Semi-Oblivious Traffic Engineering with SMORE*

Extended Abstract (ANRW'18 submission #47)

Praveen Kumar
Cornell University

Yang Yuan
Cornell University

Chris Yu
CMU

Nate Foster
Cornell University

Robert Kleinberg
Cornell University

Petr Lapukhov
Facebook

Chiun Lin Lim
Facebook

Robert Soulé
Università della Svizzera italiana

ABSTRACT

Wide-area networks are expected to deliver high performance while ensuring reliability in the presence of various operational constraints and failures. Existing traffic engineering (TE) approaches, often, are unable to achieve both simultaneously. A key factor that governs the quality of a TE system is the set of routing paths used. This paper proposes SMORE—a *semi-oblivious* TE system which combines oblivious routing to select a good set of paths with dynamic rate adaptation. Through extensive evaluation, we show that SMORE achieves near-optimal performance while ensuring good reliability in practical settings.

1 TRAFFIC ENGINEERING

Operators of wide-area networks (WANs) use traffic engineering (TE) to route traffic efficiently while accommodating for time-varying demands and operating conditions. Such WAN-TE systems aim to improve the network utilization by distributing the traffic, resulting in improved operational efficiency. To this end, various TE systems have been proposed over the years, ranging from conventional distributed approaches [2, 5] to recent centralized ones [4, 6]. However, it is difficult to achieve the competing goals of performance, i.e. high throughput and low latency, and reliability, i.e. robustness to failures, and few manage to achieve both [3].

Optimal TE. The standard approach to optimal TE formulates routing as an optimization problem—given a network topology and a traffic matrix (TM) specifying the bandwidth demand for every pair of nodes in the topology, compute a weighted set of routing paths that optimizes certain criteria, e.g. minimizes maximum link utilization (MLU). This is the multi-commodity flow (MCF) problem, and is usually solved using linear programming (LP). An MCF-based TE system needs to continuously *(i)* monitor traffic to estimate real-time TM, *(ii)* construct and solve the MCF instance, *(iii)* translate the solution to a routing scheme, and *(iv)* update the forwarding entries on routers. While this approach is optimal in theory, several factors make it impractical at scale:

*Based on previously published paper [9] at USENIX NSDI '18.

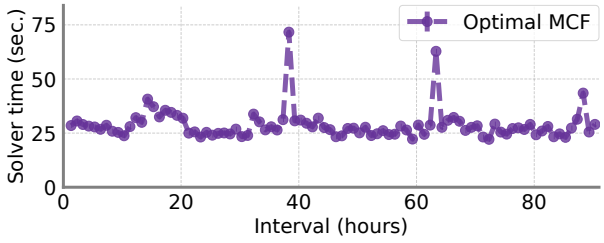


Figure 1: Time taken to solve MCF instances.

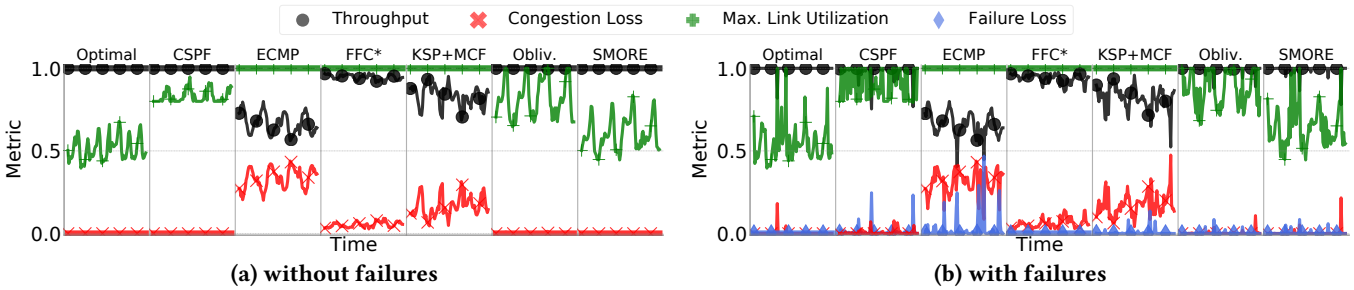
(1) Solving MCF quickly becomes a bottleneck and this limits the system's responsiveness to changing demands. For instance, Fig. 1 shows the time taken by an optimized LP solver to solve MCF instances for a WAN.
(2) Small changes in TMs can lead to a significantly different solution to MCF, which leads to excessive routing churn.
(3) Ensuring continuous consistent updates to forwarding state at routers adds significant management complexity.

Semi-oblivious system model. Taking a step back, fundamentally, there are two decisions that a TE system has to make: *(i)* which paths to route traffic over and *(ii)* what sending rates to use for each path when mapping traffic to multiple paths. While changing the set of paths frequently is impractical as it requires updating multiple geo-distributed routers, causes high churn and needs careful updates, changing the sending rates is a relatively fast and inexpensive operation. Hence, we adopt a system model that uses a static set of pre-computed paths but adapts sending rates based on current demands. Although some recent approaches [4, 6, 10] also follow a similar model, they often choose paths which are sub-optimal to achieve good performance and reliability.

2 SMORE DESIGN

SMORE [9] follows the aforementioned two-phase model by combining oblivious routing [11] for careful path selection with a restricted MCF-like formulation for dynamic rate optimization.

Path properties. The key insight for SMORE is that path selection has an outsized impact on the performance and



(a) without failures **(b) with failures**

Figure 2: Expected performance and robustness on Facebook's WAN over half a week.

robustness of TE systems. A good path selection algorithm should compute paths with the following properties:
(1) *low stretch* to minimize latency for traffic,
(2) *high diversity* to ensure robustness to failures, and
(3) *good load balancing* to achieve high performance.
In order to achieve good load balancing properties, the algorithm must be: *(i)* capacity-aware so as not to over-utilize any low-capacity links and *(ii)* globally optimizing while considering all source-destination pairs simultaneously so that different pairs do not end up cumulatively over-utilizing a common set of links. Unfortunately, current TE systems fail to guarantee at least one of these properties.

Oblivious routing. SMORE's path selection algorithm uses Räcke's oblivious routing [11] to pick a static set of paths that are low-stretch, diverse and load-balance traffic naturally. It is an iterative algorithm based on a structure called *decomposition tree*, or simply routing tree. In each iteration, the algorithm generates a new routing tree using an approximation algorithm [1] which ensures that the paths based on the routing tree have low stretch. Before the subsequent iteration, it updates the weight of each link based on its capacity and cumulative utilization in the previous trees in order to penalize further use of already heavily utilized links. This ensures that the ensemble of routing trees produces a diverse set of paths with good load balancing guarantees. In fact, oblivious routing, which doesn't even consider the TMs, has been shown to be $O(\log n)$ competitive with optimal MCF.

Rate Adaptation. In order to dynamically optimize the routing scheme as the TM changes over time, SMORE computes updated values of path weights by constructing an LP which minimizes MLU while using the same set of paths. This LP is much simpler than the LP for the corresponding MCF instance because the paths are fixed in this case, and hence the LP can be solved almost instantaneously.

3 EVALUATION

We evaluate SMORE under various operational conditions and compare it with other TE systems using the YATES framework [8]. We perform high fidelity simulations based on topology and hourly TMs collected from Facebook's backbone network that carries production traffic. Fig. 2a shows the performance in terms of normalized throughput, MLU, and fraction of traffic that could not be admitted. We find that SMORE has near-optimal performance with MLU within 16% of optimal, on average. In another set of experiments to compare robustness, we fail a unique link every hour and measure performance after allowing TE systems to react to failures. As shown in Fig. 2b, SMORE is able to deliver almost 100% throughput and still remains close to optimal in terms of MLU. With respect to latency, we find SMORE to be competitive with shortest path based approaches.

Through large-scale simulations with other ISP topologies from Internet Topology Zoo [7], we find that these results generalize fairly well over a wide range of scenarios.

4 DISCUSSION

Despite the vast literature on TE, it continues to be an area of active research. The conventional approach to TE relies on carefully tuning link weights in distributed routing protocols to steer traffic in desired ways [2, 5]. However, optimizing link weights for good performance is a computationally difficult problem. With centralized control over the network becoming practical, recent proposals [4, 6] advocate selecting paths and load balancing traffic in a globally optimal manner while optimizing for metrics such as throughput, fairness, utilization etc.

There is a fundamental trade-off between performance and reliability for TE systems. Most systems optimize for one or the other, but not both. SMORE navigates these trade-offs by combining careful path selection with dynamic load balancing. In doing so, SMORE also avoids the overheads, such as churn, of theoretically optimal approaches that make them impractical. We conducted extensive experiments using data from content providers and ISPs, and our results show that SMORE achieves near-optimal performance and high level of robustness while satisfying various operational constraints. Finally, SMORE is also readily deployable in systems with centralized control.

REFERENCES

[1] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2003. A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics. In *35th STOC*.

[2] B. Fortz and M. Thorup. 2000. Internet Traffic Engineering by Optimizing OSPF Weights. In *IEEE INFOCOM*.

[3] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2016. Evolve or Die: High-Availability Design Principles Drawn from Google's Network Infrastructure. In *ACM SIGCOMM*.

[4] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. 2013. Achieving High Utilization with Software-Driven WAN. In *ACM SIGCOMM*.

[5] Christian E Hopps. 2000. Analysis of an equal-cost multi-path algorithm. http://tools.ietf.org/html/rfc2992. (2000).

[6] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2013. B4: Experience with a Globally Deployed Software Defined WAN. In *ACM SIGCOMM*.

[7] Simon Knight, Hung X. Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. 2011. The Internet Topology Zoo. http://www.topology-zoo.org. (2011).

[8] Praveen Kumar, Chris Yu, Yang Yuan, Nate Foster, Robert Kleinberg, and Robert Soulé. 2018. YATES: Rapid Prototyping for Traffic Engineering Systems. In *ACM SOSR*.

[9] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiun Lin Lim, and Robert Soulé. 2018. Semi-Oblivious Traffic Engineering: The Road Not Taken. In *USENIX NSDI*.

[10] Hongqiang Harry Liu, Srikanth Kandula, Ratul Mahajan, Ming Zhang, and David Gelernter. 2014. Traffic Engineering with Forward Fault Correction. In *ACM SIGCOMM*.

[11] Harald Räcke. 2008. Optimal Hierarchical Decompositions for Congestion Minimization in Networks. In *40th STOC*.