# S-OSPF: A Traffic Engineering Solution for OSPF based Best Effort Networks

Aditya Kumar Mishra and Anirudha Sahoo
Kanwal Rekhi School of Information Technology
Indian Institute of Technology Bombay, Powai, Mumbai - 400076, India
Email: {adityam, sahoo}@it.iitb.ac.in

***Abstract*—Open Shortest Path First (OSPF) is one of the most widely used intra-domain routing protocol. It is well known that OSPF protocol does not provide flexibility in terms of packet forwarding to achieve any network optimization objective. Because of the high cost of network assets and commercial and competitive nature of Internet service provisioning, service providers are interested in performance optimization of their networks. This helps in reducing congestion hotspots and improving resource utilization across the network, which, in turn, results in an increased revenue collection. One way of achieving this is through *Traffic Engineering*. Currently traffic engineering is mostly done by using MPLS. But legacy networks running OSPF would need to be upgraded to MPLS. To achieve better resource utilization without upgrading OSPF network to MPLS is a challenge. In this paper we present a simple but effective algorithm, called Smart OSPF (S-OSPF) to provide traffic engineering solution in an OSPF based best effort network. We formulate an optimization problem based on the traffic demand to minimize the maximum link utilization in the network. Routing of the traffic demand is achieved using OSPF. We have simulated S-OSPF on real networks of two service providers. Simulation results show that S-OSPF based traffic engineering solution performance very closely follows the optimal solution.**

## I. Introduction

The Internet mostly offers best-effort service i.e. it tries its best to forward traffic, but cannot provide any guarantees in terms of bandwidth, latency and packet loss. This type of service is acceptable for some legacy applications like FTP and email, but not suitable for real-time applications such as Internet Telephony, Video conferencing and Video on Demand. The requirements of these applications cannot be met by best-effort services without Quality of Service (QoS) support. This is a major motivation behind the study of QoS routing algorithms. One of the ways of providing QoS is through Traffic Engineering. *Traffic Engineering* is concerned with the performance optimization of operation of networks [1]. Its main objective is to reduce congestion hot spots and improve resource utilization across the network through careful management of traffic distribution inside a network. Hence traffic engineering helps in minimizing packet loss and delay, and maximizing throughput. In general, traffic engineering encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic, and the application of such knowledge and techniques to achieve specific performance objectives [2]. The problem can be defined as follows. Given the network topology of the network and an estimate of the traffic matrix to be routed on it, the problem is to find a routing scheme that optimizes the network in terms of user performance and efficient use of network resources [3].

Traffic Engineering has become an integral part of many large Autonomous Systems, mainly because of the high cost of network assets and the commercial and competitive nature of the Internet. Also there is mounting pressure on service providers to meet customer's expectation in terms of delay, throughput and packet losses. Traffic engineering helps in reducing congestion hotspots and improving resource utilization across the network, which, in turn, results in an increased revenue collection. In today's Internet, Open Shortest Path First (OSPF) is one of the most widely used intra-domain routing protocols [4]. It is well known that OSPF protocol is not flexible enough to facilitate traffic engineering. This is because OSPF always forwards packet through shortest path. Even when the shortest path is congested and there is an alternate less congested path it does not have the capability to reroute packets through the alternate path. Multi Protocol Label Switching (MPLS) is a well known technology which can be used for traffic engineering [2]. But legacy networks running OSPF would need to be upgraded to MPLS. To achieve better resource utilization without upgrading OSPF network to MPLS is a challenge. In this paper, we propose a method of deploying traffic engineering solution in an OSPF based best effort network. We assume that the traffic demand in the network is known. These traffic demands should be distributed across different paths in the network so that the maximum of link utilization over all links in the network is minimized. We formulate this as an optimization problem. The solution of the optimization problem is used to route packets in the network with the help of OSPF such that the optimization objective is achieved.

The rest of the paper is organized as follows. We present some related literature of this study in Section II. The details of our proposed traffic engineering solution based on OSPF is presented in Section III. Since getting an optimal solution is an intractable problem, we propose a heuristic algorithm in Section IV. Some experimental results are presented in

This research was partially supported by IRCC, IIT Bombay under grant number 03IR059.


1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

Section V. Finally, we conclude our paper in Section VI.

## II. Related Work

There are many research work reported in traffic engineering. In [5] the author discusses the applications of MPLS to traffic engineering in IP networks. An overlay model, based on IP over ATM is also discussed. The authors in [6] consider explicit routing as an effective way of improving network utilization. They have modelled traffic engineering problem as an optimization problem with the objective of minimizing congestion and maximizing potential for traffic growth, i.e., the objective function minimizes the maximum of link utilization over all the links across the network. In [7] the authors propose a way of changing the weights in OSPF/IS-IS network adaptively according to the change of traffic load on the links. The paper discusses why Minimal-Delay Adaptive Routing (MDAR), the routing used in the early ARPANET, was not stable and proposes some techniques to make their Load-Sensitive Adaptive Routing (LSAR) stable. The objective of the paper in [8] was to optimize the weight settings for the proposed AT&T WorldNet backbone based on the projected demands. This 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. When tested on synthetic internetworks the performance of this search heuristic was not as close to the optimal general routing. In [9] authors propose techniques for optimizing OSPF or IS-IS weights for intradomain routing in a dynamic setting, by changing as few weights as possible. An algorithm for dynamic routing of bandwidth guaranteed tunnels, where tunnel routing requests arrive at runtime and there is no knowledge about future requests, is presented in [10]. This 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.

## III. Traffic Engineering in OSPF Network

### A. Problem Formulation

The general *traffic engineering* problem 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 maximum of link utilization over all the links in the network is minimized. The traffic demand matrix can be obtained by measurement of flow between source and destination nodes or can be specified by customers as their bandwidth requirements in the Service Level Agreements (SLAs). For a given link once the fraction of a traffic demand to be routed on the link is known, a flexible routing scheme like MPLS is required to achieve the goal. Most of the IP routing protocols do not have this capability to be flexible in routing packets in any path. For example, in OSPF routing protocol, which is the most widely used intra-domain routing protocol, the traffic demand can be routed only over the shortest path from the source to the destination. In our proposed Smart OSPF (S-OSPF) algorithm, a source node can potentially forward the flows to all its neighbors except to the neighbors which are its predecessors in the sink tree[1] of the given destination node. Splitting of traffic demand is done only at the source nodes (originating nodes of the demands). From next hop onwards demands are routed over the OSPF path to the destination node.

### B. System Model

We represent a network as a graph $G = (V, E)$, where $V$ is the set of nodes and $E$ is the set of links. For each $link(i, j)$, let $c_{ij}$ be the capacity of the link. We assume that the capacity $c_{ij} = 0$ if there is no link between nodes $i$ and $j$. Let $K$ be the set of all traffic demands between different pairs of source and destination nodes. For each $k \in K$, let $d_k$, $s_k$, $t_k$ be the bandwidth demand, 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 $\alpha$ represent the maximum of link utilization among all the links. Let $PATH(s_k, t_k)$ represent the ordered list of all nodes along the OSPF path from $s_k$ 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 OSPF nexthop of node $i$ for the destination node $t_k$.

### C. General Traffic Engineering Formulation

The general traffic engineering problem, assuming the total flexibility of splitting the demands between source and destinations, is as follows [6].

$$minimize \quad \alpha \quad such \quad that \tag{1}$$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 0 \qquad k \in K, i \neq s_k, i \neq t_k \tag{2}$$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 1 \qquad k \in K, i = s_k \tag{3}$$

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

$$0 \leq X_{ij}^k \leq 1, \alpha \geq 0 \tag{5}$$

The objective function in (1) says that the variable to be minimized is the maximum link utilization across all the links. Constraints (2) and (3) are the flow conservation constraints. Constraint (2) represents the fact that the traffic flowing into a node must be equal to the traffic flowing out of the node for

[1] A sink tree rooted at a node of a graph is the union of the shortest paths from all other nodes to that particular node.


1930-529X/07/$25.00 © 2007 IEEE

any node other than the source node and the destination node for each demand. Constraint (3) says that the net flow out of the source node is 1, which is the total required normalized bandwidth of the traffic demand. Constraint (4) is the link capacity utilization constraint. It makes sure that the sum of the fractions of traffic demands routed over a link should not exceed the maximum link utilization times the total capacity of the link.

### D. Smart-OSPF Based Traffic Engineering

But the generic traffic engineering optimization stated in the previous section cannot be applied to an OSPF network, since OSPF does not provide much flexibility in terms of routing of packets. Hence the optimization formulation has to be suitably changed so that the solution can be applied to an OSPF network. We refer to this solution as S-OSPF. The main idea in this solution is to split traffic demand only at the source node. And the source nodes makes sure that it does not forward any traffic to its ospf ancestors $OSPF\_ancestor^k_{s_k}$ for any demand $k$. Intermediate nodes route packets along OSPF path. The traffic engineering formulation can be modified for S-OSPF as a Linear Programming Formulation (LPF) as follows.

$$minimize \quad \alpha \quad such \quad that \tag{6}$$

$$X^k_{pq} - \sum_{j:(j,p)\in E} X^k_{jp} = 0 \qquad k \in K, p \neq s_k, p \neq t_k, \quad q = OSPF\_nexthop^k_p \tag{7}$$

$$\sum_{j:(i,j)\in E} X^k_{ij} - \sum_{p:(p,i)\in E} X^k_{pi} = 1 \qquad k \in K, i = s_k, \quad j \neq OSPF\_ancestor^k_i \tag{8}$$

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

$$0 \leq X^k_{ij} \leq 1, \alpha \geq 0 \tag{10}$$

The main changes are in (7) and in (8). (7) puts the constraint on intermediate nodes to only route packets along the OSPF path. In (8) it is made sure that fraction of the traffic demand $k$ is not forwarded to OSPF ancestor by the source node $s_k$. This makes sure that packets do not loop.

The solution to the LPF equations will produce the best routes for traffic demands between all source destination pairs. The problem with this solution is that individual demands may be split over multiple paths. If packets of the same flow are sent over different paths, then different delay may change ordering of packets in TCP flows. This may, in turn, 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 $X^k_{ij}$ variables must be either 0 or 1 so that either the entire demand is put on a link or no demand is put on the link. Following is the corresponding Integer Programming Formulation (IPF).

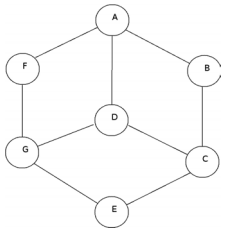


Fig. 1. Example topology: Traffic demand from A to B

$$minimize \quad \alpha \quad such \quad that \tag{11}$$

$$X^k_{pq} - \sum_{j:(j,p)\in E} X^k_{jp} = 0 \qquad k \in K, p \neq s_k, p \neq t_k, \quad q = OSPF\_nexthop^k_p \tag{12}$$

$$\sum_{j:(i,j)\in E} X^k_{ij} - \sum_{p:(p,i)\in E} X^k_{pi} = 1 \qquad k \in K, i = s_k, \quad j \neq OSPF\_ancestor^k_i \tag{13}$$

$$\sum_{k\in K} d_k X^k_{ij} \leq c_{ij}\alpha \qquad (i,j) \in E \tag{14}$$

$$X^k_{ij} \in \{0,1\}, \alpha \geq 0 \tag{15}$$

But solving the above IPF is a NP-hard problem. *Set-partition* problem is a well known NP-complete problem [11]. The set-partition problem takes a set $S_n$ of numbers as input. The question is whether the numbers can be partitioned into two sets $S1$ and $S2 = S_n - S1$ such that $\sum_{x\in S1} x = \sum_{x\in S2} x$. It can be proved that the above IPF is NP-Hard by reducing set-partition problem to it. Details of this proof can be found in [12]. We provide a heuristic solution to this IPF in Section IV.

### E. An Example

Let us take an example to understand, how do the different underlying routing algorithms affect the paths chosen by traffic engineering algorithms. Consider the network topology as shown in Figure 1. Assume that each link has 10 units of capacity and that there is a demand of 6 units of bandwidth from source node $A$ to destination node $B$. For simplicity, we assume that there is only one demand in the network. Let us further assume that each link has unit OSPF cost.

Now, if we have to use OSPF protocol for routing this demand, then the only path that can be chosen is the $link(A, B)$, because this is the shortest path from $nodeA$ to $nodeB$. This will make the maximum of link utilization, in the network, equal to $60\%$ (on link $AB$).

But if the underlying routing protocol provides complete flexibility in terms of packet routing, then the demand from node $A$ to node $B$ can be split as follows. 3 units can be sent directly on $link(A, B)$, two units can be sent over path $A \rightarrow D \rightarrow C \rightarrow B$, and the remaining one unit over $A \rightarrow F \rightarrow G \rightarrow E \rightarrow C \rightarrow B$. This traffic distribution gives us a maximum of link utilization of $30\%$ (on link $BC$ and $AB$).


1930-529X/07/$25.00 © 2007 IEEE

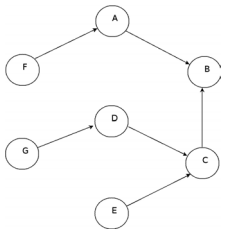


Fig. 2. Sink tree rooted at node B

Let us now consider the case of our proposed S-OSPF routing algorithm. Here a node can forward the traffic to all its neighbors, except to its OSPF ancestor nodes, for the given demand. Consider the sink tree for $node\ B$ shown in Figure 2. $Node\ F$ is the parent node of $node\ A$. So, $node\ A$ cannot forward any traffic, which is destined to $node\ B$, to $node\ F$. Keeping the above constraint in mind, the demand from $node\ A$ to $node\ B$ can be split in the following manner. Half of the demand goes $A \rightarrow D \rightarrow C \rightarrow B$ and the other half goes over $A \rightarrow B$. This splitting, again, gives us a maximum of link utilization of $30\%$ (on link $BC$ and $AB$).

### F. Loop Free Property of S-OSPF

Looping is a major issue in any routing protocol. S-OSPF is carefully designed such that packets do not loop in the steady state. In S-OSPF, 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.

## IV. Heuristic Solution

Since finding traffic engineering paths for traffic demands without bifurcating the demands is NP-hard we provide a heuristic solution to the problem here. Let $X_{ij}^k$ and $\alpha$ be the results of the optimal LPF solution (given in (6)). We use the following algorithm for rerouting the split demands.

1) Solve LPF and get optimal solution for $X_{ij}^k$ and $\alpha$.
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 $\alpha$.
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 maximum link utilization (along the path) is minimum, after routing the demand over that path.
   c) Recompute the $\alpha$ value.
   d) take off $L$ from the set $S$.

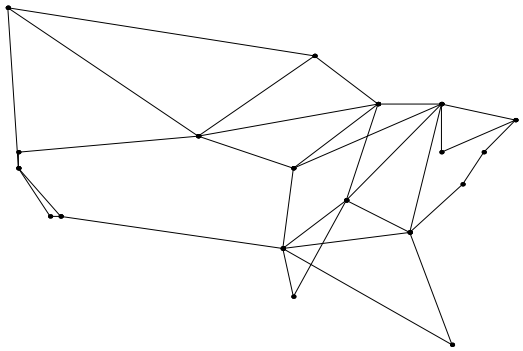

Fig. 3. Cable & Wireless Network (source: [13])

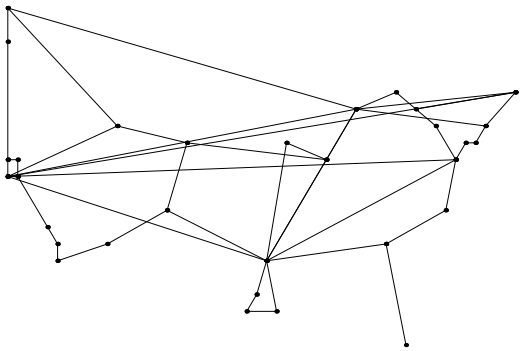

Fig. 4. CRL Network Services (source: [13])

   e) If $S$ is not empty go to 4a.

Permissible paths for a demand $k$ with source node $s$ and destination node $t$, are all those paths which start at node $s$ go through a neighbouring node $n$ ($n \neq OSPF\_ancestor_s^k$) of $s$ and follow the shortest path from $s$ to $t$.

## V. Experimental Results

In this section, we present our simulation setup and performance comparison of S-OSPF based traffic engineering with optimal traffic engineering and OSPF.

### A. Simulation Setup

We have used network topologies of the Cable and Wireless network and CRL networks available at [13]. Simulations were coded in Java and lp_solve 5.1.1.3 package( [14]) was used for solving the LPFs. We wanted to measure the performance of S-OSPF on a small and a large size network. So we chose Cable and Wireless network which has 20 nodes and 68 directional links (see Figure 3) and CRL network which has 35 nodes and 100 directional links (see Figure 4).

For the given network topologies, the link capacities are generated randomly with a uniform distribution in the range [800, 1200]. Total generated capacity of Cable and wireless netwok is $67386$ and that of CRL network is $99730$. For Cable and wireless network, we simulate $14$ network scenarios, with the number of demands ranging from $100$ to $1400$ in the increments of $100$. For CRL Network, we simulate $20$ network demand scenarios, with the number of demands ranging from $100$ to $2000$ in the increments of $100$. For all simulation runs, size of each demand is chosen randomly from uniform


1930-529X/07/$25.00 © 2007 IEEE

TABLE II
RESULTS OF CRL NETWORK

| No. of demands | Total demand | $\alpha_{opt}$ | $\alpha_{sospf}^{split}$. | $\alpha_{sospf}^{no-split}$ | No. of split demands | $\alpha_{ospf}$ |
|---|---|---|---|---|---|---|
| 100 | 555 | 0.04634 | 0.05942 | 0.06506 | 3 | 0.07228 |
| 200 | 1097 | 0.08682 | 0.09343 | 0.09498 | 1 | 0.10215 |
| 300 | 1661 | 0.14292 | 0.15877 | 0.15877 | 3 | 0.18291 |
| 400 | 2096 | 0.15024 | 0.17747 | 0.17747 | 5 | 0.19721 |
| 500 | 2775 | 0.17170 | 0.20800 | 0.20966 | 5 | 0.21780 |
| 600 | 3265 | 0.22731 | 0.23465 | 0.23465 | 7 | 0.26740 |
| 700 | 3875 | 0.22 | 0.27297 | 0.27536 | 9 | 0.36977 |
| 800 | 4349 | 0.28585 | 0.28585 | 0.29393 | 9 | 0.38906 |
| 900 | 4864 | 0.33658 | 0.33658 | 0.34326 | 16 | 0.4372 |
| 1000 | 5409 | 0.33121 | 0.36548 | 0.36548 | 5 | 0.55519 |
| 1100 | 6161 | 0.40780 | 0.40780 | 0.41418 | 6 | 0.55091 |
| 1200 | 6728 | 0.42585 | 0.47719 | 0.47719 | 6 | 0.62915 |
| 1300 | 7216 | 0.47414 | 0.53153 | 0.53153 | 4 | 0.65702 |
| 1400 | 7787 | 0.52243 | 0.52243 | 0.52518 | 9 | 0.64201 |
| 1500 | 8242 | 0.57073 | 0.60259 | 0.60259 | 6 | 0.74062 |
| 1600 | 8707 | 0.57317 | 0.57317 | 0.60380 | 10 | 0.71489 |
| 1700 | 9345 | 0.66829 | 0.66829 | 0.68895 | 7 | 0.83172 |
| 1800 | 10046 | 0.62292 | 0.63333 | 0.64361 | 11 | 0.89924 |
| 1900 | 10333 | 0.67073 | 0.70180 | 0.70878 | 9 | 0.93354 |
| 2000 | 11066 | 0.68195 | 0.81711 | 0.83708 | 6 | 1.00964 |

TABLE I
RESULTS OF CABLE AND WIRELESS NETWORK

| No. of demands | Total demand | $\alpha_{opt}$ | $\alpha_{sospf}^{split}$. | $\alpha_{sospf}^{no-split}$ | No. of split demands | $\alpha_{ospf}$ |
|---|---|---|---|---|---|---|
| 100 | 568 | 0.08272 | 0.09777 | 0.09777 | 0 | 0.11733 |
| 200 | 1106 | 0.10332 | 0.11822 | 0.12035 | 5 | 0.17511 |
| 300 | 1633 | 0.18678 | 0.21244 | 0.21244 | 3 | 0.25777 |
| 400 | 2190 | 0.22722 | 0.22722 | 0.232 | 5 | 0.26933 |
| 500 | 2705 | 0.31832 | 0.32533 | 0.32560 | 3 | 0.44977 |
| 600 | 3334 | 0.30785 | 0.31911 | 0.32056 | 3 | 0.45066 |
| 700 | 3810 | 0.39476 | 0.41155 | 0.44734 | 4 | 0.54044 |
| 800 | 4440 | 0.39642 | 0.41955 | 0.42995 | 7 | 0.59022 |
| 900 | 4950 | 0.50575 | 0.50575 | 0.52297 | 4 | 0.63466 |
| 1000 | 5589 | 0.55191 | 0.59555 | 0.59555 | 1 | 0.76444 |
| 1100 | 5992 | 0.72879 | 0.72879 | 0.73719 | 3 | 0.83733 |
| 1200 | 6486 | 0.67434 | 0.67644 | 0.68599 | 4 | 0.96533 |
| 1300 | 7154 | 0.70785 | 0.70785 | 0.70785 | 4 | 0.92533 |
| 1400 | 7631 | 0.83874 | 0.83874 | 0.83874 | 4 | 1.10844 |

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.

### *B. Simulation Results*

Table I and II presents our simulation results. The first column shows the number of demands, second column shows the total demand, the third column shows the maximum link utilization when demands are routed using the optimal LPF ($\alpha_{opt}$). When S-OSPF is used and the individual traffic demands are allowed to be split over multiple paths, the maximum link utilization for this case is shown in fourth column ($\alpha_{sospf}^{split}$). The corresponding maximum link utilization when split is not allowed in S-OSPF is shown in fifth column ($\alpha_{sospf}^{no-split}$). The sixth column presents the number of split demands when splitting is allowed. Finally, the last column shows the maximum link utilization when plain OSPF is used ($\alpha_{ospf}$). It can be seen from the tables that $\alpha_{sospf}^{split}$ and $\alpha_{sospf}^{no-split}$ are very close to $\alpha_{opt}$ for all the randomly generated demands for both the networks. These results can be explained from the fact that the maximum of link utilization, in most of the cases, is determined by the out-degree and in-degree of the source and destination nodes and also the bandwidths of links comprising those edges. In such cases, splitting the traffic appropriately, only at the source node, is sufficient to achieve good load balancing across the network. Performance of OSPF is quite poor as compared to S-OSPF and optimal LPF because of rigidity of OSPF to route packets only through the shortest path. Thus, S-OSPF can be used to provide quite effective traffic engineering performance and S-OSPF can be implemented with very minimal change to OSPF.

## VI. CONCLUSION

We presented a simple yet effective traffic engineering solution for OSPF based best effort network. We presented a Linear Programming Formulation (LPF) for the traffic engineering problem of minimizing the maximum link utilization across all the links in an OSPF network for a given set of traffic demands. This LPF would allow nodes to split traffic of a given demand, which affects the performance of higher layer protocols. Hence we modified the LPF so that individual demands are not split. This is represented as an IPF which is NP-hard. So we provided a heuristic algorithm to solve the LP. We simulated our S-OSPF for one small and one large size real backbone network. Our simulation results not only showed that S-OSPF performed much better than OSPF but also its performance was quite comparable to that of optimal LPF.

## REFERENCES


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

[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," RFC 2702 (Informational), Sept. 1999.

[3] S. Balon, F. Skivée, and G. Leduc, "Comparing traffic engineering objective functions," in *CoNEXT'05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*. New York, NY, USA: ACM Press, 2005, pp. 224–225.

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

[5] D. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, vol. 37, no. 12, pp. 42–47, 1999.

[6] 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.

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

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

[9] 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, 2002.

[10] M. S. Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *INFOCOM*, 2000, pp. 884–893. [Online]. Available: citeseer.ist.psu.edu/kodialam00minimum.html

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 2nd Ed.* Prentice Hall of India, 2003.

[12] A. K. Mishra, *Traffic Engineering Solutions for OSPF Based Best Effort Networks*. M.Tech Thesis, K. R. School of Information Technology, IIT Bombay, 2007.

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

[14] "lpsolve," http://lpsolve.sourceforge.net/5.1/.




1930-529X/07/$25.00 © 2007 IEEE