Multipath Oblivious Routing for Traffic Engineering – Stable and Robust Routing in Changing and Uncertain Environments

Yuxi Li, Baochun Bai, Janelle Harms, Robert Holte
Department of Computing Science, University of Alberta

Abstract— **Intra-domain traffic engineering is essential for the operation of an Internet Service Provider (ISP). Demand-oblivious routing [5] promises excellent performance guarantee with changing and uncertain traffic demands. However, it is non-trivial to implement it. We investigate an efficient and deployable implementation of oblivious routing. We study its performance by both numerical experiments and simulation. The performance study shows that the multipath implementation achieves a close approximation to oblivious routing [5], especially when approximate knowledge of traffic is available. The study shows its robustness under varying traffic demands, link failures and an adversary attack. Its performance is excellent even with a 100% error in traffic estimation. We open the door for a deployable demand-oblivious routing, which can provide robust network services with good quality to network users.**

I. Introduction

Intra-domain traffic engineering is essential for the operation of an Internet Service Provider (ISP). It is desirable to design a routing protocol that can balance network utilization, mitigate the impact of failures and attacks, and thus provide good quality of service to network users, with economic provisioning of network resources. However, it is challenging to design such a routing protocol due to traffic changes and uncertainty. Network traffic is inherently changing and uncertain, due to factors such as the diurnal pattern, dynamic inter-domain routing, link failures, and attacks. Adaptive traffic resulting from overlay routing or multihoming [2], [13], [25] further aggravates the problems.

There are three classes of solutions: link weight optimization [10], [11], [34], [35], traffic-adaptive approaches [8], [14], [29] and demand-oblivious routing [4], [5], [33]. Link weight optimization guarantees performance only for a limited set of traffic demands, while how to find critical traffic matrices is investigated in [36]. An adaptive approach is responsive to traffic changes, so that the issues of stability and convergence [15] have to be addressed both in theory and in practice. Demand-oblivious routing is particularly promising; it promises excellent performance guarantee with changing and uncertain traffic demands. Its performance is particularly good with approximate knowledge of traffic demands, which is made available by the recent great progress in traffic estimation, e.g. [7], [9], [16], [20], [31], [37], [38]. In [33], the performance is optimized for expected scenarios and is guaranteed for unexpected scenarios.

However, it is non-trivial to implement oblivious routing in [5]. A straightforward implementation is for each node to forward incoming packets according to the routing fractions computed by [5]. However, without careful attention, such a distributed implementation may lead to loops. Furthermore, an oblivious routing may involve a large number of paths between each origin-destination (OD) pair, which requires a large number of labels in an MPLS deployment [28]. This is shown in our previous study [17]. It is thus desirable to route traffic on a small number of paths. However, since there are many paths between each OD pair, it may be difficult to select a small set of paths that gives good performance.

The challenges to implement oblivious routing are shared by many other optimization-based routing strategies. Recent progress along this line of research, including those being linear [1], [21], convex [23] or with game-theoretic concerns [3], has greatly advanced the state-of-the-art of routing. However, to achieve the optimal solution, such optimization-based approaches typically use an arc-formulation (see §III-A) as oblivious routing [5]. Thus it may encounter the implementation issues as discussed above.

We investigate an efficient and deployable implementation of oblivious routing. We design MORE, Multipath Oblivious Routing for traffic Engineering, to obtain a close approximation to [5]. MORE can achieve very excellent performance guarantee when combined with approximate knowledge of traffic demands. However, it does not need frequent collection of network information like an adaptive approach. An oblivious routing guarantees the performance for much broader traffic demands than those specified by the traffic knowledge, since its performance is invariant with the scaling of traffic demands, thus temporary traffic spikes may be covered. Oblivious routing optimizes a worst case performance metric. However, our empirical study will show that MORE achieves a performance close to the optimal. Its performance is excellent even with a 100% error in traffic estimation. In addition, as a quasi-static solution, MORE can be static on an hourly, multi-hourly or even daily basis. Thus, MORE is much less concerned with stability and convergence issues [15] than an adaptive approach, which is responsive on a small time-scale, like seconds. MORE does not need changes to core routers, thus it can be efficiently implemented and gradually deployed.

Contribution. To the best of our knowledge, we are the first to investigate a feasible implementation of demand-oblivious routing [5]. We design MORE, a multipath approximation to [5]. Through extensive numerical experiments and simulation, we show the excellent performance of MORE under varying TMs, link failures and an adversary attack. Our work is complementary to [4], [5] and [33].

We open the door for a viable deployment of oblivious routing, thus providing an intra-domain traffic engineering technique robust to changing and uncertain environments. MORE is a promising option for traffic engineering, along with link weight optimization [10], [11], [34], [35] and adaptive approaches, like MATE [8], TeXCP [14] and [29].

Paper organization. We present related work in §II and preliminaries in §III. Then we present the design of multi-path oblivious routing in §IV, and evaluate its performance in §V. After discussing implementation and deployment issues in §VI, we draw conclusions.

II. Related work

Our work is built on the achievements of a large body of previous work in traffic engineering.

With accurate knowledge of the traffic matrix (TM), an optimal routing for a flexible architecture like MPLS is achievable by solving a multi-commodity linear programming (LP) flow problem [21]. For OSPF/IS-IS, Fortz and Thorup [10], [11] deploy a local search technique to find a set of link weights for shortest path computation, which gives good performance for a given TM or a set of TMs. This is compatible with OSPF/IS-IS. However, it may not guarantee good performance for some traffic demands. Zhang and Ge investigate how to find critical traffic matrices [36]. Zhang et al. [34], [35] investigate routing optimization over multiple TMs and the tradeoff between average- and worst-case performance.

The research on oblivious routing [5], [6], [26] has made great achievements. The oblivious routing problem is to design a routing that achieves close to the optimal performance, with no or only approximate knowledge of the traffic demand, without considering the current network load. Räcke [26] investigates oblivious routing on general symmetric networks. Azar et al. [6] show that an optimal oblivious routing can be computed by an LP with a polynomial number of variables, but an infinite number of constraints. Applegate and Cohen [5] design a simple polynomial size LP to obtain demand-oblivious routing schemes that achieve good performance. Applegate et al. [4] study demand oblivious restoration strategies. Wang et al. [33] design a scheme so that the performance is optimized for expected scenarios and is guaranteed for unexpected scenarios. In [18], we design oblivious routing for energy efficiency in wireless networks. This work is the first to investigate a feasible implementation of oblivious routing.

Gallager's work [12] is a classic in adaptive routing. Recently, Kandula et al. [14] propose an adaptive routing on multiple paths. Shaikh et al. [29] and MATE [8] are also adaptive approaches. MORE, as a quasi-static solution, operates in a static way on a large time-scale, thus the issues of stability and convergence are much less severe than for adaptive approaches.

Traffic estimation has made great progress, e.g. Feldmann et al. [9], Medina et al. [20], Bhattacharyya et al. [7], Zhang et al. [37], [38], Lakhina et al. [16] and Soule et al. [31]. The network community begins to enjoy deeper understanding of the traffic structure and more accurate demand estimation. Various models are proposed to study the spatial and temporal structure of the traffic. Techniques for fairly accurate traffic estimation are available, e.g., the Gravity model [37]. See Soule et al. [31] for a recent survey.

III. Preliminaries

In this section, we introduce preliminaries about routing and the competitive analysis framework [5].

A traffic matrix specifies the amount of traffic between each OD pair over a certain time interval. An entry d_{ij} denotes the amount of traffic for OD pair $i \to j$. The capacity of edge e is denoted as $c(e)$.

A. Routing

A routing specifies how to route the traffic between each OD pair across a given network. Open Shortest Path First Protocol (OSPF) and Intermediate System to Intermediate System (IS-IS), two popular Internet routing protocols, follow a destination-based evenly-split approach. The MultiProtocol Label Switching (MPLS) architecture allows for more flexible routing. Both OSPF/IS-IS and MPLS can take advantage of path diversity. OSPF/IS-IS distributes traffic evenly on multiple paths with equal cost. MPLS may support arbitrary routing fractions over multiple paths. Our work is applicable to MPLS, which is widely deployed by ISPs.

We differentiate arc- and path-routing (§IV-B). An *arc-routing* $f_{ij}(e)$ specifies the fraction of traffic demand d_{ij} on edge e [5], [6]. An arc-routing is not readily implementable for either OSPF or MPLS. We use $in(k)$ and $out(k)$ to denote edges "entering" or "leaving" node k respectively. Arc-routing $\mathbf{f}$ is defined as:

$$\begin{cases} \forall \text{ pairs } i \to j : \sum\limits_{e \in out(i)} f_{ij}(e) - \sum\limits_{e \in in(i)} f_{ij}(e) = 1 \\ \forall \text{ pairs } i \to j, \forall \text{ nodes } k \neq i,j : \\ \quad \sum\limits_{e \in out(k)} f_{ij}(e) - \sum\limits_{e \in in(k)} f_{ij}(e) = 0 \\ \forall \text{ pairs } i \to j, \forall \text{ edges } e : f_{ij}(e) \geq 0 \end{cases} \tag{1}$$

B. Link utilization

For a given arc-routing $\mathbf{f}$ and a given traffic demand $\mathbf{tm}$, the maximum link utilization (MLU) measures the goodness of the routing, i.e., the lower the maximum link utilization, the better the routing:

$$\text{MLU}_{\text{arc}}(\mathbf{tm}, \mathbf{f}) = \max_{e \in E} \sum_{i,j} d_{ij} f_{ij}(e)/c(e) \tag{2}$$

Given a TM $\mathbf{tm}$, an *optimal arc-routing* minimizes the maximum link utilization:

$$\text{OPTU}_{\text{arc}}(\mathbf{tm}) = \min_{f} \max_{e \in E} \sum_{i,j} d_{ij} f_{ij}(e)/c(e) \tag{3}$$

C. Competitive Analysis

The routing computed by (3) does not guarantee performance for other traffic matrices. Applegate and Cohen [5] developed LP models to compute an optimal routing that minimizes the oblivious ratio with a weak assumption on the traffic demand. We present the metric of performance ratio that follows the competitive analysis [5], [24].

For a given routing **f** and a given traffic matrix **tm**, the *performance ratio* is defined as the ratio of the maximum link utilization of the routing **f** on the traffic matrix **tm** to the maximum link utilization of the optimal routing for **tm**. The performance ratio measures how far routing **f** is from the optimal routing for traffic matrix **tm**.

$$\text{PERF}(\mathbf{f}, \{\mathbf{tm}\}) = \frac{\text{MLU}(\mathbf{tm}, \mathbf{f})}{\text{OPTU}_{\text{arc}}(\mathbf{tm})} \quad (4)$$

This applies to both an arc- and a path-routing, thus we do not add a subscript to MLU. The performance ratio is usually greater than 1. It is equal to 1 only when the routing **f** is an optimal routing for **tm**.

When we are considering a set of traffic matrices **TM**, the performance ratio of a routing **f** is defined as

$$\text{PERF}(\mathbf{f}, \mathbf{TM}) = \max_{\mathbf{tm} \in \mathbf{TM}} \text{PERF}(\mathbf{f}, \{\mathbf{tm}\}) \quad (5)$$

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

When the set **TM** includes all possible traffic matrices, PERF(**f**, **TM**) is referred to as the *oblivious performance ratio* of the routing **f**. This is the worst performance ratio the routing **f** achieves with respect to all traffic matrices. An *optimal oblivious routing* is the routing that minimizes the oblivious performance ratio. Its oblivious ratio is the *optimal oblivious ratio* of the network.

Suppose there is an oracle that knows the instant traffic matrix **tm** and computes its optimal routing with link utilization u. The link utilization of the optimal oblivious routing for **tm** is guaranteed to be within $[u, r * u]$, where r is the oblivious ratio. It may achieve lower link utilization than $r * u$ for the particular traffic matrix **tm**. The oblivious routing guarantees the performance of what an oracle can achieve multiplied by the oblivious ratio for all traffic matrices.

Table III presents an example to illustrate the performance metric of competitive ratio. Suppose all other TMs have CR(**f**, {**tm**}) less than 1.5. Then, $\max_{\mathbf{tm} \in \mathbf{TM}} \text{CR}(\mathbf{f}, \{\mathbf{tm}\})$, or oblivious ratio, is 1.5. Table III also shows that for some TM, the performance ratio can be lower than 1.5, e.g., for TM_3, the ratio is 1.2. This shows that the oblivious routing **f** can performance better than the oblivious ratio predicts.

	TM_1	TM_2	TM_3	...	TM_∞
MLU(**tm**, {**f**})	0.7	0.6	0.6		
OPTU(**tm**)	0.6	0.4	0.5		
CR(**f**, {**tm**})	1.1	1.5	1.2		
$\max_{\mathbf{tm} \in \mathbf{TM}}$ CR(**f**, {**tm**})			**1.5**		

TABLE I

EXAMPLE: OBLIVIOUS RATIO

IV. MORE: MULTIPATH OBLIVIOUS ROUTING FOR TRAFFIC ENGINEERING

A. Overview

As discussed in the Introduction, there are obstacles to the implementation of oblivious routing in [5], such as potential routing loops and a large number of MPLS labels. We investigate a deployable oblivious routing, MORE, Multipath Oblivious Routing for traffic Engineering.

We use a quasi-static routing, so that the fractions of traffic on the multiple paths between an OD pair do not change over a large time period, in contrast to an adaptive routing. The routing fractions may have to change, e.g., after severe network failures have happened. Such an implementation has the nice feature that issues like stability and convergence are much less severe than for adaptive approaches. As well, MORE alleviates the reliance on global network information: it can achieve excellent performance with a large time-scale traffic estimation, but it does not need to collect the instantaneous link load. The oblivious ratio can be computed by the reformulation of the oblivious routing on K paths in LP (15), which gives the worst case performance guarantee.

Figure 1 gives an illustration of MORE. Between the OD pair, there are three paths, with routing fractions 0.5, 0.2 and 0.3, computed by LP (14) or LP (15). The incoming traffic will be forwarded on the three paths according to their routing fractions, i.e., 50%, 20% and 30% respectively.

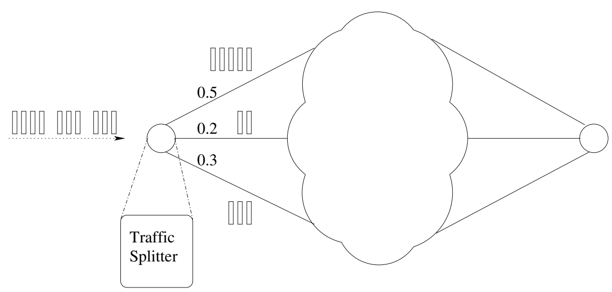

Fig. 1. Illustration of MORE

B. Multipath routing

Each OD pair $i \to j$ is configured with up to K_{ij} paths. For notational brevity, we use K paths for each OD pair. The set of paths for OD pair $i \to j$ is denoted as $P_{ij} = \{P_{ij}^1, ..., P_{ij}^K\}$. A multipath routing computes, for each OD pair $i \to j$, a routing fraction vector, defined as

$$< f_{ij}^1, ..., f_{ij}^K >, \sum_k f_{ij}^k = 1, f_{ij}^k \geq 0 \quad (6)$$

on the set of paths for OD pair $i \to j$. A *path-routing* f_{ij}^k specifies the fraction of traffic demand d_{ij} on path P_{ij}^k. A path-routing is readily implementable for MPLS.

Given path-routing $\mathbf{f}$ and traffic demand $\mathbf{tm}$, the maximum link utilization is:

$$\text{MLU}_{\text{path}}(\mathbf{tm}, \mathbf{f}) = \max_{l \in E} \sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \quad (7)$$

Here $\delta_{ij}^k(l)$ is an indicator function, which is 1 if $l \in P_{ij}^k$, 0 otherwise. We use $l \in P_{ij}^k$ to denote edge l is on path P_{ij}^k.

Given $\mathbf{tm}$, an *optimal path-routing* that minimizes the maximum link utilization is:

$$\text{OPTU}_{\text{path}}(\mathbf{tm}) = \min_{\mathbf{f}} \max_{l \in E} \sum_{ij} d_{ij} \sum_{k} \delta_{ij}^k(l) f_{ij}^k / c(l) \quad (8)$$

C. LP Formulation

We give LP models for multipath oblivious routing. We replace the arc formulation in Applegate and Cohen [5] with a path formulation to compute an optimal oblivious routing and its ratio. In an arc formulation, routing variables are on links and flow conservation constraints are at each node for each OD pair. In a path formulation, routing variables are on paths and flow conservation constraints are implicitly satisfied on each path. We start with the case in which there is approximate knowledge of traffic demand.

Similar to Applegate and Cohen [5], the optimal oblivious routing can be obtained by solving an LP with a polynomial number of variables, but infinitely many constraints. With the approximate knowledge that d_{ij} is in the range of $[a_{ij}, b_{ij}]$, we have the "master LP":

$$\begin{aligned} &\min r \\ &\mathbf{f} \text{ is a path-routing} \\ &\forall \text{ edges } l, \forall \alpha > 0 \\ &\quad \forall \text{ TMs } \mathbf{tm} \text{ with } \text{OPTU}_{\text{arc}}(\mathbf{tm}) = \alpha, a_{ij} \le d_{ij} \le b_{ij}: \\ &\quad\quad \sum_{ij} d_{ij} \sum_{k} \delta_{ij}^k(l) f_{ij}^k / c(l) \le \alpha r \end{aligned} \quad (9)$$

The oblivious ratio is invariant with the scaling of the traffic matrices or the scaling of the edge capacity. Thus, when computing the oblivious ratio, it is sufficient to consider traffic matrices with $\text{OPTU}_{\text{arc}}(\mathbf{tm}) = 1$. Another benefit of using traffic matrices with $\text{OPTU}_{\text{arc}}(\mathbf{tm}) = 1$ is that the objective of the LP, the oblivious ratio r, is equal to the maximum link utilization of the oblivious routing.

Since the oblivious ratio r is invariant with respect to the scaling of TMs, we can consider a scaled TM $\mathbf{tm}' = \alpha \cdot \mathbf{tm}$. With $\alpha = 1/\text{OPTU}_{\text{arc}}(\mathbf{tm})$, we have $\text{OPTU}_{\text{arc}}(\mathbf{tm}') = 1$. Under these conditions, the master LP (9) becomes:

$$\begin{aligned} &\min r \\ &\mathbf{f} \text{ is a path-routing} \\ &\forall \text{ edges } l: \\ &\quad \forall \text{ TMs } \mathbf{tm} \text{ with } \text{OPTU}_{\text{arc}}(\mathbf{tm}) = 1, \\ &\quad \lambda > 0, \lambda a_{ij} \le d_{ij} \le \lambda b_{ij}: \\ &\quad\quad \sum_{ij} d_{ij} \sum_{k} \delta_{ij}^k(l) f_{ij}^k / c(l) \le r \end{aligned} \quad (10)$$

For the condition "$\forall$ TMs $\mathbf{tm}$ with $\text{OPTU}_{\text{arc}}(\mathbf{tm}) = 1$", we need the flow definition on edges. Flow $\mathbf{g}$ is defined as,

$$\begin{cases} \forall \text{ pairs } i \to j, k \ne i, j: \sum_{e \in out(k)} g_{ij}(e) - \sum_{e \in in(k)} g_{ij}(e) = 0 \\ \forall \text{ pairs } i \to j: \sum_{e \in out(j)} g_{ij}(e) - \sum_{e \in in(j)} g_{ij}(e) + d_{ij} = 0 \\ \forall \text{ pairs } i \to j, \forall \text{ edges } e: g_{ij}(e) \ge 0 \\ \forall \text{ pairs } i \to j: d_{ij} \ge 0 \end{cases} \quad (11)$$

LP formulations can be simplified by collapsing flows g_{ij} on an edge e with the same origin by $g_i(e) = \sum_j g_{ij}(e)$.

Given a path-routing $\mathbf{f}$, the constraint of the master LP (10) can be checked by solving the following "slave LP" for each edge l to examine whether the objective is $\le r$ or not. In (12), routing f_{ij}^k are constant and flow $g_{ij}(e)$, demand d_{ij} and λ are variables.

$$\begin{aligned} &\max \sum_{ij} d_{ij} \sum_{k} \delta_{ij}^k(l) f_{ij}^k / c(l) \\ &\forall \text{ pairs } i \to j: && \Leftarrow w_l(i,j) \\ &\quad \sum_{e \in out(j)} g_i(e) - \sum_{e \in in(j)} g_i(e) + d_{ij} \le 0 \\ &\forall \text{ edges } e: \sum_i g_i(e) \le c(e) && \Leftarrow \pi_l(e) \\ &\forall \text{ pairs } i \to j: d_{ij} - \lambda b_{ij} \le 0 && \Leftarrow \kappa_l^+(i,j) \\ &\forall \text{ pairs } i \to j: -d_{ij} + \lambda a_{ij} \le 0 && \Leftarrow \kappa_l^-(i,j) \\ &\forall \text{ pairs } i \to j: d_{ij} \ge 0, g_{ij}^k \ge 0 \\ &\lambda > 0 \end{aligned} \quad (12)$$

The flow conservation constraint is relaxed from equality to ≤ 0, which allows for OD pair $i \to j$ to deliver more flow than demanded, and does not affect the maximum link utilization of 1. The constraints of LP (12) guarantee the traffic can be routed with maximum link utilization of 1.

The dual of LP (12) is LP (13). To help make the derivation of the dual LP (13) clearer, we use leftarrow $\Leftarrow$ to indicate dual variables corresponding with primal constraints in LP (12). In dual LP (13), we indicate primal variables corresponding to dual constraints.

$$\begin{aligned} &\min \sum_e c(e) \pi_l(e) \\ &\forall \text{ pairs } i \to j: && \Leftarrow d_{ij} \\ &\quad w_l(i,j) + \kappa_l^+(i,j) - \kappa_l^-(i,j) \ge \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \\ &\forall \text{ nodes } i, \forall \text{ edges } (u,v): && \Leftarrow g_i(u,v) \\ &\quad \pi_l(u,v) + w_l(i,u) - w_l(i,v) \ge 0 \\ &\sum_{i,j} \{a_{ij} \kappa_l^-(i,j) - b_{ij} \kappa_l^+(i,j)\} \ge 0 && \Leftarrow \lambda \\ &\forall \text{ edges } e: \pi_l(e) \ge 0 \\ &\forall \text{ pairs } i \to j: w_l(i,j) \ge 0, \kappa_l^+(i,j) \ge 0, \kappa_l^-(i,j) \ge 0 \\ &\forall \text{ nodes } i: w_l(i,i) = 0, \kappa_l^+(i,i) = 0, \kappa_l^-(i,i) = 0 \end{aligned} \quad (13)$$

According to the LP duality theory [1], the primal LP and its dual LP have the same optimal value if they exist. That is, LP (12) and LP (13) are equivalent. Because LP (13) is a minimization problem, we can use its objective in place of the objective of LP (12) in the "$\le r$" constraints of LP (10). Replacing the constraint in the master LP (10) with LP (13), we obtain a single LP to compute the oblivious performance ratio using K paths.

$$
\begin{array}{l}
\min\ r \\
\mathbf{f} \text{ is a path-routing} \\
\forall \text{ edges } l: \\
\quad \sum_e c(e)\pi_l(e) \le r \\
\quad \forall \text{ pairs } i \to j: \\
\quad\quad w_l(i,j) + \kappa_l^+(i,j) - \kappa_l^-(i,j) \ge \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \\
\quad \forall \text{ nodes } i, \forall \text{ edges } (u,v): \\
\quad\quad \pi_l(u,v) + w_l(i,u) - w_l(i,v) \ge 0 \\
\quad \sum_{i,j}\{a_{ij}\kappa_l^-(i,j) - b_{ij}\kappa_l^+(i,j)\} \ge 0 \\
\quad \forall \text{ edges } e: \pi_l(e) \ge 0 \\
\quad \forall \text{ pairs } i \to j: w_l(i,j) \ge 0, \kappa_l^+(i,j) \ge 0, \kappa_l^-(i,j) \ge 0 \\
\quad \forall \text{ nodes } i: w_l(i,i) = 0, \kappa_l^+(i,i) = 0, \kappa_l^-(i,i) = 0
\end{array} \tag{14}
$$

When there is no knowledge of the traffic demand, i.e., the range $[a_{ij}, b_{ij}]$ for d_{ij} becomes $[0, \infty)$, the LP to compute the oblivious routing is obtained by removing the variables $\kappa_l^+(i,j)$ and $\kappa_l^-(i,j)$, as in LP (15).

$$
\begin{array}{l}
\min\ r \\
\mathbf{f} \text{ is a path-routing} \\
\forall \text{ edges } l: \\
\quad \sum_e c(e)\pi_l(e) \le r \\
\quad \forall \text{ pairs } i \to j: w_l(i,j) \ge \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \\
\quad \forall \text{ nodes } i, \forall \text{ edges } (u,v): \\
\quad\quad \pi_l(u,v) + w_l(i,u) - w_l(i,v) \ge 0 \\
\quad \forall \text{ edges } e: \pi_l(e) \ge 0 \\
\quad \forall \text{ pairs } i \to j: w_l(i,j) \ge 0 \\
\quad \forall \text{ nodes } i: w_l(i,i) = 0
\end{array} \tag{15}
$$

The derivation of LP formulation is based on the "master-slave" approach in Applegate and Cohen [5]. We use multiple paths for each OD pair.

D. MultiPath Selection

In this section, we discuss how to select multiple paths for each OD pair. The objective is to select multiple paths that give a low oblivious ratio. We investigate three approaches, namely, sp$K$, mix$K$ and focus$K$.

In sp$K$, we select K shortest paths with respect to hop count for each OD pair.

In mix$K$, we first find K shortest paths with respect to hop count, as in sp$K$. These shortest paths serve as base paths. Then, we sort the K paths in increasing order of their hop counts. After that, for each shortest path, we search for its edge-disjoint paths and record them, until K paths are found. Long paths are not preferred, so that we only search for disjoint paths that are not M hop longer than the base paths ($M = 3$). We use the name "mix$K$" to reflect that it is a mixture of shortest paths and their disjoint paths. We find K shortest paths first, in case none of them has an eligible disjoint path. In this case, the K shortest paths are chosen as the mix$K$ paths.

The method focus$K$ is based on our previous work [17]. The oblivious routing in Applegate and Cohen [5] considers the objective of lowering oblivious ratio, but not the number of paths and path lengths. In [17], we design a method to implicitly reduce the the number of paths and path lengths, with only negligible increase of the oblivious ratio. The basic idea is to put a penalty on using an edge far away from the shortest path for an OD pair. Thus, this method essentially focuses on short paths for each OD pair. We make an extension to [17] by considering range restrictions on traffic demand.

LP (16) is from Applegate and Cohen [5]. It computes the oblivious routing and its ratio of a topology, when knowledge of traffic demands is given in the range restriction format, i.e., $\forall$ pairs $i, j, 0 \le a_{ij} \le d_{ij} \le b_{ij}$.

$$
\begin{array}{l}
\min\ r \\
f_{ij}(e) \text{ is an arc-routing} \\
\forall \text{ links } l: \sum_m cap(m)\pi(l,m) \le r \\
\forall \text{ links } l, \forall \text{ pairs } i \to j: \\
\quad p_l(i,j) + s^+(i,j) - s^-(i,j) \ge f_{ij}(l)/cap(l) \\
\forall \text{ links } l, \forall \text{ nodes } i, \forall \text{ edges } e = j \to k: \\
\quad \pi(l, \text{link-of}(e)) + p_l(i,j) - p_l(i,k) \ge 0 \\
\forall \text{ links } l: \sum_{ij}(b_{ij}s^+(i,j) - a_{ij}s^-(i,j)) \le 0 \\
\forall \text{links } l, m: \pi(l,m) \ge 0 \\
\forall \text{links } l, \forall \text{nodes } i: p_l(i,i) = 0 \\
\forall \text{links } l, \forall \text{nodes } i,j: p_l(i,j), s^+(i,j), s^-(i,j) \ge 0
\end{array} \tag{16}
$$

After computing the arc-routing $\mathbf{f}$ using LP (16), similar to [17], we obtain the following "penalty LP":

$$
\begin{array}{l}
\min\ r + t \\
f'_{ij}(e) \text{ is an arc-routing by LP (16)} \\
\quad \alpha = \sum_{ij}\sum_e\{f'_{ij}(e) penalty_e(i,j)\} \\
f_{ij}(e) \text{ is an arc-routing} \\
\sum_{ij}\sum_e\{f_{ij}(e) penalty_e(i,j)\} - \frac{\alpha}{\beta} t = 0 \\
\text{Other constraints and variables in LP (16)}
\end{array} \tag{17}
$$

Here β is the penalty factor and $penalty_{uv}(i,j)$ measures the distance from edge (u,v) to OD pair $i \to j$. The larger β, the more pressure deterring use of an edge far away from the shortest path. The penalty $penalty_{uv}(i,j)$ is half the sum of the distances of nodes u and v to the shortest path of i to j. Similar to [17], when computing the shortest path of OD pair $i \to j$, we use the metric of link weight; when computing the shortest distance from a node to an OD pair $i \to j$, we use the metric of hop count. We use the CISCO heuristic of setting link weight inversely proportional to the link capacity (referred to as InvCap).

After computing the modified oblivious routing using LP (17), we extract K paths. In the performance study, we extract up to 20 shortest paths from the resultant oblivious routing with routing fractions ≥ 0.001.

LP (16) and LP (17) can handle the case in which no knowledge of traffic demands is available, by removing the constraints about $a_{ij} \le d_{ij} \le b_{ij} \ge 0$, i.e., by removing variables $s^+(i,j)$ and $s^-(i,j)$.

The path selection methods are complementary to the work using multipath routing, like TeXCP [14]. The path selection methods we discuss here are not exhaustive - we expect to see better designs.

V. Performance Study

We evaluate the performance of MORE by numerical experiments and simulation. We use the oblivious ratio of a routing and the maximum link utilization (MLU) a routing incurs as performance metrics. We solve LPs with CPLEX.[1]

A. Data

Topology. ISP topologies and traffic demands are regarded as proprietary information. The Rocketfuel project [32] deployed new techniques to measure ISP topologies and made them publicly available. Table II shows, for the topologies from Rocketfuel, the AS name and number, as well as the number of PoPs and links. The OSPF weights on the links are also provided [19]. The capacities of links are assigned according to the CISCO heuristics as in [5], i.e., the link weight is inversely proportional to the link capacity. POP 12 is the tier-1 ISP topology in Nucci et al. [22], with the scaled link capacity provided in [22]. We also use random topologies generated by GT-ITM.[2]

Topology	ID	PoPs #	links #
Ebone (Europe)	AS 1755	23	38
Exodus (Europe)	AS 3967	22	37
Abovenet (US)	AS 6461	22	42
Sprint (Europe)	POP 12	12	17

TABLE II

Tier-1 Topologies: AS 1755, AS 3967 and AS 6461 (Rocketfuel [32]) and POP 12 ([22]).

Gravity TM. Similar to [5], [14], we use the Gravity model [37] to determine the estimated traffic matrices. The Gravity model is developed in [37] as a fast and accurate estimation of traffic matrices, in which, the traffic demand between an OD pair is proportional to the product of the traffic flowing into/out of the origin/the destination. We use a heuristic approach similar to that in [5], in which the volume of traffic flowing into/out of a POP is proportional to the combined capacity of links connecting with the POP. Then we extrapolate a complete Gravity TM.

Lognormal TM. We also use the log-normal model in Nucci et al. [22] to generate synthetic TMs. In the first step, we generate traffic entries using a log-normal distribution. Then these entries are associated with OD pairs according to a heuristic approach similar to that recommended in [22]. That is, OD pairs are ordered by the first metric of their fan-out capacities. The fan-out capacity of a node is the sum of the capacities of links incident with it. The fan-out capacity of an OD pair is the minimum of the fan-out capacities of the two nodes. Ties are broken by the second metric of connectivity, defined as the number of links incident to a node. Similarly, the minimum is taken for the two nodes.

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

[2]http://www.cc.gatech.edu/projects/gtitm/

Similar to [5], in the experiments, when approximate knowledge is available, we consider a base TM, with the entry d_{ij} for OD pair $i \rightarrow j$, and an error margin $w > 1$, so that the traffic for $i \rightarrow j$ is in the range of $[d_{ij}/w, w * d_{ij}]$.

B. MultiPath selection

First, we study the performance of the path selection methods, namely, sp$K$, mix$K$ and focus$K$. The benchmark is the method in Applegate and Cohen [5], which can achieve the lowest oblivious ratio for a given topology. Hereafter, we refer to the method in Applegate and Cohen [5] as **AC**. Recall that it is non-trivial to implement the routing computed by AC as we discuss in Introduction. Thus a close multipath approximation to AC is desirable.

In Figure 2, we show the performance of the various path selection methods, when approximate knowledge of the TM is available, with a Gravity base TM and $w = 2.0$. For AS 1755, all path selection methods have good performance when the error margin is small, with sp20 jumping up when error margin increases and mix20 maintaining the best performance. For AS3967 and AS6461, focus20 has overall good performance. For POP12, sp$K$ and mix$K$, for $K = 10, 20$, have similar results, with performance very close to AC. As well, comparing with the results in Table III, the performance of focus20 for AS 3967 and mix20 for AS 1755 are much better when there is approximate knowledge of traffic demand.

We compare our path selection methods with the link weight optimization [10] (referred to as WtOpt) and InvCap.[3] For WtOpt, we search the set of link weights for 5 synthetic Lognormal TMs [22] which have optimal MLU=0.3. We also search link weights for a Gravity TM for WtOpt. The results are not as good as those shown here.

Table III shows the oblivious ratios for various path selection methods, as well as the ratios for WtOpt and the ratios computed by AC. As expected, there is a gap between the oblivious ratios computed by AC and those by the multipath approximation, namely, sp20, mix20 and focus20. With more paths, e.g., 50 paths, the gap can become narrower. However, a small number of paths may be desirable, thus we do not present results for 30 or 50 paths. For POP 12, multipath approximation approaches sp20 and mix20 can achieve the same oblivious ratio as AC. The results also show that WtOpt [10] has large oblivious ratios. WtOpt can consider multiple TMs. However, it is non-trivial for WtOpt to optimize for all or a continuous set of TMs.[4]

We also study the performance of InvCap, WtOpt, focus20 and mix20 on 100 random TMs. We study the MLU an routing incurs compared with an optimal, denoted as MLU/OPT. When there is no knowledge of the TM, an entry is uniformly set on $[10, 100]$; when the error margin $w = 2.0$, we first

[3]WtOpt and InvCap are methods to set link weights, with which, a shortest path algorithm can determine a routing. We also use WtOpt and InvCap to refer to the routings computed by the link weights they find respectively.

[4]WtOpt and InvCap are compatible with OSPF, while MORE is generally not. Here we study which routing can achieve better performance w.r.t. oblivious ratio, without considering its forwarding scheme.

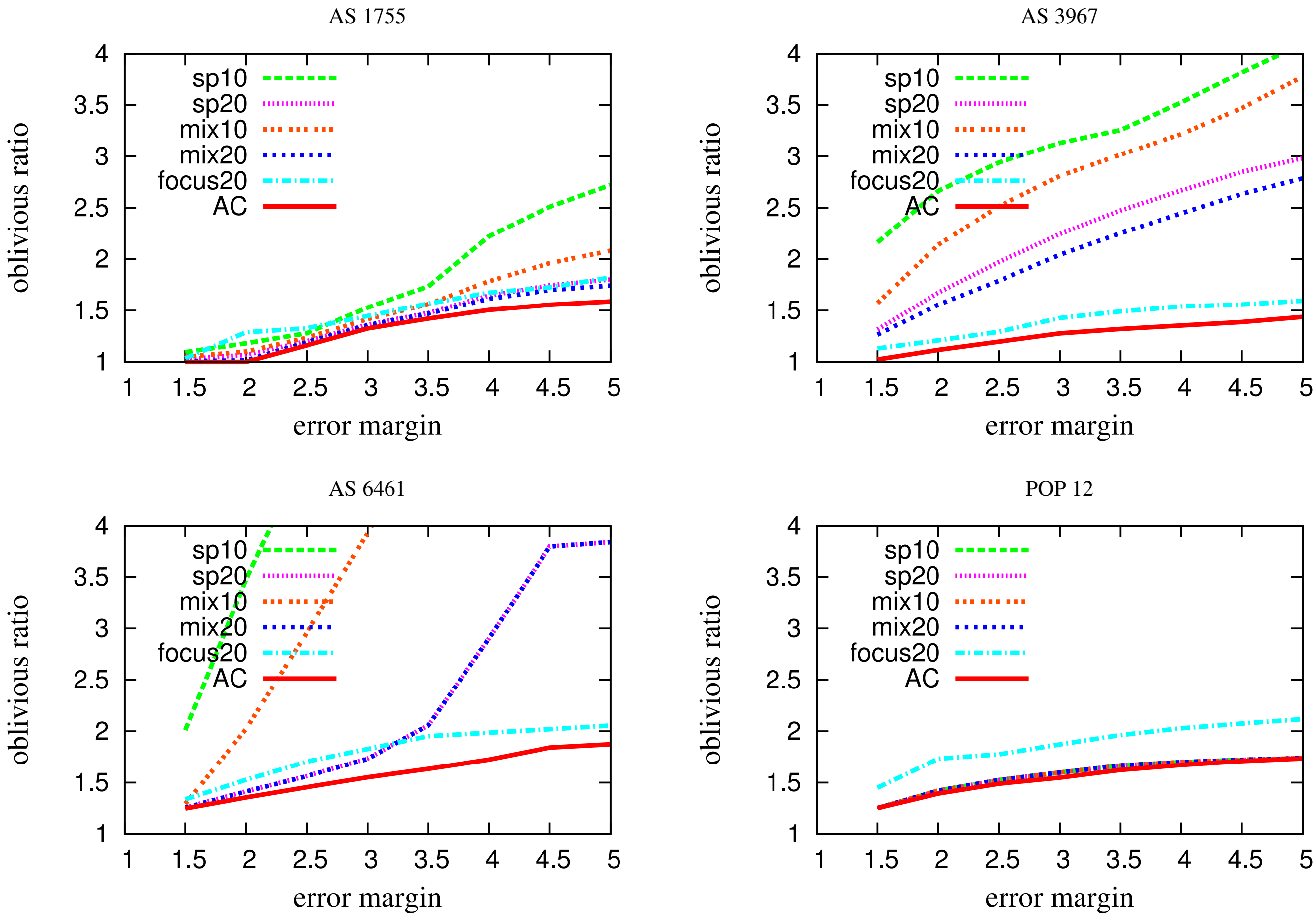

Fig. 2. Oblivious ratio vs. error margin for various path selection methods

Topology	WtOpt	sp20	mix20	focus20	AC
AS 1755	27.840	3.306	2.718	1.950	1.781
AS 3967	17.613	3.442	5.061	3.428	1.623
AS 6461	28.889	4.250	4.250	2.107	1.910
POP 12	3.813	1.785	1.785	2.161	1.785

TABLE III

OBLIVIOUS RATIOS FOR VARIOUS ROUTINGS METHODS: WTOPT [10], SPK, MIXK, FOCUSK AND AC [5], WHEN NO KNOWLEDGE OF TRAFFIC DEMANDS IS AVAILABLE.

decide d_{ij} by the Gravity model, then generate 100 random TMs uniformly on $[d_{ij}/w, d_{ij} * w]$. We show the average MLU/OPT and the 95% confidence interval of each routing method in Figure 3. The results show that MORE can achieve good performance. When there is no knowledge of the TM, focus20 has good performance for AS1755, AS3967 and AS6461; while mix20 has good performance for POP 12. When the error margin $w = 2.0$, both focus20 and mix20 have low MLU/OPT. InvCap and WtOpt may have good performance in some cases, however, they may incur high MLU/OPT. It is expected that the heuristic approach InvCap may not have good performance for some TMs.

For lack of ISP topologies, we use random topologies to attempt to justify the performance of the path selection methods. We use GT-ITM to generate 100 random topologies with 25 PoPs. Link capacities are uniformly chosen on $[10, 100]$. In Figure 4, we show the performance of focus20 and mix20 on random topologies, comparing with AC. We order the topologies in increasing order of their oblivious ratios using AC. We observe that on the studied random topologies, both focus20 and mix20 have good performance: they can achieve oblivious ratios close to that achieved by AC; while mix20 performs particularly well, by tracking closely the curve of AC, especially in the case where there is approximate knowledge of traffic demands (w = 2.0).

We may not be able to make a conclusive judgment on the performance of the proposed path selection methods based on a sample of ISP topologies and 100 random topologies. However, we gain high confidence that a multipath oblivious routing can have a close approximation to AC.

In MORE, we can choose paths and compute an optimal oblivious routing before conducting further traffic engineering tasks. That is, we can choose the best path selection method for a network. In later studies, we use path selection methods according to Table IV. When there is approximate knowledge

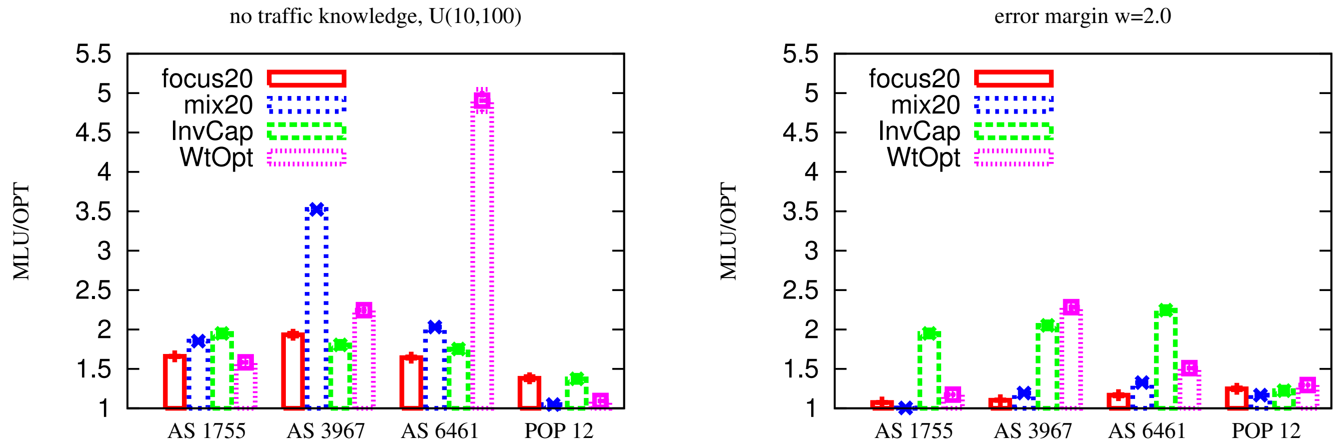

Fig. 3. MLU/OPT of various routing methods: InvCap, WtOpt, focus20 and mix20. Average MLU/OPT and 95% confidence interval for 100 random TMs 1) without any knowledge of TM, uniformly chosen on [10,100] (left), and 2) with error margin 2.0 (right).

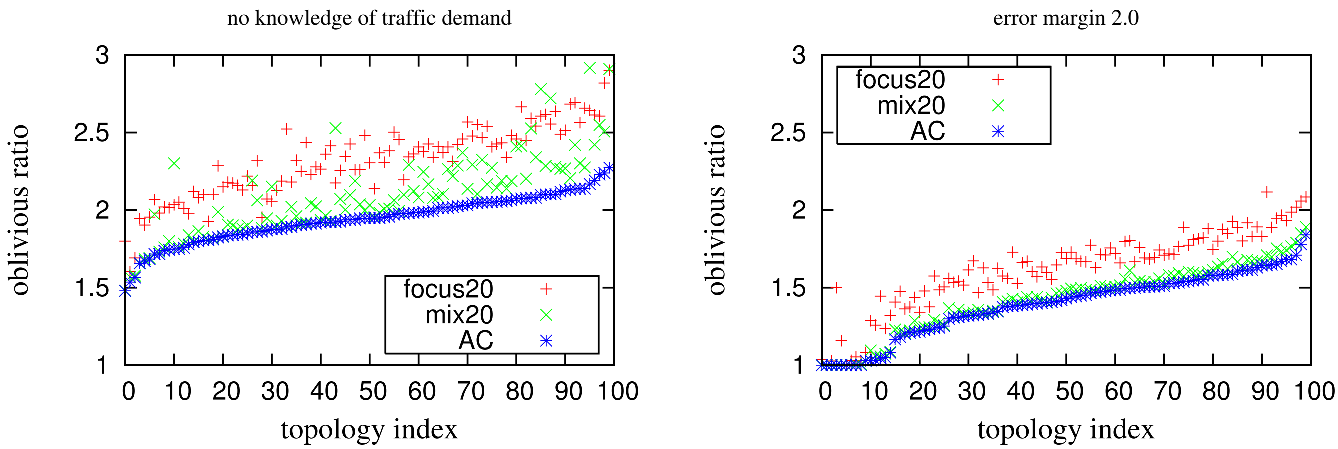

Fig. 4. Performance of path selection methods on 100 random topologies, comparing with AC.

of traffic demands, we use error margin $w = 2.0$, which can be interpreted as a tolerance of 100% error in traffic estimation.[5] The current traffic estimation techniques can achieve an error much finer than 100%, e.g., Zhang et al. [37] and Soule et al. [31]. Table IV also shows the oblivious ratios for $w = 2.0$ for the path selection methods.

Topology	AS 1755	AS 3967	AS 6461	POP 12
Path selection	mix20	focus20	focus20	mix10
obliv. ratio (w = 2.0)	1.068	1.156	1.513	1.422

TABLE IV
PATH SELECTION METHODS IN EXPERIMENTS

C. Link failure

Applegate et al. [4] study failure restoration for arc-based oblivious routing [5]. We study failure restoration for MORE.

We investigate three restoration strategies: *nochange*, *reoptimization* and *augmentation*. In nochange, the routing keeps unchanged (if the failure does not cause a path to break). To evaluate this, we use LP (12) to compute the oblivious ratio. In another extreme, reoptimization, we reoptimize multipath oblivious routing for the new topology after link failures occur, using LP (14) or LP (15). An approach in between, augmentation, is to reoptimize only for the affected OD pairs, which use the link(s) with failure. The LP derivation for augmentation is similar to that for LP (14) and LP (15), and the resultant LPs are similar, except that the routing variables for the unaffected OD pairs are constant. We show the LP formulation for augmentation with the approximate knowledge that d_{ij} is in the range of $[a_{ij}, b_{ij}]$ in LP (18).

[5] An optimal oblivious routing for the range $[a_{ij}, b_{ij}]$ gurantees the performance not only for TMs in the range, but also those scaled TMs.

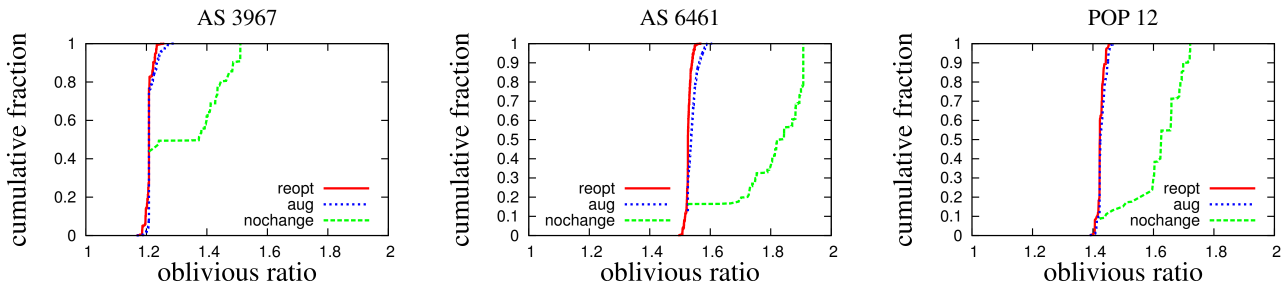

Fig. 5. Cumulative distribution of the oblivious ratios, when one- or two-link failure (20% capacity reduction) happens.

$$
\begin{array}{l}
\min\ r \\
\mathbf{f} \text{ is a path-routing defined on affected OD pairs} \\
\forall \text{ edges } l: \\
\quad \sum_e c(e)\pi_l(e) \le r \\
\quad \forall \text{ pairs } i \to j: \\
\quad\quad w_l(i,j) + \kappa_l^+(i,j) - \kappa_l^-(i,j) \ge \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \\
\quad\quad (f_{ij}^k \text{ is constant if OD pair } i \to j \text{ is not affected}) \\
\quad \forall \text{ nodes } i, \forall \text{ edges } (u,v): \\
\quad\quad \pi_l(u,v) + w_l(i,u) - w_l(i,v) \ge 0 \\
\quad \sum_{i,j}\{a_{ij}\kappa_l^-(i,j) - b_{ij}\kappa_l^+(i,j)\} \ge 0 \\
\quad \forall \text{ edges } e: \pi_l(e) \ge 0 \\
\quad \forall \text{ pairs } i \to j: w_l(i,j) \ge 0, \kappa_l^+(i,j) \ge 0, \kappa_l^-(i,j) \ge 0 \\
\quad \forall \text{ nodes } i: w_l(i,i) = 0, \kappa_l^+(i,i) = 0, \kappa_l^-(i,i) = 0
\end{array}
\tag{18}
$$

Usually a failure restoration approach like reoptimization and augmentation causes routing disruptions. An arc-routing may introduce a high degree of routing disruptions for failure restoration. MORE may restrict the routing disruptions, since it is a multipath approach. Moreover, simulation results in §V-E demonstrate that MORE is robust to link failures and routing changes. Thus, for MORE, reoptimization and augmentation may provide affordable performance with respect to potential routing disruptions.

Since a link in a PoP-level topology may represent many physical links, we attempt to study reasonable scenarios of 20% capacity reduction of PoP links. We study all the cases in which one or two links lose 20% of the capacity, with a Gravity base TM and $w = 2.0$. Then we compute the cumulative distribution of the oblivious ratios of all these failure cases. For example, in Figure 5, for AS 3967, more than 60% of the failure cases result in an oblivious ratio smaller than 1.40

From Figure 5, we observe that augmentation (optimize only for the affected OD pairs) has a similar performance to reoptimization. However, a more careful checking of the results shows that there are indeed some portion of OD pairs that are not affected by the failures. Even for nochange with the routing, we see that for a large fraction of failure cases, the 20% capacity reduction may not increase the oblivious ratio significantly. The results for AS 1755 are similar.

The analysis on link failure can also be used to improve the network provisioning: for example, for those links whose 20% capacity reduction cause large oblivous ratio increases, adding 25% of the link capacity will maintain a low oblivious ratio when a 20% capacity reduction failure happens.

Simulation results in §V-E show the robustness of MORE over link failures. It is desirable to further study more severe failures, e.g., whole link or even node failures as studied in [4].

D. Adversary attack

We introduce an attack which can exploit a routing $\mathbf{f}$, by generating a TM for $\mathbf{f}$ to incur a high MLU. We will show that an oblivious routing is robust to such an attack. However, an adaptive routing may suffer much higher MLU.

We illustrate how such an attack works. For a given arc-routing $\mathbf{f}$,[6] LP (19) computes the traffic demand $\mathbf{d}$ that gives the maximum link utilization on edge l, assuming there is a range restriction on traffic demand $0 \le a_{ij} \le d_{ij} \le b_{ij}$. Then the adversary demand is the demand that gives the largest MLU over all edges. LP (19) is based on the work in [5]. We call LP (19) "adversary LP". To obtain the LP formulation when there is no knowledge of the traffic, i.e., no range restriction, we remove the constraints $\forall$ demands $i \to j : d_{ij} - \lambda b_{ij} \le 0, -d_{ij} + \lambda a_{ij} \le 0$ and $\lambda \ge 0$.

$$
\begin{array}{l}
\max \sum_{ij} d_{ij} f_{ij}(l)/cap(l) \\
\forall \text{ pairs } i \to j: \sum_{e \in out(j)} g_i(e) - \sum_{e \in in(j)} g_i(e) + d_{ij} \le 0 \\
\forall \text{ edges } e: \sum_i g_i(e) \le cap(m) \\
\forall \text{ demands } i \to j: d_{ij} - \lambda b_{ij} \le 0 \\
\forall \text{ demands } i \to j: -d_{ij} + \lambda a_{ij} \le 0 \\
\forall \text{ nodes } i,\ \text{edges } e: g_i(m) \ge 0, \lambda \ge 0
\end{array}
\tag{19}
$$

We compare MORE with an adaptive arc-routing, denoted as *adaptive-arc*, which computes an optimal arc-routing for a given TM. The experiments runs in iterations. In iteration 1, adaptive-arc computes an optimal routing $\mathbf{f}_1$ for demand $\mathbf{TM}_0$. Before iteration 2, the attacker computes the adversary demand $\mathbf{TM}_1$ for routing $\mathbf{f}_1$. In iteration 2, routing $\mathbf{f}_1$ is used for the demand $\mathbf{TM}_1$, thus it incurs high MLU/OPT, the MLU compared with an optimal. In Iteration 3, adaptive-arc computes an optimal routing again and then the adversary

[6]For a path-routing, convert it to an arc-routing first. Alternatively, LP (12) can be used to compute an adversary.

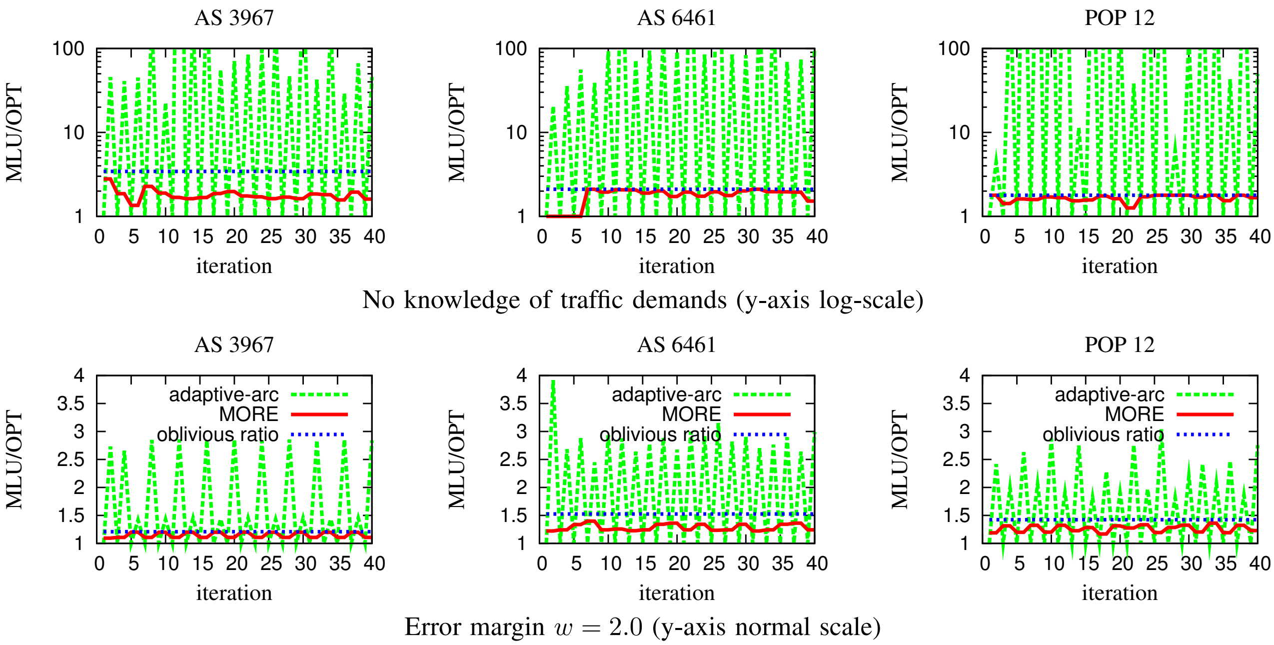

Fig. 6. Comparing MORE with adaptive-arc under adversary attack. **Legend**: in bottom figures.

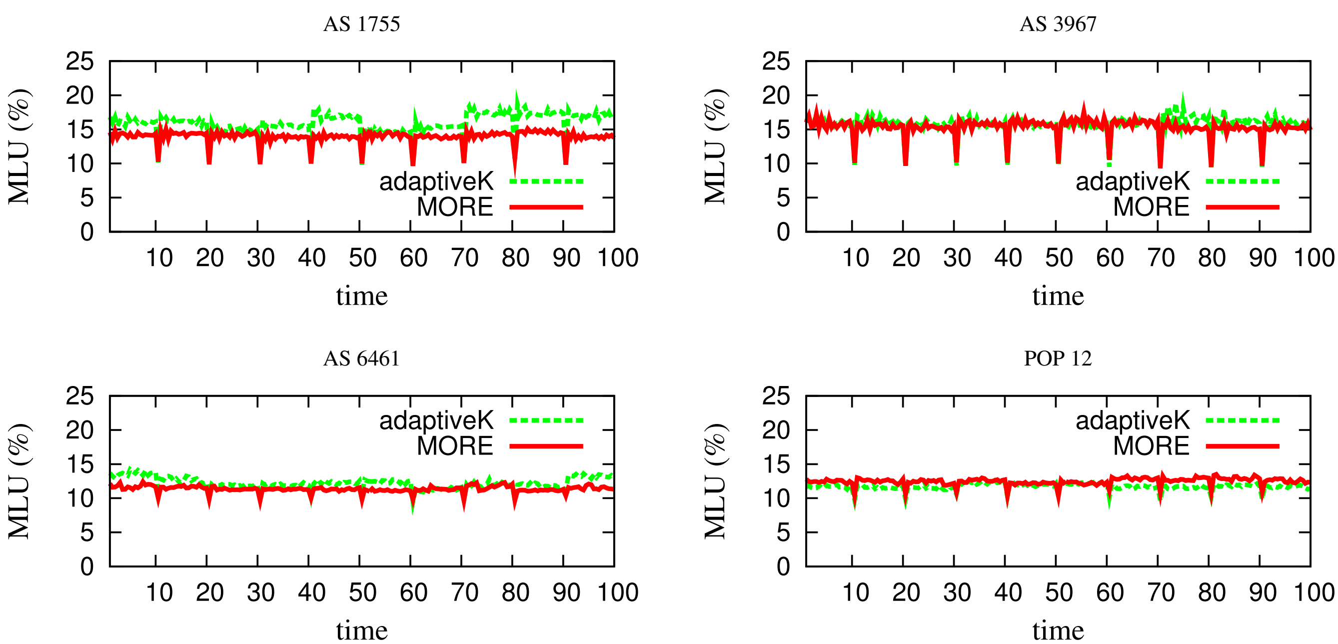

Fig. 7. Robustness of MORE over varying TMs and routings. For each 10 seconds, a random TM is generated, and MORE responds with an optimal multipath oblivious routing over the same set of paths. For adaptive$K$, it computes, for each TM, an optimal routing on K-shortest paths ($K = 20$).

attacker computes the adversary demand for the new routing. And so on. MORE keeps the routing unchanged.

In Figure 6, we show how MORE and adaptive-arc perform when there is an adversary attack. The top row shows the results when there is no restriction on the demand, i.e., $w = \infty$. Adaptive-arc may have a very large MLU/OPT. We truncate the y-axis at 100.0, and use log scale. The bottom row shows the results when the error margin $w = 2.0$. We also show the ratios of corresponding oblivious routings, as the horizontal straight lines.

Figure 6 shows that the adversary attack can exploit an adaptive routing, and make MLU/OPT prohibitively high. However, MORE is robust to the attack. The oblivious ratio predicts its worst performance. An oblivious routing can be viewed as an optimal equilibrium point in a game in which a network operator combats with all possible adversary traffic demands (within the range restriction when it is stipulated). This shows the robustness of MORE against the adversary attack, and the potential vulnerability of an adaptive routing. The results also show that MORE can perform better than what the oblivious ratio predicts, i.e., the curve of MORE is sometimes below the straight line for the ratio. The results for AS 1755 are similar. For w=2.0, the ratio is 1.068, thus the curve for MORE is very close to x-axis.

E. Simulation

We analyze the performance of LP models for MORE in previous sections. In this section, we study the performance of MORE using packet-level simulation with NS2[7]. We implement the robust weighted hashing by Ross [27], so that traffic can be split into multiple paths according to the routing fraction of each path. We use either the Gravity or the Lognormal model to generate synthetic TMs. Then, with the synthetic TMs, we generate Pareto traffic to obtain variability in the actual traffic. Note that although a TM may not change, traffic varies due to the Pareto distribution. For every 0.5 second, we average the link utilization and take the maximum to obtain the maximum link utilization (MLU).

Robust under varying TMs and routings. MORE is a quasi-static solution, it may have to change the routing when necessary. We attempt to study the robustness of MORE over changing TMs and routings by simulation. We generate 10 Lognormal TMs [22]. Each TM lasts 10 seconds. MORE computes an optimal multipath oblivious routing for a given TM with error margin $w = 2.0$. Thus there are potentially different routings for different TMs. Adaptive$K$ computes an optimal routing with K-shortest paths for each TM, with $K = 20$. We assume both MORE and adaptive$K$ know a new TM and reoptimize the routing for it instantaneously. Adaptive$K$ represents an adaptive scheme on K-shortest paths that can respond to traffic changes without any delay, i.e., it is an unachievable best case for adaptive schemes.

Results are shown in Figure 7. [8] We scale the TMs, so that optimal arc-routings of these TMs have the same MLU. The results show that MORE incurs similar MLUs over varying TMs and routings. We also observe that MORE achieves similar performance as adaptive$K$.

TeXCP vs. MORE. We compare MORE with TeXCP, an adaptive multipath routing approach [14]. TeXCP collects network load information and adjusts routing fractions on pre-selected multiple paths for each OD pair to balance the network load. TeXCP also uses MLU as the performance metric. For comparison with TeXCP, we set link capacity in a way similar to [14], i.e., links with high-degree nodes have large capacity and links with low-degree nodes have small capacity. We use the setting for TeXCP as suggested in [14]. Traffic is generated according to a Gravity TM. During time intervals $[25, 50]$ and $[75, 100]$, an extra TM is activated, so that extra traffic is generated for each OD pair. Figure 8 shows the comparison results. We show the results after 10 seconds, so that TeXCP may have passed the "warm-up" phase. We see both TeXCP and MORE respond to the traffic increases. The results show that MORE has a comparable performance to TeXCP. When TeXCP is in the transition of adapting to its optimal routing, MORE may have better performance, e.g. in the time interval $[25, 50]$ for POP 12. However, TeXCP may adapt to a better routing than MORE, e.g., in the time interval $[75, 100]$ for AS 3967. MORE, being oblivious to traffic changes, saves resources consumed by TeXCP for frequently collecting network information. With a longer time period (35 seconds) for the "warm-up", TeXCP has similar performance.

Link failure. We study the robustness of MORE over link failures using simulation. At each 10's second, a random link failure occurs with 20% link capacity reduction. After each link failure, the augmentation strategy with $w = 2.0$ (§V-C) for failure restoration is used to optimize the oblivious routing for the affected paths. The TM keeps unchanged, generated according to a Gravity TM. Figure 9 shows the results. We observe that the networks have rather stable performance, after several consecutive link failures. Reoptimization has similar performance.

Adversary attack. We study the performance of MORE and adaptive$K$ under an adversary attack.[9] Adaptive$K$ computes an optimal routing on K-shortest paths ($K = 20$) for a given TM. An adversary attack can exploit an adaptive routing for the last TM, by generating a new TM. Refer to §V-D for details of how an adversary attack works. MORE does not change paths and routing fractions.

The simulation runs in iteration, each with 20 seconds. For the first 10 seconds, adaptive$K$ encounters an adversary attack; while for the second 10 seconds, it uses the optimal routing for the adversary in the last 10 seconds. We assume adaptive$K$ can know the exact TM, and deploys the new optimal routing instantaneously in the middle point of an iteration. The oblivious routing does not change over the whole run of the simulation.

[7] http://www.isi.edu/nsnam/ns/

[8] For Figure 7 and 10, there are downward spikes for both adaptive$K$ and MORE. These are due to the transition of stopping and starting TMs.

[9] Adaptive$K$ responds to traffic changes instantaneously, while TeXCP takes time for convergence, thus we do not compare MORE with TeXCP here.

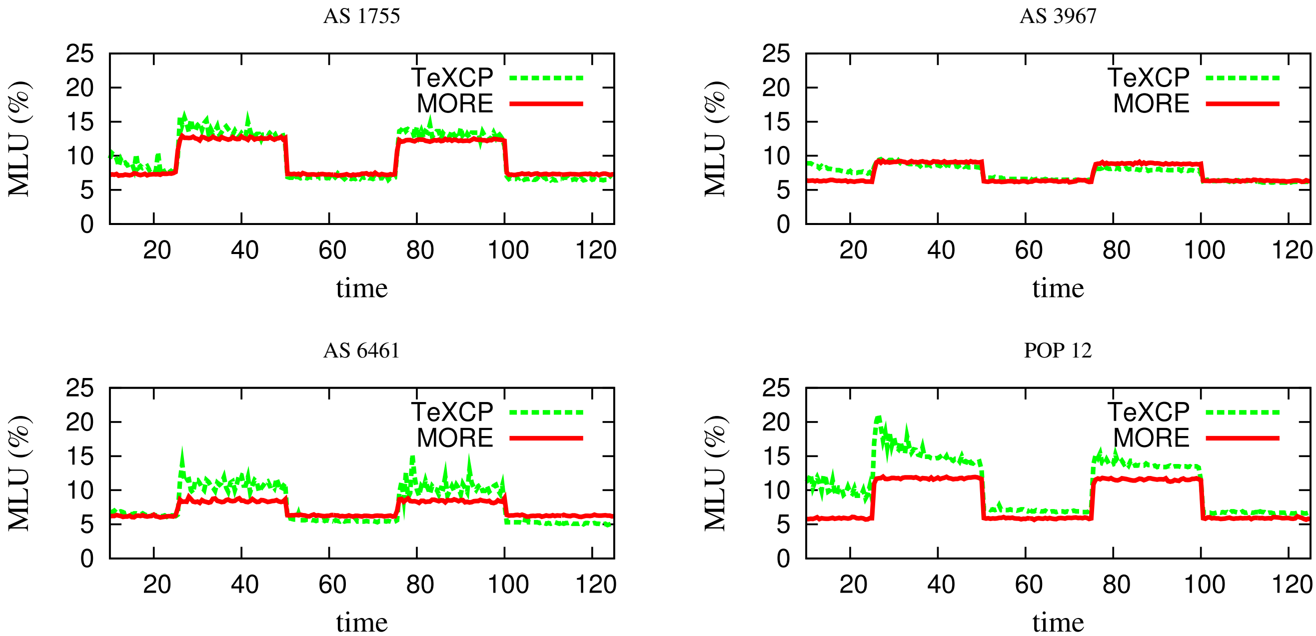

Fig. 8. TeXCP vs. MORE. During time interval [25,50] and [75,100], extra random traffic is generated.

The results are shown in Figure 10. We observe that when adaptive$K$ is under the adversary attack, it has much larger MLU than MORE. However, when adaptive$K$ operates in optimal, its performance is comparable to or slightly better than that of MORE. The results show that, MORE is robust under an adversary attack, and it has a performance close to adaptive$K$ when adaptive$K$ is not under attack.

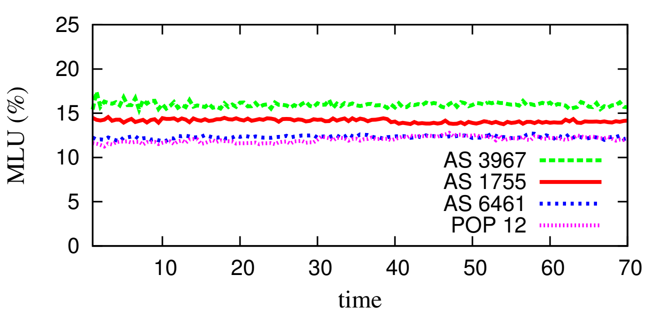

Fig. 9. Robustness of MORE over failures. At each 10's second, a random link failure occurs, and MORE uses the augmentation strategy for failure restoration over the same set of paths. The TM does not change.

VI. Implementation and Deployment Issues

MORE is a quasi-static solution, so that it is not responsive to traffic dynamics on a small time scale. Armed with recent achievements in traffic estimation, we are confident with adjusting an oblivious routing on an hourly, multi-hourly or even daily basis. However, we need to make routing adjustments when severe failures occur or if we hope to enhance the performance.

Adjustments are necessary if the current traffic deviates much from the last estimation. We can leverage traffic estimation techniques such as those in [31], [37] to keep track of the accurate traffic estimation. In contrast to the frequent collection of network information for an adaptive approach, a large time-scale traffic estimation is sufficient for MORE. Traffic estimation entails information about the traffic across the network. However, the data for traffic estimation, e.g. Simple Network Management Protocol (SNMP) data, are available from routine network management tasks [37], so that no extra network devices or software is needed.

With routing adjustments, there is an issue of how to mitigate potential routing disruptions. An approach is to exploit the robust weighted hashing [27], which claims the least service disruption when failures occur. Simulation results show that MORE is robust under varying routings due to traffic changes (Figure 7) and link failures (Figure 9), with the robust weighted hashing [27] for flow-based multipath routing. Further improvements may be achieved by exploiting the traffic burstiness, as studied recently in [30].

MORE provides an efficient implementation of oblivious routing and is amenable to gradual deployment. MORE needs to centrally compute the routing and to set up the routing at edge routers. Then an edge router splits incoming traffic according to the routing fractions. MORE does not need to collect instantaneous network information. Thus, there is no need to change the core routers. This also eases the management and operation of the deployment of MORE.

VII. Conclusions

We investigate a promising approach for stable and robust intra-domain traffic engineering in a changing and uncertain environments. We present MORE, a multipath implementation of demand-oblivious routing [5]. We evaluate the performance

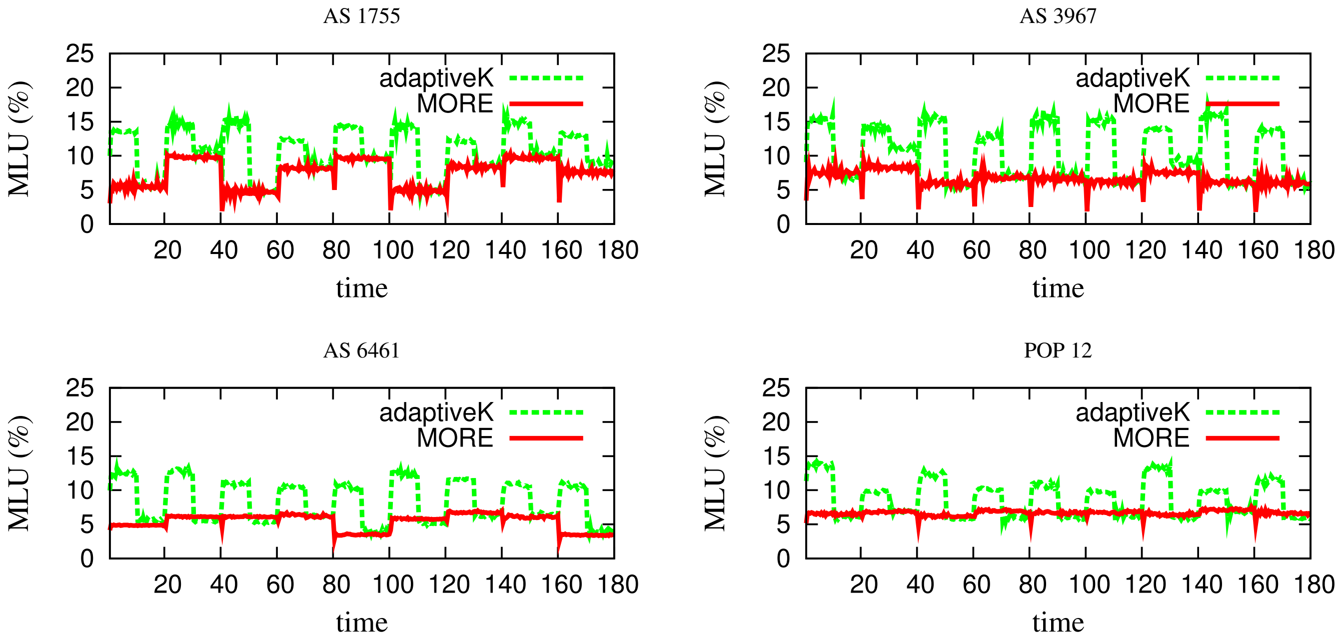

Fig. 10. Adaptive$K$ vs. MORE. During each iteration (20 seconds), for the first half, adaptive$K$ encounters an adversary attack; while for the second half, adaptive$K$ operates with an optimal routing. MORE does not change the routing over the whole run of the simulation.

of MORE by both numerical experiments and simulation. The performance study shows that MORE can obtain a close multipath approximation to [5]. The results also show the excellent performance of MORE under varying traffic demands, link failures and an adversary attack. Its performance is excellent even with a 100% error in traffic estimation.

We open the door for a viable deployment of demand-oblivious routing, thus an intra-domain traffic engineering technique robust to changing and uncertain environments.

Acknowledgments

We are thankful to Yang R. Yang (Yale University) for valuable discussions, Chun Zhang (University of Massachusetts, Amherst) for sharing the code for link weight optimization, Weiguang Shi (Random Knowledge Inc.) for discussions, in particular, on weighted hashing, and Srikanth Kandula (MIT) for discussions on TeXCP.

References

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

[2] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh. A comparison of overlay routing and multihoming route control. In *Proceedings of SIGCOMM'04*, pages 93 – 106. Portland, August 2004.

[3] E. Altman, T. Boulogne, R. E. Azouzi, T. Jimenez, and L. Wynter. A survey on networking games. *Computers and Operations Research*, 33(2):286–311, February 2006.

[4] D. Applegate, L. Breslau, and E. Cohen. Coping with network failures: Routing strategies for optimal demand oblivious restoration. In *Proceedings of SIGMETRICS'04*, pages 270–281. New York, June 2004.

[5] 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.

[6] 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.

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

[8] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *Proceedings of INFOCOM'01*, pages 1300–1309. Anchorage, April 2001.

[9] 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.

[10] 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.

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

[12] R. G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25(1):73– 85, January 1977.

[13] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang. Optimizing cost and performance for multihoming. In *Proceedings of SIGCOMM'04*, pages 79 – 92. Portland, August 2004.

[14] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proceedings of SIGCOMM'05*, pages 253 – 264. Philadelphia, August 2005.

[15] A. Khanna and J. Zinky. The revised ARPANET routing metric. In *SIGCOMM'89*, pages 45–56. Austin, September 1989.

[16] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proceedings of ACM SIGMETRICS / Performance 2004*, pages 61–72. New York, June 2004.

[17] Y. Li, J. Harms, and R. Holte. A simple method for balancing network utilization and quality of routing. In *Proceedings of IEEE ICCCN'05*, pages 71–76. San Diego, October 2005.

[18] Y. Li, J. Harms, and R. Holte. Optimal traffic-oblivious energy-aware routing for multihop wireless networks. In *Proceedings of INFOCOM'06*. Barcelona, Spain, April 2006.

[19] 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.
[20] 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.
[21] 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, Dec. 1999.
[22] A. Nucci, A. Sridharan, and N. Taft. The problem of synthetically generating ip traffic matrices: initial recommendations. *ACM SIGCOMM Computer Communication Review*, 35(3):19 – 32, July 2005.
[23] 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.
[24] S. Plotkin. Competitive routing of virtual circuits in ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1128–1136, August 1995.
[25] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in Internet-like environments. In *Proceedings of SIGCOMM'03*, pages 151–162. Karlsruhe, Germany, August 2003.
[26] H. Räcke. Minimizing congestion in general networks. In *Proceedings of 43th IEEE FOCS*, pages 43–52. Vancouver, Canada, November 2002.
[27] K. W. Ross. Hash routing for collections of shared web caches. *IEEE Network*, 11(7):37–44, Nov/Dec 1997.
[28] M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. In *the 3rd ACM SIGCOMM conference on Internet measurement*, pages 248 – 258. Miami, October 2003.
[29] A. Shaikh, J. Rexford, and K. G. Shin. Load-sensitive routing of long-lived IP flows. In *Proceedings of SIGCOMM'99*, pages 215 – 226. Cambridge, MA, USA, August 1999.
[30] S. Sinha, S. Kandula, and D. Katabi. Harnessing TCP's burstiness with flowlet switching. In *Proceedings of HotNets-III*. San Diego, November 2004.
[31] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. Traffic matrices: balancing measurements, inference and modeling. In *Proceedings of SIGMETRICS'05*, pages 362 – 373. Banff, Canada, June 2005.
[32] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of SIGCOMM'02*, pages 133–146. Pittsburgh, August 2002.
[33] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. COPE: Traffic Engineering in dynamic networks. In *SIGCOMM'06*. Pisa, Italy, September 2006.
[34] 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. In *Proceedings of ICNP'05*. Boston, November 2005.
[35] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley. On optimal routing with multiple traffic matrices. In *Proceedings of INFOCOM'05, on CD*. Miami, March 2005.
[36] Y. Zhang and Z. Ge. Finding critical traffic matrices. In *Proceedings of DSN '05*. Yokohama, Japan, June 2005.
[37] 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.
[38] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *Proceedings of SIGCOMM'03*, pages 301 – 312. Karlsruhe, Germany, August 2003.