics of the topology. We also use a penalty factor to make our scheme flexible, i.e., as shown in the experiments, we can tune the penalty factor to balance the quality of routing and the oblivious ratio. We now discuss how to express the new constraints.

We favor a small number of short paths. A natural way to achieve this is to penalize using paths composed of edges "far away" from the shortest path. By intuition, with an edge far away from the shortest path, its path won't be short. We now define the distance of an edge to an OD pair, by calculating the distances of the two nodes on the edge to the shortest path of the OD pair. (Note there may not be a unique shortest path. We choose the one returned by Dijkstra's algorithm [2].)

Denote $|(u,v)|_{hop}$ as the shortest distance between two nodes u and v with respect to hop count; $sp_{weight}(u,v)$ as the shortest path between two nodes u and v with respect to link weight. We say a node $n \in sp(u,v)_{weight}$ if n is on $sp_{weight}(u,v)$. Define the distance from a node u to the OD pair $i \to j$ as:

$$dist_u(i,j) = min_{v:v \in sp_{weight}(i,j)} |(u,v)|_{hop}$$

That is, the distance from u to the OD pair $i \to j$ is the minimum hop distance from u to all the nodes on the shortest path with respect to weight for $i \to j$.

The distance of an edge $u \to v$ to an OD pair $i \to j$ is:

$$dist_{uv}(i,j) = \frac{1}{2}[dist_u(i,j) + dist_v(i,j)]$$

We define the penalty of using an edge $u \to v$ for an OD pair $i \to j$ as $dist_{uv}(i,j)$, the distance of the edge $u \to v$ to the OD pair $i \to j$:

$$penalty_{uv}(i,j) = dist_{uv}(i,j) = \frac{1}{2}[dist_u(i,j) + dist_v(i,j)]$$

The definition of the penalty function of using an edge says that using an edge far away from the shortest path of an OD pair will receive a large penalty.

The new constraint to LP (2) follows:

$$t = \frac{\beta}{\alpha} \sum_{i,j} \sum_{e} \{f_{ij}(e) penalty_e(i,j)\}$$

where β is called the *penalty factor*, and α is the penalty incurred by the optimal oblivious routing for the given topology. The oblivious routing $f^{'}_{ij}(e)$ is calculated using LP (2). Then,

$$\alpha = \sum_{i,j} \sum_{e} \{f^{'}_{ij}(e) penalty_e(i,j)\}.$$

Dividing $\sum_e \{f_{ij}(e) penalty_e(ij)\}$ by α, which can be regarded as the characterization of the topology, we make the penalty term t tailored to the topology. As a consequence, it is easier to select the penalty factor β. As shown in experiments, a single β can achieve good performance across various topologies.

From the definition of the penalty term t, we can see that if we want to minimize the objective $r + t$ as well as the penalty t, we are forced to use edges close to the shortest path. Consequently, we expect to make the paths shorter. At the same time, because we have shortened the paths, traffic is squeezed onto the paths. As a result, the penalty term also reduces the number of paths. This will be shown in experimental results.

We call the LP with the penalty component "penalty LP":

$$\begin{array}{l} \min r + t \\ f^{'}_{ij}(e) \text{ is a routing by LP (2)} \\ \quad \alpha = \sum_{ij} \sum_e \{f^{'}_{ij}(e) penalty_e(i,j)\} \\ f_{ij}(e) \text{ is a routing} \\ \sum_{ij} \sum_e \{f_{ij}(e) penalty_e(i,j)\} - \frac{\alpha}{\beta} t = 0 \\ \text{Other constraints and variables in LP (2)} \end{array} \qquad (3)$$

This LP still has $O(mn^2)$ variables and $O(nm^2)$ constraints like LP (2). That is, it is polynomial.

A. Experimental Results

We study LP (3) by varying the penalty factor β from 1 to 10 with step size 1. We present the results for *average number of paths* and *average path length difference* vs. β. The average number of paths is the total number of calculated paths divided by the total number of OD pairs. To calculate the average path length difference, we first calculate the difference between the length of the shortest path and the average of the lengths of all the paths computed by the LP for each OD pair. Then we sum the length differences and divide it by the total number of OD pairs. That is, the average path length difference is the average over all OD pairs. The results are shown in Table I for β of 0, 1, 2 and 3. The numbers in the brackets are the corresponding standard deviations. Note that $\beta = 0$ is a special case for the result calculated by LP (2) (without the penalty component). LP (2) yields the lowest oblivious ratio a network can obtain. Another extreme is generated by the OSPF routing. For reference, it has oblivious ratios according to [3]: 4.16 (AS1221), 16.60 (AS1755), 49.20 (AS3967) and 233.98 (AS6461).

Table I shows that as the penalty factor β increases, both average number of paths and average path length difference decrease. At the same time, the oblivious ratio increases. The results are expected, since LP (3) minimizes $r + \frac{\beta}{\alpha} \sum_{ij} \sum_e \{f_{ij}(e) penalty_e(ij)\}$, thus a larger β puts a larger penalty on using edges farther away from the shortest path. The good news is, it shows that the average number of paths and average path length (and their standard deviations) decrease rapidly and level off, while the oblivious ratio increases only gradually. Thus, it is possible to choose a proper β to balance the oblivious ratio and average number of paths and average path length. For example, a penalty factor of 1 gives good performance with respect to both the oblivious ratio and the quality of routing. Compared with an LP without a penalty term (β = 0), our penalty method can reduce the mean and the variance of average number of paths and average path length difference dramatically, and pay a reasonably low cost of an increase in the oblivious ratio (an increase in oblivious ratio of around 5% if β = 1).

In constructing the penalty model, we use a linear combination of $penalty_e(ij)$ with $f_{ij}(e)$ as coefficients. Other variants are possible. The penalty constraint of $t = \frac{\beta}{\alpha} \sum_{ij} \sum_e \{f_{ij}(e)[penalty_e(ij)]^2\}$ with $\alpha = \sum_{ij} \sum_e \{f^{'}_{ij}(e)[penalty_e(ij)]^2\}$ yields similar results.

V. Applying The Penalty Method to Minimize Maximum Link Utilization Given Traffic Matrix

It is expected that high path dispersion and path variation are inherent to a routing calculated by an LP with a single objective like link utilization, since there are no constraints to restrict the number of paths nor the path length. Thus our penalty method will be beneficial in balancing the link utilization and the quality of routing. In this section, we study the penalty method for the problem of minimizing the maximum link utilization given a traffic matrix. As introduced in Section II, given a TM $\mathbf{D}$, the *optimal routing*

AS	β	Avg. Number of Paths	Avg. Path Len. Δ	oblivious ratio
AS 1221	0	2.1128 (1.0958)	0.4242 (0.3452)	1.43378
	1	1.2569 (0.6177)	0.0887 (0.2147)	1.50906
	2	1.2569 (0.6177)	0.0887 (0.2147)	1.50906
	3	1.2569 (0.6177)	0.0887 (0.2147)	1.50906
AS 1755	0	37.0435 (46.8444)	2.6404 (1.3981)	1.80574
	1	9.9901 (10.1403)	1.3805 (0.7424)	1.88721
	2	7.2984 (7.6857)	1.0937 (0.7057)	1.97399
	3	7.5158 (9.6115)	1.0092 (0.7165)	2.06780
AS 3967	0	58.4156 (66.2676)	3.9538 (2.4060)	1.60053
	1	14.1970 (19.0946)	1.8752 (1.1371)	1.71021
	2	10.7294 (16.2205)	1.5555 (1.1758)	1.77533
	3	8.3874 (11.1978)	1.3527 (1.0658)	1.95732
AS 6461	0	12.5022 (9.1238)	1.6774 (0.9584)	1.92253
	1	8.3853 (6.6586)	1.2137 (0.7695)	2.02123
	2	6.5563 (5.4207)	1.0216 (0.7020)	2.18085
	3	5.3593 (5.1461)	0.8104 (0.6757)	2.37735

TABLE I

RESULTS FOR OBLIVIOUS ROUTING WITH STANDARD DEVIATION IN BRACKETS

minimizes the maximum link utilization, i.e., OPTU($\mathbf{D}$) $= min_f max_l \sum_{i,j} d_{ij} f_{ij}(l)/cap(l)$.

The LP formulation follows:

$$\begin{array}{l} \text{min } u \\ f_{ij}(e) \text{ is a routing} \\ \forall \text{ links } l : \sum_{i,j} f_{ij}(l) d_{ij}/cap(l) \leq u \end{array} \quad (4)$$

Note that in LP (4), traffic demand, d_{ij}'s, are constants, in contrast to the LPs for demand oblivious routing, e.g. LP (2).

The formulation of the LP for minimizing the maximum link utilization with penalty term t follows a similar approach as penalty LP (3). The major difference is that α is calculated using LP (4). The LP follows,

$$\begin{array}{l} \text{min } u + t \\ f'_{ij}(e) \text{ is a routing by LP (4)} \\ \quad \alpha = \sum_{ij} \sum_e \{f'_{ij}(e) penalty_e(i,j)\} \\ f_{ij}(e) \text{ is a routing} \\ \sum_{ij} \sum_e \{f_{ij}(e) penalty_e(i,j)\} - \frac{\alpha}{\beta} t = 0 \\ \forall \text{ links } l : \sum_{i,j} f_{ij}(l) d_{ij}/cap(l) \leq u \end{array} \quad (5)$$

A. Experimental Results

In the following, we study the performance of the penalty method for the problem of minimizing the maximum link utilization given a traffic matrix. We still use the measured ISP topologies from the Rocketfuel project. For traffic matrices, we use synthetic models, Gravity model and Bimodal. The Gravity model is developed in [19] as a fast and accurate estimation of traffic matrices. We use a heuristic approach similar to that in [3], in which the volume of traffic flowing into/out of a POP is proportional to the combined capacity of links connecting with the POP. Bimodal, a.k.a. elephant-mice phenomena, is studied in [7], [16]. We use a random Bimodal as in [3], [11].

The results are shown in Table II. We present the results for the penalty factor $\beta = 0$ and 0.01. Recall that $\beta = 0$ means there is no penalty term in the LP, i.e., LP (4) is used. The numbers in the brackets are the corresponding standard deviations. We can see the huge improvement in the average number of paths and the average path length differences, while (almost) the same maximum link utilization (MLU). We indicate the cases where the maximum link utilization does not increase by a star * in the table. Note that the maximum link utilization scales with the scaling of traffic matrix. Wang et al. [18] established that TCP Active Queue Management and IP can achieve the optimal performance using a single-path routing under certain conditions. Our penalty method may be an alternative approach to obtain such a single-path routing, maybe by redesigning the penalty term. It is interesting to further study this issue. We do not show the results for AS1221, since without a penalty term, the quality of routing for the given TMs is good.

We present an example to explain why with much fewer paths, a routing can achieve a maximum link utilization very close to the optimal. We use Figure 1, where we present two routings. Recall the vector $(f_{i_1,j_1}(l), f_{i_2,j_2}(l))$ on each link l specifies the routing. We assume the traffic demands between the OD pairs $i_1 \to j_1$ and $i_2 \to j_2$ are equal. As well, the capacities on all the links are the same. The two routings in Figure 1(a) and Figure 1(b) achieve the same maximum link utilization. However, in routing (a), each OD pair uses two paths; while in routing (b), each pair uses only one path.

B. Discussion

For the two sets of problems, to achieve good performance, we need to choose different penalty factors. An approach to selecting the penalty factor is to choose an initial penalty factor, e.g. set $\beta = 1$, and then adjust β to achieve a satisfactory objective value (either oblivious ratio or minimal maximum utilization). Starting from $\beta = 1$, for LP (3), we can adjust β by adding 1 each time ($\beta = 1$ or 2 is a good choice). For LP (5), we can adjust β by dividing 10 each time ($\beta =$ 0.01 is a good choice). It will be easy to choose β by observing the LP solutions. This also justifies the way we determine the penalty term, $t = \frac{\beta}{\alpha} \sum_{ij} \sum_e \{f_{ij}(e) penalty_e(ij)\}$, where β is an adjustable parameter, α and the rest of the expression are characterized by the topology.

AS	TM	β	Avg. Number of Paths	Avg. Path Len. Δ	MLU
AS 1755	Gravity	0	499.3083 (496.1111)	5.8101 (2.6204)	0.12277
		0.01	1.1008 (0.3443)	0.0461 (0.1511)	0.12277 *
	Bimodal	0	551.2213 (482.0415)	6.4338 (1.8370)	0.09554
		0.01	1.1008 (0.3443)	0.0461 (0.1511)	0.09929
AS 3967	Gravity	0	286.6753 (349.9402)	5.3179 (2.9386)	0.09163
		0.01	2.3074 (2.3083)	0.4857 (0.6699)	0.09163 *
	Bimodal	0	305.5368 (349.4327)	6.0176 (2.3408)	0.11483
		0.01	2.5173 (2.4090)	0.5514 (0.6766)	0.11535
AS 6461	Gravity	0	786.4892 (1018.8641)	4.4442 (3.1886)	0.12417
		0.01	1.1991 (0.5848)	0.0738 (0.1923)	0.12417 *
	Bimodal	0	1977.3961 (1492.9976)	6.4955 (2.1852)	0.12005
		0.01	1.5087 (0.9049)	0.2334 (0.3804)	0.12005 *

TABLE II

RESULTS FOR MINIMIZING MAXIMUM LINK UTILIZATION WITH GIVEN TM WITH STANDARD DEVIATION

VI. CONCLUSION

We study the quality of oblivious routing with no knowledge of traffic demands based on the path dispersion and the path variation. We observe that the oblivious routing has a large number of paths and the paths may be much longer than the shortest paths. We propose a penalty method to balance the oblivious ratio and the quality of oblivious routing. The penalty method works well: it can achieve a low oblivious ratio as well as high quality of routing with respect to the number of paths and the path lengths. The penalty method strikes a good balance between the conflicting objectives of minimizing the oblivious ratio and optimizing the quality of oblivious routing.

Furthermore, we apply the penalty method to the problem of minimizing the maximum link utilization given a traffic matrix. With the penalty method, we can achieve almost the same maximum link utilization, and improve the quality of routing to almost perfect, i.e. one or two paths that are very close to the shortest paths between each pair of nodes.

REFERENCES

[1] *CPLEX: Mathematical programming solver.* http://www.cplex.com.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall Inc., 1993.

[3] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of SIGCOMM'03*, pages 313–324. Karlsruhe, Germany, August 2003.

[4] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. *IETF RFC 2702*, September 1999.

[5] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke. Optimal oblivious routing in polynomial time. In *Proceedings of 35th STOC*, pages 383–388. San Diego, June 2003.

[6] D. P. Bertsekas and R. G. Gallager. *Data Networks (2nd Edition)*. Printice Hall, 1992.

[7] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. Geographical and temporal characteristics of Inter-POP flows: View from a single POP. *European Transactions on Telecommunications*, 13(1):5–22, February 2002.

[8] 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 Transactions on Networking*, 3(9):265–279, June 2001.

[9] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of INFOCOM'00*, pages 519–528. Tel-Aviv, Israel, March 2000.

[10] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *Proceedings of IMW'02*, pages 231–236. Marseille, France, November 2002.

[11] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: existing techniques and new directions. In *Proceedings of SIGCOMM'03*, pages 161–174. Karlsruhe, Germany, August 2003.

[12] D. Mitra and K. G. Ramakrishna. A case study of multiservice, multipriority traffic engineering design for data networks. In *Proceedings of Globecom'99*, pages 1077–1083. Rio de Janeiro, Brazil, December 1999.

[13] A. Ouorou, P. Mahey, and J.-P. Vial. A survey of algorithms for convex multicommodity flow problems. *Management Science*, 46(1):126–147, January 2000.

[14] H. Räcke. Minimizing congestion in general networks. In *Proceedings of 43th IEEE FOCS*, pages 43–52. Vancouver, Canada, November 2002.

[15] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. *IETF RFC 3031, Standards Track*, January 2001.

[16] A. Soule, A. Nucci, E. Leonardi, R. Cruz, and N. Taft. How to identify and estimate top largest OD flows in a dynamic environment. In *Proceedings of SIGMETRICS'04*, pages 49–60. New York, June 2004.

[17] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of SIGCOMM'02*, pages 133–146. Pittsburgh, August 2002.

[18] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking*, 13(3):568–582, June 2005.

[19] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of SIGMETRICS'03*, pages 206–217. San Diego, June 2003.