Stable and Robust Multipath Oblivious Routing for Traffic Engineering*

Yuxi Li, Baochun Bai, Janelle Harms, and Robert Holte

Department of Computing Science, University of Alberta

Abstract. Intra-domain traffic engineering is essential for the operation of an Internet Service Provider. Demand-oblivious routing [2] promises excellent performance guarantee with changing and uncertain traffic demands. However, it is difficult 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 [2], especially when approximate knowledge of traffic is available. The study also 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.

1 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 further aggravates the problems.

There are three classes of solutions: link weight optimization [4, 13], traffic-adaptive approaches [3, 5, 10] and demand-oblivious routing [1, 2, 12]. The approach of link weight optimization guarantees performance only for a limited set of traffic demands. An adaptive approach is responsive to traffic changes, so that the issues of stability and convergence 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

* Y. Li received honorary Izaak Walton Killam memorial scholarship and Informatics Circle of Research Excellence (iCore) graduate student scholarship. B. Bai received Province of Alberta Graduate Fellowship. This research was partially supported by the Natural Science and Engineering Research Council of Canada (NSERC).

L. Mason, T. Drwiega, and J. Yan (Eds.): ITC 2007, LNCS 4516, pp. 129–140, 2007.

estimation, e.g. [14]. In [12], the performance is optimized for expected scenarios and is guaranteed for unexpected scenarios.

However, it is difficult to implement oblivious routing in [2]. A straightforward implementation is for each node to forward incoming packets according to the routing fractions computed by [2]. 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. 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.

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 [2]. MORE achieves a 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 variability. 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 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.

We are the first to investigate a feasible implementation of demand-oblivious routing [2]. We design MORE, a multipath approximation to [2]. Through extensive numerical experiments and simulation, we show the excellent performance of MORE under varying traffic matrices, link failures and an adversary attack. Our work is complementary to [1, 2] and [12]. MORE is a promising option for traffic engineering, along with link weight optimization [4, 13] and adaptive approaches, like MATE [3], TeXCP [5] and [10].

2 Preliminaries

A traffic matrix (TM) 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 $c(e)$.

Routing. A routing specifies how to route the traffic between each OD pair across a given network. OSPF and IS-IS, two popular Internet routing protocols, follow a destination-based evenly-split approach. The 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.

An *arc-routing* $f_{ij}(e)$ specifies the fraction of traffic demand d_{ij} on edge e [2]. An arc-routing is not readily implementable for either OSPF or MPLS.

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:

$$\mathrm{MLU}_{\mathrm{arc}}(\mathbf{tm}, \mathbf{f}) = \max_{e \in E} \sum_{i,j} d_{ij} f_{ij}(e)/c(e) \tag{1}$$

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

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

Performance Ratio. The routing computed by (2) does not guarantee performance for other traffic matrices. Applegate and Cohen [2] 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.

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

$$\mathrm{PERF}(\mathbf{f}, \{\mathbf{tm}\}) = \frac{\mathrm{MLU}(\mathbf{tm}, \mathbf{f})}{\mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm})} \tag{3}$$

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 $\mathbf{f}$ is an optimal routing for $\mathbf{tm}$.

When we are considering a set of traffic matrices $\mathbf{TM}$, the performance ratio of a routing $\mathbf{f}$ is defined as

$$\mathrm{PERF}(\mathbf{f}, \mathbf{TM}) = \max_{\mathbf{tm} \in \mathbf{TM}} \mathrm{PERF}(\mathbf{f}, \{\mathbf{tm}\}) \tag{4}$$

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 $\mathbf{TM}$ includes all possible traffic matrices, $\mathrm{PERF}(\mathbf{f}, \mathbf{TM})$ 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 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.

3 Multipath Oblivious Routing for Traffic Engineering

As discussed in the Introduction, there are obstacles to the implementation of oblivious routing in [2], 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. 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 (12), which gives the worst case performance guarantee.

3.1 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 $< f_{ij}^1, ..., f_{ij}^K >, \sum_k f_{ij}^k = 1, f_{ij}^k \geq 0$ 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:

$$\text{MLU}_{\text{path}}(\mathbf{tm}, \mathbf{f}) = \max_{l \in E} \sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \tag{5}$$

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:

$$\text{OPTU}_{\text{path}}(\mathbf{tm}) = \min_{\mathbf{f}} \max_{l \in E} \sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \tag{6}$$

3.2 LP Formulation

We give LP models for multipath oblivious routing. We replace the arc formulation in Applegate and Cohen [2] 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 [2], 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":

$$\begin{aligned} &\min_{r,f,d} r \\ &\mathbf{f} \text{ is a path-routing} \\ &\forall \text{ edges } l, \forall \alpha > 0 : \forall \text{ TMs } \mathbf{tm} \text{ with } \text{OPTU}_{\text{arc}}(\mathbf{tm}) = \alpha, a_{ij} \leq d_{ij} \leq b_{ij} : \\ &\quad \textstyle\sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \leq \alpha r \end{aligned} \tag{7}$$

The oblivious ratio is invariant with the scaling of TMs or the scaling of the edge capacity. Thus, when computing the oblivious ratio, it is sufficient to consider TMs with $\mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm}) = 1$. Another benefit of using TMs with $\mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm}) = 1$ 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 $\mathbf{tm}^{'} = \lambda \cdot \mathbf{tm}$. With $\lambda = 1/\mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm})$, we have $\mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm}^{'}) = 1$. Under these conditions, the master LP (7) becomes:

$$
\begin{array}{l}
\min\limits_{r,f,d} r \\
\mathbf{f} \text{ is a path-routing} \\
\forall \text{ edges } l : \forall \text{ TMs } \mathbf{tm} \text{ with } \mathrm{OPTU}_{\mathrm{arc}}(\mathbf{tm}) = 1, \lambda > 0, \lambda a_{ij} \le d_{ij} \le \lambda b_{ij} : \\
\quad \sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \le r
\end{array}
\tag{8}
$$

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

$$
\begin{cases}
\forall \text{ pairs } i \to j, k \ne i, j : \sum_{e \in out(k)} g_{ij}(e) - \sum_{e \in in(k)} g_{ij}(e) = 0 \\
\forall \text{ pairs } i \to j : \sum_{e \in out(j)} g_{ij}(e) - \sum_{e \in in(j)} g_{ij}(e) + d_{ij} = 0 \\
\forall \text{ pairs } i \to j, \forall \text{ edges } e : g_{ij}(e) \ge 0, d_{ij} \ge 0
\end{cases}
\tag{9}
$$

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

$$
\begin{array}{lr}
\max\limits_{g,d,\lambda} \sum_{ij} d_{ij} \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) & \\
\forall \text{ pairs } i \to j : \sum_{e \in out(j)} g_i(e) - \sum_{e \in in(j)} g_i(e) + d_{ij} \le 0 & \Leftarrow w_l(i,j) \\
\forall \text{ edges } e : \sum_i g_i(e) \le c(e) & \Leftarrow \pi_l(e) \\
\forall \text{ pairs } i \to j : d_{ij} - \lambda b_{ij} \le 0 & \Leftarrow \kappa_l^+(i,j) \\
\forall \text{ pairs } i \to j : -d_{ij} + \lambda a_{ij} \le 0 & \Leftarrow \kappa_l^-(i,j) \\
\forall \text{ pairs } i \to j : d_{ij} \ge 0, g_{ij}^k \ge 0, \lambda > 0 &
\end{array}
\tag{10}
$$

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 (10) guarantee the traffic can be routed with maximum link utilization of 1.

The dual of LP (10) is LP (11). To help make the derivation of the dual LP (11) clearer, we use leftarrow $\Leftarrow$ to indicate dual variables corresponding with primal constraints in LP (10). In dual LP (11), we indicate primal variables corresponding to dual constraints.

$$
\begin{array}{ll}
\min\limits_{\pi,w,\kappa^+,\kappa^-} \sum_e c(e)\pi_l(e) & \\
\forall \text{ pairs } i \rightarrow j : w_l(i,j) + \kappa_l^+(i,j) - \kappa_l^-(i,j) \geq \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) & \Leftarrow d_{ij} \\
\forall \text{ nodes } i, \forall \text{ edges } (u,v) : \pi_l(u,v) + w_l(i,u) - w_l(i,v) \geq 0 & \Leftarrow g_i(u,v) \\
\sum_{i,j}\{a_{ij}\kappa_l^-(i,j) - b_{ij}\kappa_l^+(i,j)\} \geq 0 & \Leftarrow \lambda \\
\forall \text{ edges } e : \pi_l(e) \geq 0 & \\
\forall \text{ pairs } i \rightarrow j : w_l(i,j) \geq 0, \kappa_l^+(i,j) \geq 0, \kappa_l^-(i,j) \geq 0 & \\
\forall \text{ nodes } i : w_l(i,i) = 0, \kappa_l^+(i,i) = 0, \kappa_l^-(i,i) = 0 &
\end{array}
\tag{11}
$$

According to the LP duality theory, the primal LP and its dual LP have the same optimal value if they exist. That is, LP (10) and LP (11) are equivalent. Because LP (11) is a minimization problem, we can use its objective in place of the objective of LP (10) in the "$\leq r$" constraints of LP (8). Replacing the constraint in the master LP (8) with LP (11), we obtain a single LP to compute the oblivious performance ratio using K paths.

$$
\begin{array}{l}
\min\limits_{r,f,\pi,w,\kappa^+,\kappa^-} r \\
\mathbf{f} \text{ is a path-routing} \\
\forall \text{ edges } l : \\
\quad \sum_e c(e)\pi_l(e) \leq r \\
\quad \forall \text{ pairs } i \rightarrow j : w_l(i,j) + \kappa_l^+(i,j) - \kappa_l^-(i,j) \geq \sum_k \delta_{ij}^k(l) f_{ij}^k / c(l) \\
\quad \forall \text{ nodes } i, \forall \text{ edges } (u,v) : \pi_l(u,v) + w_l(i,u) - w_l(i,v) \geq 0 \\
\quad \sum_{i,j}\{a_{ij}\kappa_l^-(i,j) - b_{ij}\kappa_l^+(i,j)\} \geq 0 \\
\quad \forall \text{ edges } e : \pi_l(e) \geq 0 \\
\quad \forall \text{ pairs } i \rightarrow j : w_l(i,j) \geq 0, \kappa_l^+(i,j) \geq 0, \kappa_l^-(i,j) \geq 0 \\
\quad \forall \text{ nodes } i : w_l(i,i) = 0, \kappa_l^+(i,i) = 0, \kappa_l^-(i,i) = 0
\end{array}
\tag{12}
$$

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 $\kappa_l^-(i,j)$.

3.3 MultiPath Selection

We discuss three approaches, sp$K$, mix$K$ and focus$K$, for multiple paths selection for each OD pair, to achieve a low oblivious ratio.

In sp$K$, we select K shortest paths w.r.t. 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 [7], where 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 [7] by considering range restrictions on traffic demands.

After computing the modified oblivious routing using the extended LP to [7], 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.

4 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]

Topology. ISP topologies and traffic demands are regarded as proprietary information. The Rocketfuel project [11] deployed new techniques to measure ISP topologies and made them publicly available. The OSPF weights on the links are also provided. The capacities of links are assigned according to the CISCO heuristics as in [2], referred to as InvCap, i.e., the link weight is inversely proportional to the link capacity. POP 12 is the tier-1 ISP topology in Nucci et al. [8], with the scaled link capacity provided in [8]. We also use random topologies generated by GT-ITM.[2]

Gravity TM. Similar to [2, 5], we use the Gravity model [14] to determine the estimated traffic matrices. The Gravity model is developed in [14] 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 [2], 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. [8] 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 [8]. 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 [2], 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 $[d_{ij}/w, w * d_{ij}]$.

4.1 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 [2], which can achieve the lowest oblivious ratio for a given topology. Hereafter, we refer to the method in Applegate and Cohen [2] as **AC**. Recall that it is non-trivial to implement the routing computed by AC. Thus a close multipath approximation to AC is desirable.

In Figure 1, 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 (see [6]), sp$K$ and mix$K$, for $K = 10, 20$, have similar results, with performance very close to AC.

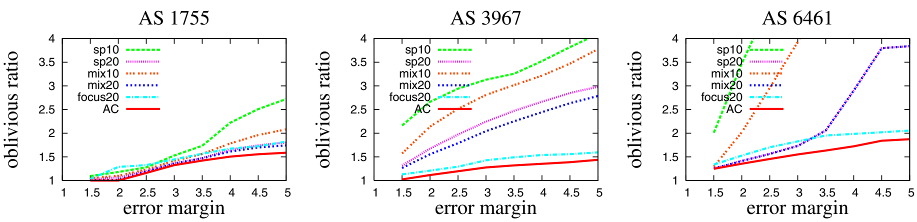

Fig. 1. Oblivious ratio vs. error margin for various path selection methods

Experiments also show the multipath selection methods outperform the link weight optimization [4] and InvCap on ISP and random topologies. See [6] for details.[3] In later studies, we use path selection methods as follows: mix20 for AS 1755, focus20 for AS 3967 and AS 6461, and mix10 for POP 12. When there is approximate knowledge of traffic demands, we use error margin $w = 2.0$, which can be interpreted as a tolerance of 100% error in traffic estimation.

4.2 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[4]. We implement the robust weighted hashing by Ross [9], so that traffic can be split into multiple paths according to the routing fraction of each path.

[3] See [6] also for numerical experiments on link failures and adversary attacks.

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

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 [8]. 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 2.[5] 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 to adaptive$K$. MORE also has similar performance for AS 6461 and POP 12, see [6].

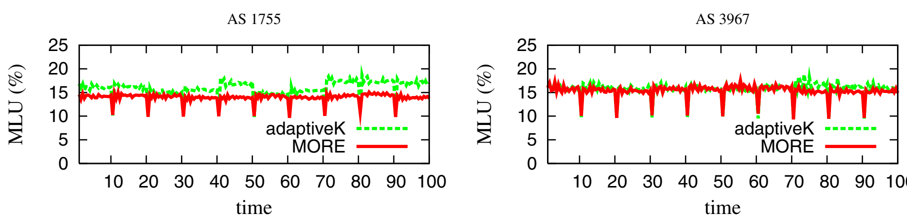

Fig. 2. Robustness of MORE over varying TMs and routings. For each 10 seconds, a random TM is generated, and MORE responds with an optimal multipath oblivious routing over the same set of paths. For adaptive$K$, it computes, for each TM, an optimal routing on K-shortest paths ($K = 20$).

TeXCP vs. MORE. We compare MORE with TeXCP, an adaptive multipath routing approach [5]. 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 [5], 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 [5]. Traffic is generated according to a Gravity TM. During time intervals $[25, 50]$ and $[75, 100]$,

[5] For Figure 2 and 5, there are downward spikes for both adaptive$K$ and MORE. These are due to the transition of stopping and starting TMs.

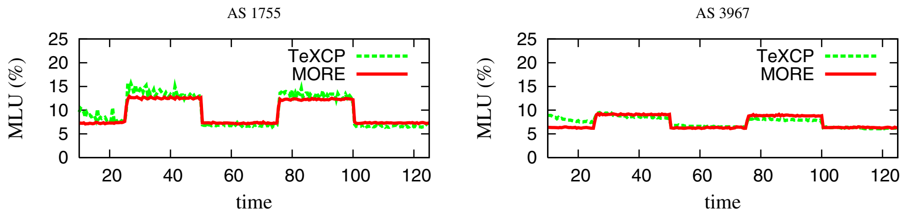

Fig. 3. TeXCP vs. MORE. During time interval [25,50] and [75,100], extra random traffic is generated.

the traffic volume is doubled for each OD pair. Figure 3 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 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 AS 1755. 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. MORE has similar performance for AS 6461 and POP 12, see [6]. With a longer time period (35 seconds) for the "warm-up", TeXCP has similar performance.

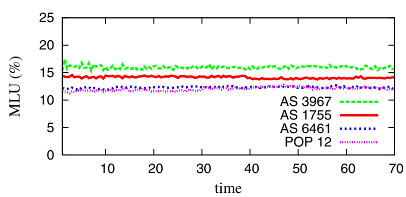

Fig. 4. Robustness of MORE over failures. At each 10's second, a random link failure occurs, and MORE uses the augmentation strategy for failure restoration over the same set of paths. The TM does not change.

Link failure. We study the robustness of MORE over link failures using simulation. We investigate two restoration strategies: *reoptimization* and *augmentation*. In reoptimization, we reoptimize multipath oblivious routing for the new topology after link failures occur. In augmentation, we reoptimize only for the affected OD pairs, which use the link(s) with failure. At each 10's second, a random link failure occurs with 20% link capacity reduction. After each link failure, the augmentation strategy 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 4 shows the results. We observe that the networks have rather stable performance, after several consecutive link failures. Reoptimization has similar performance.

Adversary attack. We introduce an attack which can exploit a routing **f**, by generating a TM for **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. An adversary TM can be obtained using LP (10).

We compare MORE and adaptive$K$ under an adversary attack.[6] 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. 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.

The results are shown in Figure 5. 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. More has similar performance for AS 6461 and POP 12, see [6].

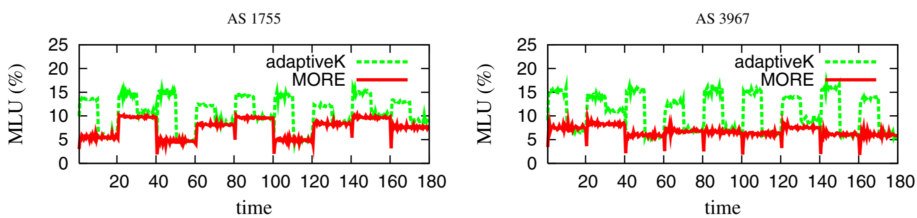

Fig. 5. Adaptive$K$ vs. MORE. During each iteration (20 seconds), for the first half, adaptive$K$ encounters an adversary attack; while for the second half, adaptive$K$ operates with an optimal routing. MORE does not change the routing over the whole run of the simulation.

5 Conclusions

We investigate a promising approach for stable and robust intra-domain traffic engineering in a changing and uncertain environment. We present MORE, a multipath implementation of demand-oblivious routing [2]. We evaluate the performance of MORE by both numerical experiments and simulation. The performance study shows that MORE can obtain a close multipath approximation to [2]. The results also show the excellent performance of MORE under varying traffic demands, link failures and an adversary attack. Its performance is

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

excellent even with a 100% error in traffic estimation. See [6] for more discussions and experimental results.

References

[1] Applegate, D., Breslau, L., Cohen, E.: Coping with network failures: Routing strategies for optimal demand oblivious restoration. In: Proceedings of SIGMETRICS'04, pp. 270–281, New York (June 2004)
[2] Applegate, D., Cohen, E.: Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In: Proceedings of SIGCOMM'03, pp. 313–324, Karlsruhe, Germany (August 2003)
[3] Elwalid, A., Jin, C., Low, S., Widjaja, I.: MATE: MPLS adaptive traffic engineering. In: Proceedings of INFOCOM'01, pp. 1300–1309, Anchorage (April 2001)
[4] Fortz, B., Thorup, M.: Optimizing OSPF/IS-IS weights in a changing world. IEEE Journal on Selected Areas in Communications 20(4), 756–767 (2002)
[5] Kandula, S., Katabi, D., Davie, B., Charny, A.: Walking the tightrope: Responsive yet stable traffic engineering. In: Proceedings of SIGCOMM'05, pp. 253–264, Philadelphia (August 2005)
[6] Li, Y., Bai, B., Harms, J., Holte, R.: Multipath oblivious routing for traffic engineering - stable and robust routing in changing and uncertain environments. Technical Report, TR06-11, Department of Computing Science, University of Alberta (May 2006)
[7] Li, Y., Harms, J., Holte, R.: A simple method for balancing network utilization and quality of routing. In: Proceedings of IEEE ICCCN'05, pp. 71–76, San Diego (October 2005)
[8] Nucci, A., Sridharan, A., Taft, N.: The problem of synthetically generating ip traffic matrices: initial recommendations. ACM SIGCOMM Computer Communication Review 35(3), 19–32 (2005)
[9] Ross, K.W.: Hash routing for collections of shared web caches. IEEE Network 11(7), 37–44 (1997)
[10] Shaikh, A., Rexford, J., Shin, K.G.: Load-sensitive routing of long-lived IP flows. In: Proceedings of SIGCOMM'99, pp. 215–226, Cambridge, MA, USA (August 1999)
[11] Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: Proceedings of SIGCOMM'02, pp. 133–146, Pittsburgh (August 2002)
[12] Wang, H., Xie, H., Qiu, L., Yang, Y.R., Zhang, Y., Greenberg, A.: COPE: Traffic Engineering in dynamic networks. In SIGCOMM'06, Pisa, Italy (September 2006)
[13] Zhang, C., Ge, Z., Kurose, J., Liu, Y., Towsley, D.: Optimal routing with multiple traffic matrices: Tradeoff between average case and worst case performance. In: Proceedings of ICNP'05, Boston (November 2005)
[14] Zhang, Y., Roughan, M., Duffield, N., Greenberg, A.: Fast accurate computation of large-scale IP traffic matrices from link loads. In: Proceedings of SIGMETRICS'03, pp. 206–217, San Diego (June 2003)