Multi-Commodity Flow Traffic Engineering with Hybrid MPLS/OSPF Routing

Mingui Zhang
Tsinghua University
Beijing, China
mingui.zhang@gmail.com

Bin Liu
Tsinghua University
Beijing, China
liub@tsinghua.edu.cn

Beichuan Zhang
The University of Arizona
Tucson, Arizona
bzhang@arizona.edu

***Abstract*—The common objective of network traffic engineering is to minimize the maximal link utilization in a network in order to accommodate more traffic and reduce the chance of congestion. Traditionally this is done by either optimizing OSPF link weights or using MPLS tunnels to direct traffic. However, they both have problems: OSPF weight optimization triggers network-wide convergence and significant traffic shift, while pure MPLS approach requires a full mesh of tunnels to be configured throughout the network. This paper formulates the traffic engineering problem as a Multi-Commodity Flow problem with hybrid MPLS/OSPF routing (MCFTE). As a result, the majority of traffic is routed by regular OSPF, while only a small number of MPLS tunnels are needed to fine-tune the traffic distribution. It keeps OSPF link weights unchanged to avoid triggering network convergence, and needs far fewer MPLS tunnels than the full-mesh to adjust traffic. Compared with existing hybrid routing approaches, MCFTE achieves the optimal link utilization, runs about two orders of magnitude faster, and is more robust against measurement inaccuracy in traffic demand.**

I. Introduction

Network operators frequently manipulate how data traffic flows through their networks in order to increase the throughput of their networks, reduce congestion and therefore improve overall quality of service. The common goal of traffic engineering (TE) is to minimize the maximal link utilization in the network, which traditionally is achieved by either optimizing the link weights in the intra-domain routing protocol (*e.g.*, OSPF), or setting up full-mesh MPLS tunnels connecting all ingress-egress router pairs and splitting traffic among multiple MPLS tunnels.

The weight optimization approach needs to adjust link weights from time to time in order to accommodate changing traffic demand. Changing link weight will trigger network-wide OSPF convergence process, which not only takes time to complete, but also induces potentially large traffic shift in the network, and both of these side effects can cause service degradation such as packet loss and delay jitter. Due to these reasons, changing link weights can only be done infrequently (*e.g.*, once per day [1] [2]), which limits the effectiveness of traffic engineering in face of varying traffic demand. Moreover, the weight optimization problem is NP-hard [3] and can only be tackled by heuristics, which may not get the optimal solutions and sometime do not even converge.

Multi-Protocol Label Switching (MPLS) enables routers to forward traffic along explicitly configured paths. This flexibility makes it easier to do traffic engineering than relying on conventional IP routing [4] [5]. Although MPLS has been deployed in many large ISPs, a pure MPLS traffic engineering approach will require a full mesh of MPLS tunnels, *i.e.*, Label Switching Paths (LSP), between any ingress and egress routers, which puts a lot of management burden on large networks [6] [7].

Hybrid routing uses both OSPF and MPLS. It relies on OSPF to carry most traffic without changing link weights, and at the same time it uses a small number of MPLS LSPs to fine-tune the traffic distribution over different links for the traffic engineering goals. The OSPF link weight is not adjusted over time, therefore network convergence and large traffic shift is avoided. When traffic demand changes, it is the MPLS LSPs that are adjusted to accommodate these changes to maintain target traffic distribution. Thus hybrid routing combines the advantages of both OSPF and MPLS TE. However, existing work all regard the hybrid routing as NP-hard and resort to heuristics for solutions, which are not only slow but also do not give optimal results. For examples, GreedyHybrid uses a greedy method to compute LSPs which can guarantee neither global nor local optimal solution [8], GAHybrid uses genetic algorithm to search for the solution [9], and SAMTE uses simulated annealing meta-heuristic to compute a set of LSPs [6].

We propose Multi-Commodity Flow Traffic Engineering (MCFTE), which formulates traffic engineering as a linear programming problem and realizes the optimal solution by hybrid MPLS/OSPF routing. Given the network topology, traffic demand, and OSPF link weights, MCFTE will compute the MPLS LSPs that are needed to establish and the traffic split ratios between OSPF and MPLS. MCFTE inherits the benefits of hybrid routing by using only a small number of MPLS paths to complement regular OSPF routing, thus it avoids the drawbacks of OSPF weight optimization and full MPLS mesh. Compared with existing hybrid routing approaches, MCFTE achieves the optimal link utilization in a network, runs about two orders of magnitude faster, and is more robust against measurement errors in traffic demand. These features make MCFTE a good candidate for real-time, distributed traffic engineering solution in operational networks.

The rest of the paper is organized as follows. Section II presents the formulation of the hybrid routing using Multi-Commodity Flow and reveals the advantages of MCFTE. Section III evaluates MCFTE using three different real topologies and their traffic demands. Section IV reviews related work and Section V concludes the paper.

II. Problem Formulation

We assume that a network runs a link-state routing protocol such as OSPF and also is capable of setting up MPLS paths throughout the network. The TE problem is that given the network topology, traffic matrix (*i.e.*, traffic demand between any ingress-egress pair), and OSPF routing, which MPLS paths need to be configured and how to split the traffic between OSPF and MPLS so that the maximal link utilization in the network is minimized. We formulate this problem using multi-commodity flows as follows.

The network is represented by a directed graph, $G = (N, A)$. Each arc l has capacity $c(l)$. Two binary parameters I and O are defined. I_l^v denotes whether arc l's head is connected to node v, and O_l^v denotes whether arc l's tail is connected to node v. In the traffic matrix D, each $D(s,t)$ represents the traffic volume that flows from the ingress router s to the egress router t. According to the theory of MCF, $D(t,t) = -\sum_{s \in N, s \neq t} D(s,t)$. A binary parameter $P_l^{s,t}$ represents whether the OSPF route from s to t goes through l. A flow variable f_l^t denotes the amount of the MPLS traffic from all the other nodes to t that goes through link l. Variable $u(l)$ is the utilization of link l. Variable $L_{OSPF}(l)$ represents the traffic that is routed according to OSPF on link l while variable $L_{MPLS}(l)$ is the traffic that is routed according to MPLS. Variable $\alpha(s,t)$ represents the percentage of $D(s,t)$ that is routed by MPLS. The traffic engineering problem then can be formulated as the following Linear Programming (LP) problem.

$$\textbf{min} \quad U \tag{1}$$

s.t.

$$\sum_{l \in A} f_l^t O_l^s - \sum_{l \in A} f_l^t I_l^s = \alpha(s,t) D(s,t) \qquad s,t \in N \tag{2}$$

$$L_{MPLS}(l) = \sum_{t \in N} f_l^t \qquad l \in A \tag{3}$$

$$L_{OSPF}(l) = \sum_{s,t \in N} P_l^{s,t} (1 - \alpha(s,t)) D(s,t) \qquad l \in A \tag{4}$$

$$u(l) = \frac{L_{OSPF}(l) + L_{MPLS}(l)}{c(l)} \qquad l \in A \tag{5}$$

$$f_l^t \geq 0 \qquad l \in A;\ t \in N \tag{6}$$

$$0 \leq \alpha(s,t) \leq 1 \qquad s,t \in N \tag{7}$$

$$0 \leq u(l) \leq U \qquad l \in A \tag{8}$$

The solution to the above problem will give the optimal LSPs and their required bandwidths in variable $L_{MPLS}(l)$. The constraint in Equation (4) is our contribution, and no previous work has done this [10]. This constraint guides the LP solver to search for the solution that includes the OSPF routes, so that fewer number of MPLS paths will be needed. In a typical case of our evaluation scenarios, MCFTE only needs four LSPs, while classical MCF without Equation (4) needs 43 LSPs. Detailed evaluations will be presented in the next section.

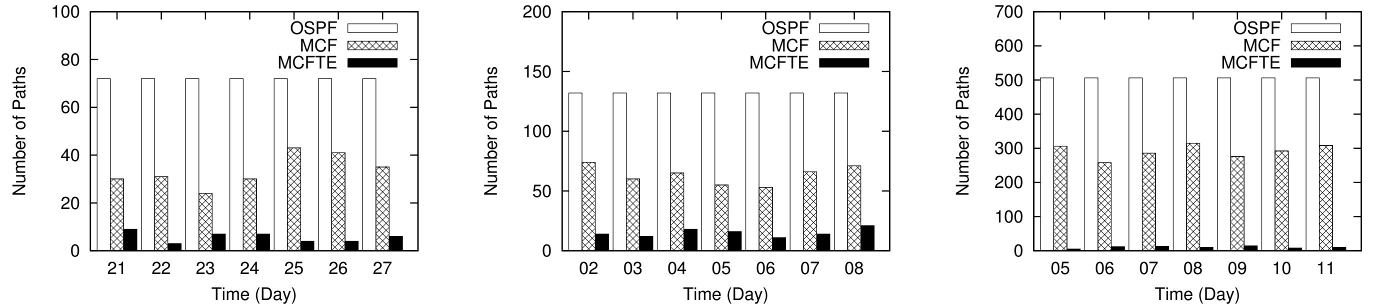
MCFTE can be much more responsive than other TE methods since its input and output can be obtained very quickly and its impact to the network is incremental. The input information to MCFTE includes the network topology, traffic matrix, and OSPF link weights. Among them, network topology and link weights are available in OSPF's link-state database. Traffic matrix can be computed from measured link utilization data. According to Zhang *et al.* [11], a backbone-router to backbone-router traffic matrix for a tier-1 ISP network can be computed in 5 seconds on a 336MHz Ultrasparc-II machine back in 2002. The output of MCFTE is the LSPs that need to be configured and the traffic amount that these LSPs will carry. As we will show in the evaluation, solving MCFTE problem takes no more than a few tens of seconds. MCFTE does not change OSPF link weights, therefore the drawbacks of network convergence and large traffic shift are avoided. When the traffic demand changes over time, MCFTE must change LSP setup to adjust the traffic distribution. Such adjustments are incremental in that they only impacts a small number of routers and a small amount of traffic that are involved in the LSPs that need to be changed. Overall, when traffic demand changes, MCFTE is able to quickly recompute the optimal solution, set up the LSPs, and only affect the network where it is necessary. Therefore it is possible to run MCFTE much more frequently (*e.g.*, every few minutes) than other TE methods to be responsive to changing traffic.

III. Evaluation

We use several real network topologies and their traffic matrices to evaluate MCFTE. Internet2 topology is configured according to the data from [12]. Abilene, GEANT and AT&T topologies come from the TOTEM toolbox [13]. All the topologies contain OSPF link weights, which are used to generate the OSPF routes. To obtain the traffic matrix for Internet2, we take the netflow data from [12], and generate one week traffic matrix using TOTEM. The measured traffic matrices of Abilene and GEANT are available from TOTEM project [14], while estimated traffic matrices of Abilene is downloaded from [15]. AT&T's traffic matrix is not publicly available. In measuring MCFTE's computation time, we use a randomly generated traffic matrix with AT&T's topology. All the evaluation is done by the open source LP solver GLPK on a Linux machine with a 3.00GHz Intel Pentium 4 CPU and 1 GB memory.

A. The Number of LSPs

MCFTE achieves optimal traffic engineering with only a small number of LSPs. Figure 1 compares the number of LSPs under MCFTE and traditional MCF using three different topologies and traffic matrices on different days in a week. The number of OSPF routes is shown for reference, which is the same as $n * (n-1)$ where n is the number of routers in the network. In theory traditional MCF would require full-mesh LSPs, but since we use hybrid routing and some LSPs are the same as OSPF paths, the traditional MCF does not need to set up full-mesh LSPs in the evaluation. MCFTE requires much fewer paths than traditional MCF, which demonstrates



(a) At each 00h00 on Internet2 from 2008-08-21 to 2008-08-27

(b) At each 00h00 on Abilene from 2004-09-02 to 2004-09-08

(c) At each 00h00 on GEANT from 2005-05-05 to 2005-05-11

Fig. 1. The number of LSPs that need to be established to optimize the objective of traffic engineering

TABLE I
CPU TIME OF MCFTE

Topology	#Nodes	#Links	MCFTE_{totem}	MCFTE_{alone}
Internet2	9	26	83.81 ms	10.0 ms
Abilene	12	30	110.38 ms	20.0 ms
GEANT	23	40	323.02 ms	90.0 ms
AT&T	154	364	26.49 s	13.84 s

TABLE II
CPU TIME OF DIFFERENT TE METHODS

Method	Internet2	Abilene	GEANT
IGPWO	6.33 s	11.67 s	120.54 s
SAMTE	16.52 s	28.91 s	24.12 s
MCFTE	83.81 ms	110.38 ms	323.02 ms

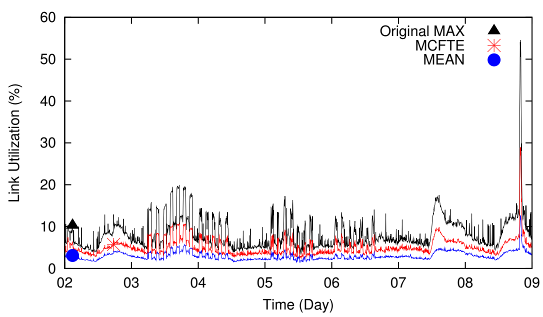


Fig. 2. Link utilization of Abilene from 2004-09-02 to 2004-09-08

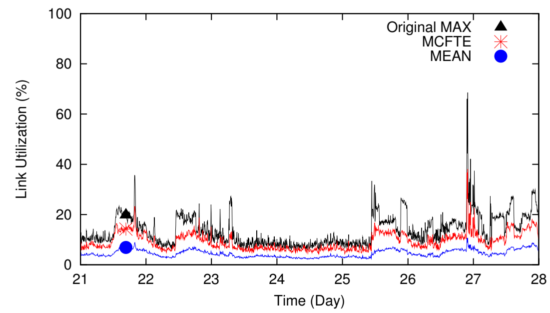


Fig. 3. Link utilization of Internet2 from 2008-08-21 to 2008-08-27

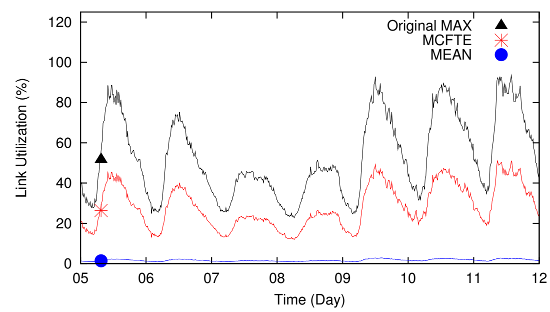


Fig. 4. Link utilization of GEANT from 2005-05-05 to 2005-05-11

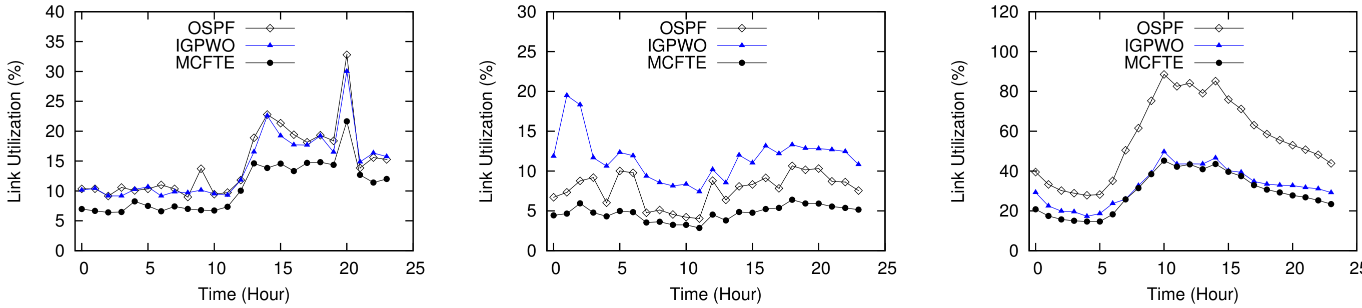
the effectiveness of Equation (4) in the MCFTE problem formulation. The number of LSPs required by MCFTE is only a small fraction of the full-mesh. We also run a test with AT&T topology, which contains 154 nodes and 364 links with a randomly generated traffic matrix, and MCFTE only needs 31 LSPs.

B. CPU Time

We measure the CPU time by MCFTE on different topologies and compare it with other TE methods. As Table I shows, MCFTE computation is generally fast. MCFTE_{totem} is the CPU time when MCFTE is implemented within the TOTEM toolbox, and MCFTE_{alone} is the CPU time when MCFTE is implemented standalone without the overhead of the toolbox. In both cases, it takes sub-second for small to medium topologies, and for the large AT&T topology it still just takes a couple tens of seconds. Table II compares the CPU time between MCFTE, SAMTE (a previously proposed hybrid routing TE solution), and IGPWO (IGP Weight Optimization). The other two methods are part of the TOTEM toolbox. The result shows that MCFTE is about two orders of magnitude faster than SAMTE and IGPWO.

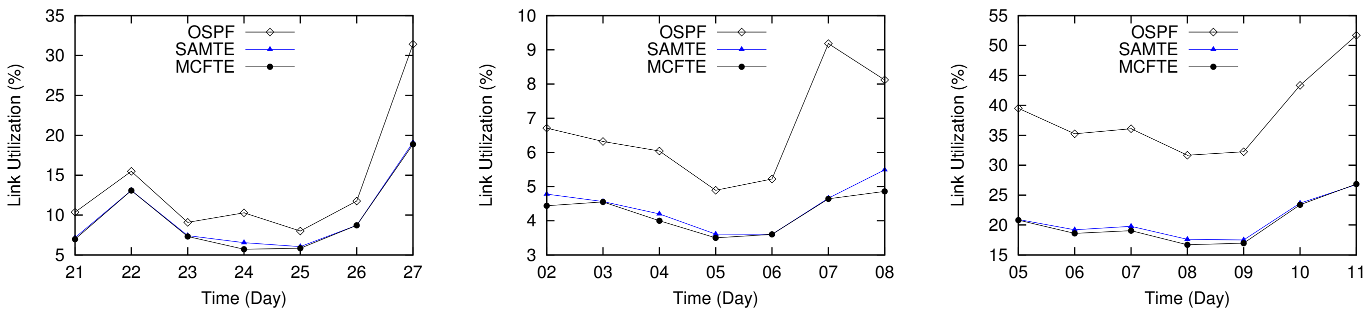
C. Maximal Link Utilization

The objective of the traffic engineering problem is to minimize the maximal link utilization. MCFTE is supposed to provide the optimal solution to the TE problem. We use three topologies and real traffic matrices to evaluate MCFTE and other TE methods regarding the maximal link utilization.



(a) At each hour on Internet2 on 2008-08-21 (b) At each hour on Abilene on 2004-09-02 (c) At each hour on GEANT on 2005-05-05

Fig. 5. Maximal Link Utilization of MCFTE and IGP Weight Optimization



(a) At each 00h00 on Internet2 from 2008-08-21 to 2008-08-27. (b) At each 00h00 on Abilene from 2004-09-02 to 2004-09-08. (c) At each 00h00 on GEANT from 2005-05-05 to 2005-05-11.

Fig. 6. Maximal Link Utilization of MCFTE and SAMTE

Figure 2 shows Abilene's link utilization sampled every 5 minutes from 2004-09-02 through 2004-09-08. The network is lightly loaded most of the time as the mean link utilization is usually below 5% and the peak link utilization often fluctuates between 5% and 20% and only in one occasion it jumps over 50%. MCFTE is able to reduce the maximal link utilization throughout the entire measurement period. For example, the maximal link utilization on 2004-09-02 at 00h00 is reduced from 6.71% to 4.44%. Figure 3 shows Internet2's link utilization from 2008-08-21 through 2008-08-27. The network is in general more loaded than Abilene. Again, MCFTE is able to reduce the maximal link utilization throughout the week. Figure 4 shows GEANT's link utilization sampled every 15 minutes from 2005-05-05 through 2005-05-11. It has an obvious diurnal pattern as the traffic reaches the peak during the day and the bottom during the night. Since the gap between the maximal link utilization and mean link utilization is quite high, MCFTE's reduction of maximal link utilization is much more pronounced than in the other two networks.

Next we compare the maximal link utilization under different TE methods using the TOTEM toolbox. IGPWO is tested using the default setting, which does a Tabu search for integer OSPF link weights starting randomly from [0, 20] and the maximum number of iterations is set to 500. Due to its heuristic nature, the search may not converge after 500 iterations and the outcome may not be the global optimal. Figure 5 shows the results in Internet2, Abilene and GEANT. For Internet2, IGPWO only slightly reduces the maximal link utilization (Figure 5(a)). It even performs worse than OSPF in Abilene (Figure 5(b)), since the heuristic cannot find better link weights within 500 iterations from the randomly selected starting values. IGPWO shows significant benefit only for GEANT (Figure 5(c)). In all three cases, MCFTE outperforms IGPWO.

We also compare MCFTE with SAMTE using "SAMTEMaxLoadOf" as the score function and parameters generated by the "Generate Parameters" function of SAMTE tool in TOTEM. Figure 6 shows that SAMTE can reduce maximal link utilization significantly, but can never outperform MCFTE, which is the optimal solution. One observation from the simulations is that SAMTE does not produce the exact same outcome due to its heuristic nature. Therefore it would be very difficult to deploy such a solution distributedly.

D. Robustness Against Inaccuracy in Traffic Matrix

The evaluation so far has assumed that the traffic matrix is known every time we run MCFTE or other TE methods. In reality, it takes time to measure, compute and report traffic matrices [11] [16]. No matter how quick traffic matrix can be obtained, we will never be able to predict the exact traffic matrix of a future time. Therefore, all TE methods must use estimates of the traffic matrix to decide the routing paths. The most common approach is to use a recently measured traffic matrix to calculate the routing paths for immediate future. A good TE method should be robust to the inaccuracy of the

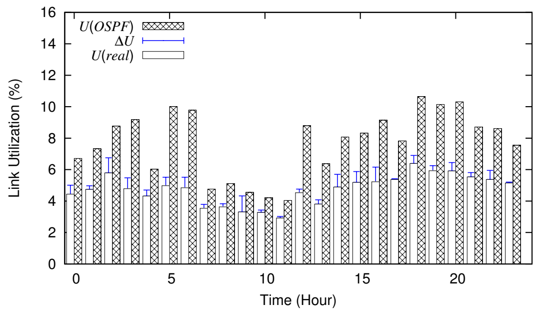


Fig. 7. The robustness of MCFTE against the inaccuracy in the traffic matrices of Abilene. $\Delta U = U(estimated) - U(real)$

traffic matrix estimates. In other words, even if the actual traffic demand is somewhat different from the traffic matrix used in the TE computation, the resulting routing paths should still have reasonably low link utilization. For instance, Roughan *et al.* [1] have demonstrated that weight optimization is robust. In this subsection, we evaluate MCFTE's robustness.

For the purpose of evaluation, we need a traffic matrix estimate and actual traffic demand in order to compare MCFTE's performance using them. For Abilene, we get the estimated traffic matrices from [15]. For Internet2, we simply use the traffic matrix measured in the previous 5-minute interval as the estimate for the next 5-minute interval. Similarly, for GEANT, we use the traffic matrix measured in the previous 15-minute interval as the estimate for the next 15-minute interval.

We use $U(real)$ to denote the maximal link utilization when the TE method uses the real traffic demand, $U(estimated)$ the maximal link utilization when the TE method uses traffic matrix estimate, and $U(OSPF)$ the maximal link utilization under OSPF. The normalized inaccuracy of is defined in Equation (9):

$$Inaccuracy = \frac{U(estimated - U(real)}{U(OSPF) - U(real)} \times 100\% \qquad (9)$$

When $Inaccuracy = 0$, MCFTE using the estimated traffic matrix performs the same as MCFTE using the real traffic matrix. When $Inaccuracy = 100\%$, MCFTE with the estimated traffic matrix performs the same as pure OSPF routing. When $Inaccuracy > 100\%$, MCFTE is worse than the pure OSPF routing. Figure 7 shows a typical result using Abilene data at each hour on 2008-08-21. It is clear that MCFTE is robust in that the reduction of maximal link utilization is still significant when estimated traffic matrices are used.

To compare the robustness of MCFTE with SAMTE and IGPWO, we plot the CDF (Cumulative Distribution Function) of $Inaccuracy$ in Figure 8. In the sub-figures, for Internet2 and Abilene, we also draw the parts of the CDF curves where the TE method improves the traffic distribution ($Inaccuracy < 100\%$). For Internet2 (Figure 8(a)), the traffic matrices of every five minutes from 2008-08-21 through 2008-08-27 are considered, and MCFTE improves 94.24% of all the cases. For Abilene (Figure 8(b)), MCFTE improves for all the cases. For GEANT (Figure 8(c)), all the three TE methods improve the traffic distribution in all cases, and the majority of them are improved significantly. It has been observed [14] that the traffic in GEANT network has certain stability in that the link with the maximal link utilization does not change very often. The same links with the low capacity often gets the highest utilization which helps three TE methods to reduce maximal link utilization.

Except that in Figure 8(a) SAMTE shows comparable robustness, MCFTE is in general more robust than the other TE methods. For example, 97.42% points are under 10% $Inaccuracy$ in MCFTE on GEANT, while for SAMTE this number is 88.69% and for IGPWO only 33.93%. The IGPWO has the worst robustness among the three. For example, in Abilene, there are only 4.61% percent of points that improve the traffic distribution.

IV. Related Work

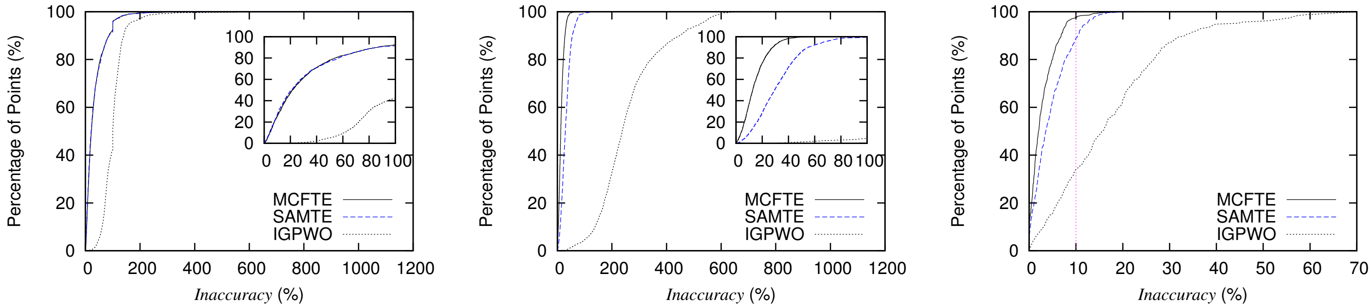
Weight optimization was first proposed by Fortz and Thorup [3] [2]. The problem was proved to be NP-hard and heuristic methods were used to search for solutions. Roughan *et al.* examined the robustness of weight optimization using the real topology and traffic matrices from a tier-1 ISP [1], where the traffic matrices were derived from link load data using techniques developed by Zhang *et al.* [11]. Wang *et al.* [7] proved that the optimal routing with respect to the objective of traffic engineering can always be achieved by shortest path routing under appropriate link weights. PEFT [17] is a scheme that sets link weights so that all the multi-commodity flows will follow the shortest paths. However, the change of the link weights still leads to network-wide routing convergence and traffic shift.

MATE [4] and TeXCP [5] work in similar fashion by splitting the traffic load among multiple MPLS paths, but they do not deal with how to establish these paths. They also need to frequently probe each paths for its congestion state. As a comparison, MCFTE gives both the LSPs need to be established and the split ratio between MPLS paths and OSPF paths. MCFTE does not need to explicitly probe the paths, but it needs the traffic matrix, which can be derived from link utilization data reported by OSPF-TE [18].

Hybrid routing uses both OSPF and MPLS to achieve traffic engineering goals and avoids the drawbacks of the both. It has been proposed and explored in previous work such as [6] [8] [9], but they all resorted to heuristics to find solutions. As we demonstrated in this paper, the problem actually can be formulated and solved through linear programming. MCFTE gives the optimal solution and runs faster than previous heuristics.

V. Conclusion

MCFTE formulates the traffic engineering problem as a linear programming multi-commodity flow problem, solves it for optimal solutions, and realizes it via hybrid OSPF/MPLS routing. It avoids network convergence and traffic shift caused by OSPF weight optimization, as well as the full-mesh tunnels required by pure MPLS approach. Compared with other hybrid routing schemes, MCFTE provides the optimal solution,



(a) Internet2 (b) Abilene (c) GEANT

Fig. 8. The CDFs of Inaccuracy

runs about two orders of magnitude faster, and is robust to measurement inaccuracy in traffic matrices. MCFTE could be deployed at multiple places in a network and invoked relatively frequently to respond to changes in traffic demand. Therefore MCFTE provides a good candidate for distributed, responsive traffic engineering solution in today's networks.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their comments. This work was supported by NSFC (60625201, 60873250), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20060003058) and 863 high-tech project (2007AA01Z216, 2007AA01Z468). This work was done during Mingui Zhang's visit at the University of Arizona when he was supported by the China Scholarship Council (2008621056) and partially by NSF award CNS-0721863.

REFERENCES

[1] M. Roughan, M. Thorup, and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices," in *Proceedings of ACM IMC*, 2003.
[2] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal On Selected Areas In Communications*, vol. 20, no. 4, pp. 756–767, May 2002.
[3] ——, "Internet Traffic Engineering by Optimizing OSPF Weights," in *Proceedings of IEEE INFOCOM*, 2000.
[4] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering," in *Proceedings of IEEE INFOCOM*, 2001.
[5] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the Tightrope: Responsive Yet Stable Traffic Engineering," in *Proceedings of ACM SIGCOMM*, 2005.
[6] F. Skivée, S. Balon, and G. Leduc, "A Scalable Heuristic for Hybrid IGP/MPLS Traffic Engineering - Case Study on an Operational Network," in *Proceedings of IEEE ICON*, 2006.
[7] Y. Wang, Z. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," in *Proceedings of IEEE INFOCOM*, 2001.
[8] H. Pham and B. Lavery, "Hybrid Routing for Scalable IP/MPLS Traffic Engineering," in *Proceedings of IEEE ICC*, 2003.
[9] E. Mulyana and U. Killat, "An Offline Hybrid IGP/MPLS Traffic Engineering Approach under LSP Constraints," in *Proceedings of the 1st International Network Optimization Conference, INOC*, 2003.
[10] H. Eiselt and C.-L. Sandblom, *Integer Programming and Network Models*. Germany: Springer, 2000.
[11] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads," in *Proceedings of ACM SIGMETRICS*, 2003.
[12] "Internet2 Netflow Data." [Online]. Available: http://www.internet2.edu/observatory/archive/data-collections.html
[13] "TOolbox for Traffic Engineering Methods (TOTEM)." [Online]. Available: http://totem.info.ucl.ac.be/
[14] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing Public Intradomain Traffic Matrices to the Research Community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, January 2006.
[15] "Traffic Matrices of Abilene." [Online]. Available: http://www.cs.utexas.edu/ yzhang/research/AbileneTM/
[16] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating Point-to-Point and Point-to-Multipoint Traffic Matrices: An Information-Theoretic Approach," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 947–960, 2005.
[17] D. Xu, M. Chiang, and J. Rexford, "Link-State Routing with Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering," in *Proceedings of IEEE INFOCOM*, 2008.
[18] D. Katz, K. Kompella, and D. Yeung, "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2," Sept 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3630.txt