WOSPF: A Traffic Engineering Solution for OSPF Networks

Aditya Mishra, Anirudha Sahoo[†], Bhavana Dalvi[‡], and Ting Zhu[§]
Seattle University mishraa@seattleu.edu
[†]IEEE Member axsahoo@yahoo.com
[‡]Allen Institute for Artificial Intelligence bhavanad@allenai.org
[§]University of Maryland Baltimore County zt@umbc.edu

Abstract—**Traffic engineering (TE) has long been used by network providers to reduce network congestion and improve resource utilization. Due to its significance, several traffic engineering algorithms have been proposed in literature. However, most of these algorithms optimize maximum link utilization (MLU) in network, and/or assume that network has the capability to route demands on arbitrary paths. Optimizing only for MLU can result in longer route computations to save bandwidth along shorter paths, thereby hurting application performance (as shown by recent research). Further, minimizing MLU can lead to solutions where several links have utilization close to MLU, while many others are under-utilized. Besides, as large fraction of today's Internet uses OSPF routing protocol, it cannot benefit from TE algorithms assuming arbitrary routing capabilities. To address these problems, we present Wise-OSPF (WOSPF), a traffic engineering solution for OSPF networks. WOSPF formulates TE as an optimization problem. The objective of WOSPF is to minimize the difference between the maximum and minimum link utilizations across the network, which leads to more uniform traffic distribution compared to optimizing MLU. As WOSPF uses OSPF for routing demands, it does not compute unnecessarily long routes and can be employed in legacy OSPF networks with minimal changes. Our results show that WOSPF reduces standard deviation of link utilizations in network by 31.35% compared to an optimal MLU based TE approach, while achieving an MLU within 1.9% of the optimal.**

I. INTRODUCTION

Traffic Engineering (TE), in general, attempts to shape and control Internet traffic to achieve specific network performance objectives [5]. Formally, the problem can be defined as: Given a network topology and an estimate of traffic matrix to be routed on it, the goal is to find a traffic routing scheme that optimizes the user perceived performance and makes efficient use of network resources. As traffic engineering helps reducing congestion hotspots and improving resource utilization (which, in turn, results in increased revenue collection and reduced costs), it has become an integral part of several large autonomous systems.

Due to its significance, several researchers have proposed quite a few TE algorithms in the recent past, e.g., [12], [17], [9], [20], [3], [6], [7]. Although these algorithms help reduce congestion and improve resource utilization in networks, many suffer from two limitations: First, most TE algorithms seek to optimize/minimize the MLU (Maximum Link Utilization)–link utilization of the most utilized link in the network; second, most TE approaches cannot be employed to OSPF based networks with minimal/no changes.

The TE schemes which optimize for MLU and predictability (e.g., [17]) tend to route traffic along longer paths to leave room for occasional traffic spikes along shorter paths, and reduce MLU. As user application performance is largely determined by propagation delays [8], [14], such TE optimizations hurt performance by choosing long routes (as shown in [15]). Another limitation of optimizing MLU is that it is largely determined by the traffic on only one link. As soon as the minimum MLU value is achieved, the TE algorithm has no further incentive to reduce/distribute traffic on links with lesser utilization. Further, this may lead to a solution in which several links have utilization equal or very close to the bottleneck link utilization—as the objective is only to minimize the MLU.

Besides, Open Shortest Path First (OSPF) is the most widely used intra-domain protocol in today's Internet [13]. As OSPF always forwards packets over the shortest route from source to destination, it is not flexible to accommodate most of the proposed TE solutions. Authors in [12] proposed Smart-OSPF (S-OSPF) to enable traffic engineering in today's networks with minimal changes to OSPF. However, S-OSPF too relies on minimizing MLU for traffic engineering, and hence is not immune to all its limitations mentioned above.

To support traffic engineering in OSPF networks and overcome MLU limitations, we propose Wise-OSPF (WOSPF)—an extension of S-OSPF. Given traffic demand in the network, WOSPF routes these demands such that the difference between the maximum and minimum of link utilizations across the network is minimized. WOSPF splits traffic demands only at the source node by "carefully" forwarding it to direct neighbors of the source (more about this later); from there the demand follows shortest path to the destination. We formulate this as an optimization problem. The solution of the optimization is used to route packets in the network with minimal changes to OSPF such that the optimization objective is achieved.

Experimental evaluation on real world network topologies show that WOSPF reduces standard deviation of link utilizations across the network by up to 31.35%, as compared to an optimal TE algorithm minimizing MLU (hence resulting in more uniform traffic distribution); further, MLU achieved by WOSPF is within 1.9% of the optimal TE algorithm.

Rest of the paper is organized as follows: We present the related work in Section II; our proposed traffic engineering solution is detailed in Section III; as solving the presented formulation in Section III is intractable, we present a heuristic algorithm in Section IV. Next, we present experimental results in Section V. Section VI concludes the paper.

II. RELATED WORK

Several traffic engineering approaches have been proposed in the recent past. Authors in [18] have modelled traffic engineering problem as an optimization problem with the objective of minimizing congestion and maximizing potential for traffic growth. They have proposed explicit routing as an effective way of improving network utilization. In [16] the authors propose techniques for setting the weights in OSPF/IS-IS network adaptively according to the traffic on links. The paper discusses why the routing used in the early ARPANET (Minimal- Delay Adaptive Routing (MDAR)) was not stable and proposes ways to make their Load-Sensitive Adaptive Routing (LSAR) stable. [7] proposes techniques for optimizing OSPF or ISIS weights for intradomain routing in a dynamic setting. Authors in [6] optimize the weight settings for the proposed AT&T WorldNet backbone based on the projected demands. The paper shows that finding the optimal weight settings for a given set of demands is NP-hard. Hence it proposes a local search heuristic, which performs very close to the optimal general routing for the proposed AT&T WorldNet backbone. However, the performance of this search heuristic on synthetic internetworks was not as close to the optimal general routing. In [4] the authors discuss the applications of MPLS to traffic engineering in IP networks. They also discuss an overlay model, based on IP over ATM. [10] presents an algorithm for dynamic routing of bandwidth guaranteed tunnels, where tunnel routing requests arrive at runtime, without assuming any knowledge about of requests. The algorithm makes use of the information regarding ingress-egress routers in the network. The key idea is to route the demand over a path that does not interfere too much with potential future LSP set-up requests between other source and destination pairs. Authors in [17] have looked at offline traffic engineering methods to optimize routing for unpredictable traffic demands. [11] presents a forward fault correction based TE approach that proactively protects a network from congestion and packet loss due to data and control plane faults.

Most of the previous TE work either optimizes OSPF link weights for routing or assumes freedom of routing demands over arbitrary paths in network. In contrast, we use OSPF as the underlying protocol for routing demands without changing/optimizing OSPF link weights. The closest work to our approach is presented in [12], where authors have proposed S-OSPF to enable traffic engineering in OSPF networks employing minimal changes to OSPF. However, S-OSPF's objective is to minimize the maximum link utilization in network, as opposed to WOSPF's, which minimizes the difference between the maximum and minimum of link utilizations in the network.

III. TRAFFIC ENGINEERING IN OSPF NETWORKS

A. Problem Formulation

The *traffic engineering* problem that we address here, can be stated as follows. Given the network topology and an estimate of traffic demand matrix to be routed on it, the problem is to find the fraction of each demand that needs to be carried by a link in the network such that the difference between the maximum of link utilization and the minimum of link utilization over all the links in the network is minimized. The linear programming formulation for this problem is such that, while minimizing the difference between maximum and minimum link utilizations across all the links in the network, the maximum of link utilization remains same as that achieved by S-OSPF.

IV. SYSTEM MODEL

We represent a network as a graph $G = (V, E)$, where V is the set of all nodes and E is the set of all links. For each $link(i, j)$, c_{ij} denotes the capacity of the link. The capacity $c_{ij} = 0$ if there is no link between nodes i and j. K denotes the set of all traffic demands between different pairs of source and destination nodes. For each demand $k \in K$, d_k, s_k, t_k denote the bandwidth demanded, the source node, and the destination node respectively. A demand may be split over multiple paths, with each path satisfying a fraction of the demand. For each $link(i, j) \in E$ and for each demand $k \in K$, let X_{ij}^k represent the percentage of bandwidth demand of d_k carried by $link(i, j)$. Let α represent the maximum of link utilization, and β represent the minimum of link utilizations across all the links. Let PATH(s_k, t_k) represent the ordered list of all nodes along the OSPF path from node s_k to t_k for traffic demand k. $OSPF\_ancestor_i^k$ represents all those nodes which are predecessors of node i in PATH(s_k, t_k). Similarly, $OSPF\_nexthop_i^k$ represents the OSPF nexthop of node i for the destination node t_k.

A. Wise-OSPF Based Traffic Engineering

We have formulated the LPF(Linear Programming Formulation) of WOSPF so as to avoid any loop formation and to ensure that the maximum of link utilization comes out to be same as that for S-OSPF. Following is the LPF.

$$minimize(\alpha - \beta) \tag{1}$$

$$X_{pq}^k - \sum_{j:(j,p) \in E} X_{jp}^k = 0, k \in K, p \neq s_k, p \neq t_k, \\ q = OSPF\_nexthop_p^k, j \neq q \tag{2}$$

$$\sum_{j:(i,j) \in E} X_{ij}^k = 1, k \in K, i = s_k, j \neq OSPF\_ancestor_i^k \tag{3}$$

$$\sum_{k \in K} d_k X_{ij}^k \leq c_{ij}\alpha, (i, j) \in E \tag{4}$$

$$\sum_{k \in K} d_k X_{ij}^k \geq c_{ij}\beta, (i, j) \in E, 0 \leq X_{ij}^k \leq 1, \alpha \geq 0 \tag{5}$$

The objective function in (1) says that the variable to be minimized is the difference between maximum and minimum of link utilization over all the links in the network. Constraints (2) and (3) are the flow conservation constraints. Constraint (2) represents the fact that the total traffic flowing into a node should be equal to the total traffic flowing out of the node for any node other than the source and the destination nodes of the demand; also, for an intermediate node, the incoming traffic can be from any of the neighboring nodes except the one to which there is outgoing traffic and outgoing traffic can be only to the OSPF next hop of the node for that demand's destination node. This constraint prevents packets from looping back and forth on the OSPF path thereby artificially increasing the link utilizations.

Constraint (3) says that the total flow out of the source node is 1, which is the total required normalized bandwidth of the traffic demand. We do not allow any incoming traffic for a demand to its source node, this is to prevent the formation of loops at source nodes.

Constraints (4) and (5) are the link capacity utilization constraints. (4) makes sure that the sum of all the fractions of traffic demands routed over a link should not exceed the maximum link utilization times the capacity of the link. Similarly, constraint (5) ensures that the sum of all the fractions of traffic demands routed over a link should not be less than the minimum link utilization times the capacity of the link.

Though the above traffic engineering formulation minimizes the maximum link utilization under stated constraints, it can split individual demands over multiple paths. If packets of the same flow are sent over different paths, then different delays may change ordering of packets in TCP flows and may lead to degradation of TCP performance.

Thus, it is desirable to route packets belonging to a particular demand over the same path. To achieve this goal we impose additional constraints on the LPF presented above. The additional constraint is that X_{ij}^k variables must be either 0 or 1 so that either the entire demand is put on a link or no part of the demand is put on the link. Following is the corresponding Integer Programming Formulation (IPF).

$$minimize(\alpha - \beta) \tag{6}$$

$$\begin{gathered} X_{pq}^k - \sum_{j:(j,p)\in E} X_{jp}^k = 0, k \in K, p \neq s_k, p \neq t_k, \\ q = OSPF\_nexthop_p^k, j \neq q \end{gathered} \tag{7}$$

$$\begin{gathered} \sum_{j:(i,j)\in E} X_{ij}^k = 1, k \in K, i = s_k, \\ j \neq OSPF\_ancestor_i^k \end{gathered} \tag{8}$$

$$\sum_{k\in K} d_k X_{ij}^k \leq c_{ij}\alpha, (i,j) \in E \tag{9}$$

$$\sum_{k\in K} d_k X_{ij}^k \geq c_{ij}\beta, (i,j) \in E \tag{10}$$

$$X_{ij}^k \in \{0,1\}, \alpha \geq 0 \tag{11}$$

Although the above IPF prevents packet reordering in TCP flows by routing each demand on a single path, it can be shown to be NP-Hard by reducing to **set-partition problem**. Hence we provide a heuristic solution for above formulation in section V.

B. WOSPF Example

Let us take an example to understand the difference between the traffic distributions generated by WOSPF, an general/optimal TE algorithm which minimizes MLU (presented in [18]), and S-OSPF (WOSPF's predecessor). Consider the network topology as shown in Figure 1(a). Assume that each link is bidirectional and has 10 units of capacity in each direction. There is an 18 units bandwidth demand from node S to destination node D and a demand of 6 units from node D to node S. Let us further assume that each link has a unit OSPF cost. Sink trees for $node\ S$ and $node\ D$ are shown in Figure 1(b) and 1(c) respectively.

The paths chosen by the optimal/general TE (presented in [18]) will be as follows. For demands from $node\ S$ to $node\ D$, 6 units of data will be sent directly from $S \rightarrow D$, another 6 units will be routed via path $S \rightarrow 1 \rightarrow 2 \rightarrow D$ and the rest 6 units will go on $S \rightarrow 3 \rightarrow 4 \rightarrow D$. The demand from $node\ D$ to $node\ S$ will be routed completely on link $D \rightarrow S$. As there are no OSPF ancestor nodes for nodes S and D in the sink trees of nodes D and S respectively, same paths will be chosen by S-OSPF also. So, maximum of link utilization for both these algorithms become 60%.

Let us now consider the case of our proposed WOSPF routing algorithm. Here also, a node can forward the traffic to all its neighbors, except to its OSPF ancestor nodes, for the given demand. Consider the sink trees for $node\ S$ and $node\ D$ shown in Figure 1(b) and 1(c) respectively. The paths and distributions chosen for routing demand from $node\ S$ to $node\ D$ will be the same as above. But the demand from $node\ D$ to S will be divided equally on paths $D \rightarrow 2 \rightarrow 1 \rightarrow S$, $D \rightarrow S$ and $D \rightarrow 4 \rightarrow 3 \rightarrow S$, i.e., 2 units will be sent on each of these paths. Again the maximum of link utilization across all the links remains 60%, but in order to minimize the difference between the maximum and minimum of link utilizations(which is 40% in this case), WOSPF distributes the traffic more uniformly.

C. Advantages of WOSPF Objective

In this section we argue that the WOSPF objective function is better than just minimizing the MLU. Following arguments are supported by the simulation results, given in the results section.

The traffic distribution generated by minimizing the MLU contains several links having utilization equal to or very close to bottleneck utilization, while several other links are either not utilized or are underutilized. Whereas WOSPF objective distributes the excessive traffic on bottlenecked links over the less utilized links to minimize the difference between α and

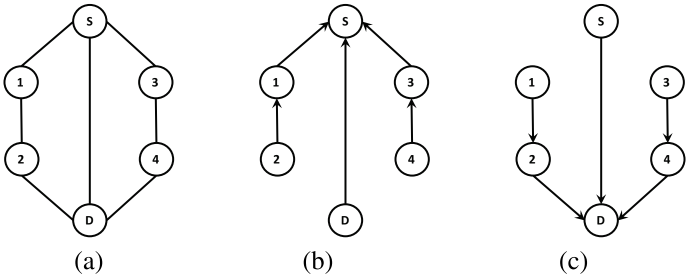


Fig. 1: (a) Traffic demands from S to D and D to S (b) Sink tree for node S (c) Sink tree for node D.

β. This distribution proves advantageous while rerouting the split demands and/or new traffic. As there are fewer bottleneck links in the network, the probability that the maximum of link utilization increases after rerouting the split/new demands is greatly reduced.

The traffic is distributed more uniformly by WOSPF. This is evident from significantly less standard deviation of link utilizations across network for WOSPF as compared to both S-OSPF and the optimal TE algorithm.

In most of the papers on Traffic Engineering(TE), the authors have proposed solutions to TE problems at two extremes: some authors assume that the demand matrix is well known in advance (eg. [19]), whereas others assume absolutely nothing about the demand matrix eg, in [10]. The actual scenario lies somewhere between. Significant amount of information about the traffic can be derived form SLAs(Service Level Agreements) and traffic monitoring and this information can be used as an input demand matrix. Still, there are considerable number of demands and bursty traffic arriving at the run time. In the above scenario using WOSPF for routing the known demands can be advantageous. Firstly, WOSPF will distribute the traffic uniformly and leave enough bandwidth on links to accommodate new demands without increasing the MLU in the network. Secondly, in no case the path chosen by WOSPF for a demand can be too long because the demands are split/routed only at the source nodes, from the next hop onwards demands follow the OSPF shortest path to the destination.

D. Loop Free Property of WOSPF

Looping is a major issue in any routing protocol. WOSPF is carefully designed such that packets do not loop in the steady state. In WOSPF, for a given demand (with a given destination), source node sends traffic to all its neighbors which are not OSPF ancestor in the sink tree rooted at that destination. Thus, the forwarding at source node ensures that it is loop free. From the next hop onwards, the packet of the given demand follows the OSPF path (no further splitting happens at those nodes). OSPF is a loop-free protocol. Thus, packets belonging to the given demand will follow a loop free path.

V. HEURISTIC SOLUTION

Since finding traffic engineering paths for traffic demands without bifurcating the demands in WOSPF is NP-hard, we propose heuristic solution for the IPF here. We define *permissible paths* for a demand k with source node s and destination node t as all those paths which start at node s go through a neighboring node n ($n \neq OSPF\_ancestor_s^k$) of s and follow the shortest path from s to t via node n.

Let X_{ij}^k and $\alpha - \beta$ be the results of the optimal LPF solution (given in (IV-A)). We define a subgraph U_L as a graph consisting only of source and destination nodes of demand L and all the nodes and links which lie on all the *permissible paths* from source node of demand L to destination node of demand L. Also let α_L and β_L be the maximum and minimum of link utilizations over all links in the subgraph U_L. We use the following algorithm for rerouting the split demands in distribution generated by WOSPF.

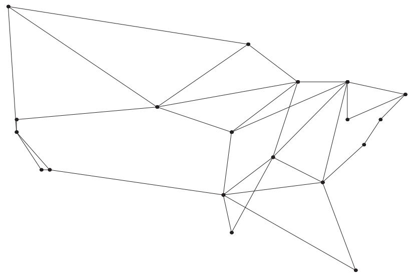
1) Solve LPF of section IV-A and get optimal solution for X_{ij}^k and $\alpha - \beta$.
2) Let S denote the set of all split demands.
3) Take out split demands in S from link bandwidth allocations. Recalculate the current load $l_{ij} = \sum_{k \in K-S} X_{ij}^k$ for each $link(i,j)$, and recompute $\alpha - \beta$.
4) Split demands are picked for rerouting in descending order of their sizes.
    a) Let L be largest demand in the set S.
    b) Among various *permissible paths*, from the source node of the demand L , route the demand over the path for which the value of $\alpha_L - \beta_L$ is minimum, after routing the demand over that path.
    c) Recompute the $\alpha - \beta$ value.
    d) Take off L from the set S.
    e) If S is not empty go to 4a.

VI. EVALUATION

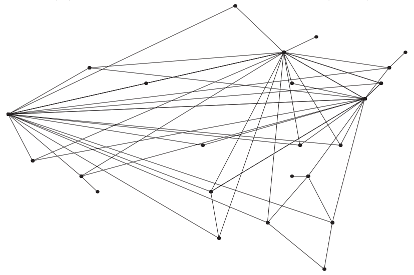
In this section, we present our simulation setup and performance comparison of WOSPF based traffic engineering with S-OSPF and general/optimal traffic engineering (general TE)—presented in [18]—and OSPF.

A. Simulation Setup and Notations

For performance evaluations we use network topologies of the Cable and Wireless network, and the GoodNet network available at [2]. Simulations were coded in Java and lp_solve 5.1.1.3 package([1]) was used for solving the linear programs. Cable and Wireless network has 20 nodes and 68 directional links (Figure 2a); GoodNet network consists of 27 nodes and 116 directional links (Figure 2b).



(a) Cable & Wireless N/W ([2])



(b) GoodNet Network ([2])

Fig. 2: Network Topologies

For the given network topologies, the link capacities were generated randomly with a uniform distribution in the range [800, 1200]. Total generated capacity of Cable and wireless network is 67386, and that of GoodNet is 113778 units. For Cable and wireless network, we simulate 14 network scenarios, with the number of demands ranging from 100 to 1400 in the increments of 100; the network saturates at 1400 demands for the OSPF routing, as indicated by its MLU greater than 1 hence we stop here. For GoodNet Network, we simulate 34 network demand scenarios, with the number of demands ranging from 100 to 3400 in the increments of 100; the GoodNet network saturates at 3400 demands for the OSPF routing. For all simulation runs, size of each demand is chosen randomly from uniform distribution in the range [1,10]. Source and destination nodes are also selected randomly among the nodes in network topology and we make sure that each pair contains distinct source and destination nodes.

We experiment and compare results for OSPF, S-OSPF, WOSPF, and an optimal traffic engineering algorithm presented in [18]; we refer to the algorithm in [18] as optimal TE or general TE because for a given network and traffic demands, it is guaranteed to find a routing with minimum/optimal MLU. For performance comparison between these algorithms, we have used maximum of link utilization, minimum of link utilization, standard deviation of link utilizations, and the mean of link utilizations across all the links in the network as metrics. For WOSPF, we present results obtained from simulations run over Cable and Wireless network and GoodNet network. The total demand corresponding to the number of demands for simulations done on Cable and Wireless network ranges from 568 units to 7631 units. As WOSPF routing can result in some demands being split across different paths, we use the heuristic presented in Section V for rerouting the split demands.

We have used the following notations in the tables. α denotes the maximum of link utilization values. β stands for the minimum of link utilization over all the links in the network. δ is used for representing the standard deviation of link utilizations of all the links in the network. Δ represents the difference between the maximum and minimum of link utilizations in the network, i.e., $\alpha - \beta$. The subscripts denote TE algorithm and superscripts, when present, indicates whether the solution allowed individual demands to be split among multiple paths (*split*), or not (*no-split*). *opt* in subscripts denotes the optimal TE algorithm from [18], whereas *sospf, wospf, ospf* denote the respective algorithms named. Note that columns for *opt, sospf, ospf* algorithms have no superscripts in the tables. In the presented results *opt, sospf* split individual demands for optimizing their objectives, however *ospf* never splits any. Wherever WOSPF is referred without split/no-split, we imply its *split* version.

B. WOSPF Results

Does WOSPF reduce congestion in network? Tables I, II compare the maximum link utilization (MLU) achieved by WOSPF against S-OSPF, OSPF, and optimal TE for Cable and Wireless, and GoodNet networks respectively. Clearly the MLU from WOSPF closely follows the MLU achieved by optimal TE formulation, and is similar to S-OSPF's MLU: On average, in Cable and Wireless network, WOSPF with split demands has an MLU only 1.88% greater than the optimal. Note the MLU values obtained by $\alpha_{wospf}^{no-split}$ after rerouting the split demands from WOSPF, is not very different from the optimal MLU either, in Table I $\alpha_{wospf}^{no-split}$ is only 3.19% greater than the optimal. In contrast, MLUs produced by OSPF are much higher than the other three approaches. For Table I, on average, WOSPF achieves an MLU 24.1% less than the OSPF. Table II shows similar results.

Does WOSPF distribute traffic more uniformly across network? Tables III and IV compare standard deviation of link utilizations (about mean) for WOSPF with the optimal TE, S-OSPF, and OSPF on Cable and Wireless, and GoodNet networks. The results show that WOSPF achieves significantly less deviation in link utilizations compared to all other approaches, hence achieving more uniform traffic distribution. For Cable and Wireless network (Table III), on average, WOSPF exhibits 31.35% less deviation than optimal TE, 29.36% less deviation than S-OSPF, and 23.36% less deviation than OSPF. Even the WOSPF with no split demands (on average) has 30.92% less deviation than the optimal TE solution. Similar results are seen for GoodNet network as well (Table IV).

The difference between the maximum and minimum link utilizations achieved by WOSPF (Table VI) for all demands further evinces its uniform traffic distribution. The difference achieved by WOSPF is 28.29% less than the difference from the optimal TE, and 30.16% less than the difference from OSPF. On average, the difference from WOSPF with no splits is 22.04% less than the optimal TE value. Differences from S-OSPF closely follow the corresponding values from optimal

No. of demands	α_{opt}	α_{sospf}	α^{split}_{wospf} (No. of split demands)	$\alpha^{no-split}_{wospf}$	α_{OSPF}
100	0.0827	0.0978	0.0978 (17)	0.1066	0.1173
200	0.1033	0.1182	0.1033 (20)	0.1048	0.1751
300	0.1868	0.2124	0.1868 (20)	0.1912	0.2578
400	0.2272	0.2272	0.2272 (16)	0.2346	0.2693
500	0.3183	0.3253	0.3183 (19)	0.3378	0.4498
600	0.3079	0.3191	0.3191 (19)	0.3191	0.4507
700	0.3948	0.4116	0.4116 (22)	0.4213	0.5404
800	0.3964	0.4196	0.4196 (21)	0.4377	0.5902
900	0.5058	0.5058	0.5058 (27)	0.5058	0.6347
1000	0.5519	0.5956	0.5956 (24)	0.5955	0.7644
1100	0.7288	0.7288	0.7288 (23)	0.7288	0.8373
1200	0.6743	0.6764	0.6764 (20)	0.6764	0.9653
1300	0.7079	0.7079	0.7079 (26)	0.7217	0.9253
1400	0.8387	0.8387	0.8387 (26)	0.8387	1.1084

TABLE I: WOSPF: Comparison of Maximum of link utilizations over all links of Cable and Wireless network

TE. Due to space constraints, only the results from Cable and Wireless network are reported in Table VI, but in our experiments we found similar results for GoodNet network.

Does WOSPF route traffic on under-utilized links? Table V shows the utilization of the least utilized link for optimal TE, SOSPF, WOSPF, and OSPF. The table shows that utilization of the least utilized link for traffic engineering approaches that optimize MLU—optimal TE, and SOSPF—is mostly zero. This happens because MLU is determined largely by one link; as soon as the optimal MLU is achieved, the TE algorithm has no incentive to further distribute traffic over under-utilized links. This may easily lead to a solution where several links have utilization close to MLU and several other links with zero utilization (as in Table V). In contrast, with WOSPF utilization of the least utilized links increases rapidly with increasing demand. As WOSPF aims to minimize the difference between the most utilized and least utilized links in the network, it has incentive to route traffic on under-utilized links to bridge the gap, thereby creating more uniform traffic distribution. From Table V, on average, WOSPF increases the utilization of least utilized link by a whooping 72,344.44% compared to optimal TE, and 667.06% compared to OSPF.

VII. CONCLUSION

We reported an effective algorithm, called Wise-OSPF (WOSPF) to provide traffic engineering in OSPF based networks. We formulated WOSPF as an optimization problem to minimize the difference between the maximum and minimum of link utilizations over all links in the network. We evaluated WOSPF on network topologies from two service providers. Simulation results showed that WOSPF outperformed the optimal TE solution in terms of standard deviation and minimizing the difference between the maximum and minimum of link utilizations across the network, hence producing more uniform traffic distribution. Additionally, in terms of minimizing the maximum of link utilization, WOSPF very closely follows the optimal MLU based TE solution.

REFERENCES

[1] lpsolve. http://lpsolve.sourceforge.net/5.1/.

No. of Demands (Total Demand)	α_{opt}	α_{sospf}	α^{split}_{wospf} (No. of split demands)	$\alpha^{no-split}_{wospf}$	α_{OSPF}
100 (552)	0.0344	0.0344	0.0344 (17)	0.0397	0.0477
200 (1101)	0.0781	0.0781	0.0781 (28)	0.0781	0.0807
300 (1599)	0.0917	0.0917	0.0917 (35)	0.0988	0.1159
400 (2237)	0.1437	0.1437	0.1437 (50)	0.1437	0.1466
500 (2800)	0.1375	0.1375	0.1375 (37)	0.1725	0.1914
600 (3422)	0.2069	0.2069	0.2069 (50)	0.2193	0.2384
700 (3797)	0.2268	0.2268	0.2268 (50)	0.2268	0.2268
800 (4224)	0.2253	0.2253	0.2253 (53)	0.2644	0.3389
900 (4996)	0.1924	0.1924	0.1985 (53)	0.2100	0.3345
1000 (5530)	0.3045	0.3045	0.3045 (53)	0.3045	0.3321
1100 (6080)	0.3035	0.3035	0.3035 (60)	0.3827	0.3529
1200 (6548)	0.3767	0.3767	0.3767 (56)	0.3767	0.3767
1300 (7172)	0.4077	0.4077	0.4077 (59)	0.4077	0.4144
1400 (7658)	0.3866	0.3866	0.3866 (46)	0.4239	0.4987
1500 (8326)	0.4387	0.4387	0.4387 (56)	0.4387	0.5053
1600 (8768)	0.4151	0.4151	0.4151 (63)	0.4647	0.4943
1700 (9270)	0.4585	0.4585	0.4585 (59)	0.4733	0.5196
1800 (9821)	0.4820	0.4820	0.4820 (30)	0.4820	0.5196
1900 (10595)	0.5229	0.5229	0.5229 (52)	0.5350	0.6146
2000 (11088)	0.5093	0.5093	0.5093 (54)	0.6358	0.6916
2100 (11779)	0.6543	0.6543	0.6543 (58)	0.7088	0.6543
2200 (12128)	0.6022	0.6022	0.6022 (58)	0.6022	0.7070
2300 (12641)	0.6097	0.6097	0.6097 (61)	0.6097	0.6664
2400 (13217)	0.6753	0.6753	0.6753 (53)	0.6791	0.7948
2500 (13961)	0.5689	0.5689	0.5689 (54)	0.5689	0.8402
2600 (14377)	0.6828	0.6828	0.6828 (44)	0.6924	0.8007
2700 (14898)	0.7237	0.7237	0.7237 (49)	0.7237	0.8762
2800 (15290)	0.8340	0.8340	0.8340 (61)	0.9963	0.8490
2900 (16101)	0.7670	0.7670	0.7670 (56)	0.7670	0.9315
3000 (16501)	0.8327	0.8327	0.8327 (66)	0.8327	0.9371
3100 (16780)	0.7175	0.7175	0.7175 (64)	0.8136	0.9517
3200 (17429)	0.8848	0.8848	0.8848 (66)	0.9467	1.0939
3300 (18044)	0.8364	0.8364	0.8364 (58)	0.8364	1.0764
3400 (18442)	0.9009	0.9009	0.9009 (60)	0.9009	1.068

TABLE II: WOSPF: Comparison of Maximum of link utilizations over all links of GoodNet network

No. of demands	δ_{opt}	δ_{sospf}	δ^{split}_{wospf}	$\delta^{no-split}_{wospf}$	δ_{ospf}
100	0.0301	0.0256	0.0211	0.0209	0.0237
200	0.0331	0.0376	0.0276	0.0262	0.0398
300	0.0588	0.0668	0.0423	0.0425	0.0502
400	0.0789	0.0721	0.0600	0.0587	0.0611
500	0.1055	0.0978	0.0766	0.0739	0.0933
600	0.0943	0.1029	0.0806	0.0779	0.1016
700	0.1266	0.1269	0.0798	0.0798	0.1175
800	0.1151	0.1304	0.0998	0.1033	0.1202
900	0.1748	0.1531	0.0918	0.0947	0.1372
1000	0.1633	0.1885	0.1120	0.1116	0.1664
1100	0.2579	0.2045	0.1588	0.1532	0.1878
1200	0.2114	0.2057	0.1468	0.1580	0.1924
1300	0.2224	0.2138	0.1419	0.1481	0.2059
1400	0.2616	0.2540	0.1878	0.1869	0.2346

TABLE III: WOSPF: Comparison of Standard Deviations of link utilizations of Cable and Wireless network

No. of demands	δ_{opt}	δ_{sospf}	δ_{wospf}^{split}	$\delta_{wospf}^{no-split}$	δ_{ospf}
100	0.0135	0.0112	0.0093	0.0092	0.0104
200	0.0286	0.0203	0.0160	0.0156	0.0180
300	0.0349	0.0278	0.0206	0.0199	0.0251
400	0.0538	0.0383	0.0267	0.0263	0.0329
500	0.0531	0.0416	0.0332	0.0343	0.0408
600	0.0761	0.0553	0.0371	0.0363	0.0473
700	0.0849	0.0590	0.0408	0.0422	0.0501
800	0.0819	0.0637	0.0375	0.0392	0.0570
900	0.0509	0.0594	0.039	0.0391	0.0661
1000	0.1138	0.0788	0.0566	0.0563	0.0720
1100	0.1102	0.0881	0.0586	0.0603	0.0811
1200	0.1356	0.0958	0.0580	0.0597	0.0811
1300	0.1476	0.1061	0.0694	0.0704	0.0926
1400	0.1380	0.1093	0.0753	0.0772	0.1042
1500	0.1596	0.1227	0.0751	0.0742	0.1077
1600	0.1438	0.1221	0.074	0.0763	0.1094
1700	0.1628	0.1292	0.0801	0.0815	0.1151
1800	0.1759	0.1412	0.1046	0.1028	0.1156
1900	0.1865	0.1500	0.1039	0.1011	0.1346
2000	0.1696	0.1483	0.097	0.1027	0.1401
2100	0.2470	0.1766	0.1024	0.1036	0.1474
2200	0.2029	0.1736	0.0960	0.0973	0.1526
2300	0.2023	0.1730	0.1047	0.1030	0.1531
2400	0.2478	0.1860	0.1217	0.1227	0.1734
2500	0.1785	0.1791	0.1017	0.1035	0.1745
2600	0.2413	0.1917	0.1401	0.1387	0.1780
2700	0.2563	0.2158	0.1234	0.1216	0.1835
2800	0.2950	0.2240	0.1517	0.1573	0.1942
2900	0.2686	0.2295	0.1295	0.1309	0.1972
3000	0.3004	0.2299	0.1460	0.1437	0.2027
3100	0.2341	0.2172	0.121	0.1225	0.1984
3200	0.3265	0.2423	0.1415	0.1436	0.2153
3300	0.3144	0.2542	0.1521	0.1484	0.2264
3400	0.3352	0.2546	0.1634	0.1636	0.2298

TABLE IV: WOSPF: Comparison of Standard Deviations of link utilizations of GoodNet network

No. of demands	β_{opt}	β_{sospf}	β_{wospf}^{split}	$\beta_{wospf}^{no-split}$	β_{ospf}
100	0.0000	0.0000	0.0100	0.0037	0
200	0.0000	0.0000	0.0285	0.0245	0
300	0.0000	0.0000	0.0401	0.0299	0
400	0.0000	0.0000	0.0526	0.04089	0.0054
500	0.0000	0.0000	0.0752	0.0559	0.0098
600	0.0025	0.0000	0.0894	0.0789	0.0173
700	0.0000	0.0000	0.1358	0.1128	0.0000
800	0.0000	0.0000	0.1287	0.0840	0.0111
900	0.0000	0.0000	0.1833	0.1650	0.0261
1000	0.0000	0.0000	0.2021	0.1860	0.0272
1100	0.0000	0.0000	0.1771	0.1423	0.0315
1200	0.0000	0.0000	0.1997	0.1657	0.0290
1300	0.0000	0.0000	0.2573	0.2228	0.0448
1400	0.0000	0.0000	0.2456	0.2120	0.0359

TABLE V: WOSPF: Comparison of Minimum of link utilizations over all links of Cable and Wireless network

No. of demands	Δ_{opt}	Δ_{sospf}	Δ_{wospf}^{split}	$\Delta_{wospf}^{no-split}$	Δ_{ospf}
100	0.0827	0.0978	0.0956	0.10295	0.1173
200	0.1033	0.1182	0.0749	0.0804	0.1751
300	0.1868	0.2124	0.1467	0.1612	0.2578
400	0.2272	0.2272	0.1746	0.1937	0.2639
500	0.3183	0.3253	0.2431	0.2818	0.4399
600	0.3054	0.3191	0.2297	0.2401	0.4333
700	0.3948	0.4116	0.2757	0.3085	0.5404
800	0.3964	0.4196	0.2909	0.3536	0.5790
900	0.5058	0.5058	0.3224	0.3408	0.6086
1000	0.5519	0.5956	0.3934	0.4095	0.7373
1100	0.7288	0.7288	0.5517	0.5865	0.8058
1200	0.6743	0.6764	0.4768	0.5107	0.9364
1300	0.7079	0.7079	0.4505	0.4989	0.8806
1400	0.8387	0.8387	0.5931	0.6268	1.0726

TABLE VI: WOSPF: Comparison of Difference between Maximum and Minimum of link utilizations over all links of Cable and Wireless network

[2] Mapnet. http://www.caida.org/tools/visualization/mapnet/Backbones/, 2007.

[3] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 313–324, New York, NY, USA, 2003. ACM.

[4] D. Awduche. MPLS and traffic engineering in IP networks. *IEEE Communications Magazine*, 37(12):42–47, 1999.

[5] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. RFC 2702 (Informational), Sept. 1999.

[6] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. Proceedings of the IEEE INFOCOM, 2000.

[7] 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.

[8] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot. Packet-level traffic measurements from the sprint ip backbone. *Network, IEEE*, 17(6):6–16, 2003.

[9] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the Tightrope: Responsive Yet Stable Traffic Engineering. In *ACM SIGCOMM*, Philadelphia, PA, August 2005.

[10] M. S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *INFOCOM*, pages 884–893, 2000.

[11] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter. Traffic engineering with forward fault correction. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 527–538, New York, NY, USA, 2014. ACM.

[12] A. K. Mishra and A. Sahoo. S-OSPF: A traffic engineering solution for ospf based best effort networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 1845–1849. IEEE, 2007.

[13] J. Moy. OSPF Version 2. RFC 2328 (Standard), Apr. 1998.

[14] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot. Measurement and analysis of single-hop delay on an ip backbone network. *Selected Areas in Communications, IEEE Journal on*, 21(6):908–921, 2003.

[15] A. Sharma, A. Mishra, V. Kumar, and A. Venkataramani. Beyond mlu: An application-centric comparison of traffic engineering schemes. In *INFOCOM, 2011 Proceedings IEEE*, pages 721–729. IEEE, 2011.

[16] H. Wang and M. R. Ito. Dynamics of load sensitive adaptive routing. Proceedings of the IEEE International Coference on Communications (ICC), 2005.

[17] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. Cope: Traffic engineering in dynamic networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 99–110, New York, NY, USA, 2006. ACM.

[18] Y. Wang and Z. Wang. Explicit routing algorithms for internet traffic engineering. Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN), 1999.

[19] Z. Wang. *Internet QoS, Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publishers, 2001.

[20] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley. On optimal routing with multiple traffic matrices. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 607–618. IEEE, 2005.