Practical OSPF Traffic Engineering

Gábor Rétvári, *Member, IEEE,* and Tibor Cinkler, *Member, IEEE*

***Abstract*—Open Shortest Path First (OSPF) traffic engineering (TE) is intended to bring long-awaited traffic management capabilities into IP networks, which still rely on today's prevailing routing protocols: OSPF or IS-IS. In OSPF, traffic is forwarded along, and split equally between, equal cost shortest paths. In this letter, we formulate the basic requirements placed on a practical TE architecture built on top of OSPF and present a theoretical framework meeting these requirements of practicality. The main contribution of our work comes from the recognition that coupled with an instance of the maximum throughput problem there exists a related inverse shortest-path problem yielding optimal OSPF link weights.**

***Index Terms*—Open Shortest Path First (OSPF), traffic engineering, linear optimization.**

I. Introduction

MOST of today's traffic engineering (TE) [1] proposals require the deployment of expensive routing and traffic forwarding hardware and software. On the other hand, ISPs have huge installation base of routers running best-effort routing protocols, like Open Shortest Path First (OSPF) [2]). OSPF provides shortest-path-first routing, simple load balancing by Equal-Cost-MultiPath (ECMP: traffic is split roughly evenly amongst equal cost paths) and means to manipulate routing through setting the administrative link weights.

Hence, it is an easy-to-deploy and overly cost-effective solution to implement traffic engineering on top of OSPF, while retaining existing routing equipment. In such an architecture, a suitable Traffic Engineer: 1) participates in OSPF signaling to learn routing information; 2) assigns paths for each session; 3) computes link weights as to assure that link weights reflect the assignment of paths (i.e., all paths, which are assigned for a particular session are shortest paths for the session); and 4) distributes the chosen link weights back to OSPF routers.

II. Motivations

The research work presented in this letter is primarily motivated by practical requirements, such as low management burden and low cost of deployment. Given that ECMP itself restricts the optimality of OSPF routing, we rather consider balanced traffic distribution and maximum achievable throughput as the overall objective of traffic engineering.

We assume that the efforts of TE are concentrated on a set of designated sessions, of which respective ingress and egress points are known in advance. However, we do not presume any knowledge on the actual demands (which would be hard to measure, predict, etc.,) other than a simple order of *priority* ("session_A transmits more traffic than session_B and both generate more than session_C"). At the moment the scope of routing information retrievable from OSPF link state information is limited to the actual topology of the network (recent standards activity is focused on this shortcoming, [3]). Therefore, OSPF TE must work with or without explicit knowledge on the capacity of network links. Additionally, a practical TE algorithm must—under all circumstances—provide reasonable link weights. Any unintended interference of the shortest paths of different sessions (ties) must be avoided. OSPF-compliant weights are integer valued and fall into the range $[1, 2^{16} - 1]$. The link weight computation algorithm must run rapidly to assure quick adaptation to topology changes or management controls. Naturally, it also must yield efficient routes.

III. Related Works

References [4] and [5] show that OSPF traffic engineering is, in general, NP hard, and propose a local search heuristic algorithm achieving close to optimal performance. However, the algorithm builds on the knowledge of the demand matrix and is claimed to be rather slow [6]. Reference [7] shows that numerous network flow problems can be transformed into shortest path problems. Reference [8] exploits this relationship to introduce a viable relaxation of the original NP hard problem, and shows that any path sets that can be of relevance to traffic engineering can be represented as shortest paths by some positive valued weight set. Reference [9] extends this result and proposes a method that is called to overcome the limitations of ECMP yet posing even more burden on network management. Neither [8] nor [9] recognizes that the shortest-path representation is not unambiguous, therefore some mechanisms to avoid ties must be in place. Tie breaking (though, in a different interpretation) is introduced in [10].

Neither of the proposals exhibited so far meets all those requirements of practicality elaborated in the previous section. Therefore, in the rest of this paper we present a novel theoretical framework suitable for serving as the foundation of an inexpensive, easy-to-deploy OSPF TE architecture.

IV. Practical OSPF Traffic Engineering

Consider the following integer linear program (ILP), the so-called *Dual Minimum Cost Maximum Throughput Problem* (D_MAX_TH). Given a connected directed graph $G(V, E)$, a set of edge capacities $u_{ij} : (i, j) \in E$, a set of edge costs $c_{ij} > 0 : (i, j) \in E$ and a set of source-destination pairs (commodities or sessions) $(s_k, d_k) : k \in K$ ordered lexicographically by their respective priorities

Manuscript received January 26, 2004. The associate editor coordinating the review of this letter and approving it for publication was Prof. N. Ghani. This work was supported by the Ministry of Education, Hungary under Reference IKTA-0092/2002.

The authors are with the High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest H-1117, Hungary (e-mail: retvari@tmit.bme.hu; cinkler@tmit.bme.hu).

Digital Object Identifier 10.1109/LCOMM.2004.837629

$\alpha_k : k \in K, \alpha_K = \max_{k\in K} \alpha_k$. The task is to compute flows (X_{ij}^k) over arcs (i,j) for any session k, such that the weighted sum of throughput Θ_k over the set of sessions is maximized

$$\max \left\{ \sum_{k\in K} \alpha_k \Theta_k - \sum_{(i,j)\in E} c_{ij} \sum_{k\in K} X_{ij}^k \right\} \tag{1}$$

$$\text{s.t.} \sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} \Theta_k & \text{if } i = s_k \\ -\Theta_k & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K; \forall i \in V \tag{2}$$

$$\sum_{k\in K} X_{ij}^k \le u_{ij} \qquad \forall (i,j) \in E \tag{3}$$

$$X_{ij}^k \ge 0 \qquad \forall k \in K; \forall (i,j) \in E. \tag{4}$$

The objective function (1) maximizes the throughput over the sessions weighted by their respective priority (see later) while minimizing the overall cost of the solution. Constraint (2) guarantees flow conservation, (3) assures that the sum of arc-flows does not violate link capacities and, finally, (4) ensures that arc-flows are nonnegative.

Now, consider the dual of the above problem, the so called *Primal Minimum Cost Maximum Throughput Problem* (P_MAX_TH). Let $U_i^k : i \in V, k \in K$ define the node potential for each session, let W_{ij} be the modified weight of a link $(i,j) \in E$. Hence, the arc-flow formulation of P_MAX_TH is

$$\min \sum_{(i,j)\in E} u_{ij} W_{ij} \tag{5}$$

$$\text{s.t.} \quad U_j^k - U_i^k \le W_{ij} \qquad \forall (i,j) \in E; \forall k \in K \tag{6}$$

$$U_{d_k} \ge \alpha_k \qquad \forall k \in K \tag{7}$$

$$W_{ij} \ge c_{ij} \qquad \forall (i,j) \in E. \tag{8}$$

P_MAX_TH is designated as the primal to follow the conventions of minimization and maximization.

Property 1: There always exists an optimal feasible solution for P_MAX_TH and D_MAX_TH, respectively. Furthermore, if c_{ij} and u_{ij} are integer valued and finite, then there exists some finite integer valued weight set W (not necessarily optimal).

Let $P_k : k \in K$ define the path set given by an optimal solution of D_MAX_TH. A path $P \in P_k$ is defined as the concatenation of links such that $\forall (i,j) \in P : X_{ij}^k > 0$.

Lemma 1: Any path P in the path set P_k defined by an optimal solution of D_MAX_TH is loop-free.

Proof: Recall that D_MAX_TH minimizes the overall cost (over a nonzero cost-set c_{ij}) of the optimal solution. Also recall that for any loopy path, there always exists a loop-free path which is cheaper and has the same capacity. ■

Property 2: The constraint system of P_MAX_TH and D_MAX_TH possesses block diagonal (angular) structure.

Hence, the *decomposition techniques* developed for the multicommodity flow problem (column-generation, Dantzig-Wolfe decomposition, etc., see [11]) can also be applied to the LP relaxation of P_MAX_TH and D_MAX_TH facilitating rapid solution of even enormously large problem instances.

The following lemma establishes the relation between an instance of the maximum throughput problem and a strongly coupled instance of the inverse shortest-path problem.

Lemma 2: Consider the weight set W defined by an optimal solution of P_MAX_TH and the path set P_k defined by *any* optimal solution of D_MAX_TH. Then, any path $P \in P_k$ is a shortest $s_k \to d_k$ path over the weight set W.

The emphasis is on *any*: an optimal weight set W represents *any* path sets P_k as shortest paths, given that P_k is optimal to the dual problem. Since the problems possess block-diagonal structure, one can use the same technique as [7], [11], [8] to prove the Lemma. Note that the weight set W yields only a rough approximation of optimal OSPF TE since the equal-splitting requirement is relaxed. Also note that even if link capacities are not precisely known it is possible to compute a reasonable path set by solving the unit-capacity relaxation of D_MAX_TH.

To provide further insight, transform the problems into path-flow formulation [7]. As of the path-flow formulation of D_MAX_TH, let f_P be the flow on path $P \in P_k$. Then, the throughput of session k is defined as $\Theta_k = \sum_{P\in P_k} f_P$ and the cost of any path P is defined as $c_P = \sum_{(i,j)\in P} c_{ij}$. Hence, the path-flow formulation of D_MAX_TH is

$$\max \sum_{k\in K} \alpha_k \Theta_k - \sum_{k\in K} \sum_{P\in P_k} c_P f_P \tag{9}$$

$$\text{s.t.} \sum_{P:(i,j)\in P} f_P \le u_{ij} \qquad \forall (i,j) \in E \tag{10}$$

$$f_P \ge 0 \qquad \forall k \in K; \forall P \in P_k \tag{11}$$

and the path-flow formulation of P_MAX_TH

$$\min \sum_{(i,j)\in E} u_{ij} W_{ij} \tag{12}$$

$$\text{s.t.} \sum_{(i,j)\in P} W_{ij} \ge \alpha_k \qquad \forall k \in K; \forall P \in P_k \tag{13}$$

$$W_{ij} \ge c_{ij} \qquad \forall (i,j) \in E. \tag{14}$$

Setting α_k accordingly ensures that: 1) maximization of throughput predominates over the minimization of cost in the objective function (1) and 2) a session with high priority takes preference over lower priority sessions. This facilitates for provisioning routing for traffic demands that are only relatively known. In order to simplify the notion, suppose that there are no two sessions that have the same priority. Let $U = \max_{(i,j)\in E} u_{ij}$ be the maximum link capacity and $m = |E|$ denote the number of arcs in G. Hence, according to (1) the following conditions must hold for α_k:

$$\alpha_{k+1} > \alpha_k mU - mU; \qquad \alpha_1 > mU.$$

This gives rise to the following adjustment of α_k:

$$\alpha_k = (mU)^k + 1. \tag{15}$$

Next, we show that in general the length of the shortest path(s) of a session equals to α_k. Thus, we shall refer to α_k as the preferred distance of session k.

Lemma 3: Consider the throughput $\Theta_k : k \in K$ defined by any optimal solution of D_MAX_TH. Furthermore, consider the weight set W defined by any optimal solution of P_MAX_TH. If for any session k the throughput is nonzero, then the length of the shortest path(s) for k is the preferred distance α_k.

Proof: From (13) we know that no path can be shorter than the preferred distance. Furthermore, $\Theta_k > 0 \Rightarrow \exists P \in$

$P_k : f_P > 0$, and, by complementary slackness we have that $f_P > 0 \Rightarrow \sum_{(i,j)\in P} W_{ij} = \alpha_k$. ■

Lemma 4: Any link weight W_{ij} in an optimal solution of P_MAX_TH is upper-bounded by α_K.

Proof: Consider an optimal solution W of P_MAX_TH, such that $\exists(i,j) : W_{ij} > \alpha_K$. In such case, there always exists a feasible solution $W' : W'_{ij} = \alpha_K$, which still satisfies (13), though, improves the objective function (12). Therefore, W is not optimal, which is a contradiction. ■

It must be emphasized that the abstraction between the path set defined by D_MAX_TH and the weight set given by P_MAX_TH is not unambiguous. From Lemma 2 we know that any path P in the optimal path set P_k is represented as a shortest path by W. However, the converse is not necessarily true. A path $P' \notin P_k$ (i.e., $f_{P'} = 0$) may very well be shortest path over W, i.e., $f_{P'} = 0 \not\Rightarrow \sum_{(i,j)\in P} W_{ij} > \alpha_k$. Hence, there is a chance that an unintended shortest path of a low priority session interferes with a path of a higher priority session. Such *ties* may lead to congestion. Tie breaking assures that certain critical links are circumvented when routing low priority traffic:

Lemma 5 [Tie Breaking]: For any link (i,j) contained in the optimal path set P_k of session $k : c_{ij} \le \alpha_k$.

Proof: From (14) and Lemma 3 we know that for any link (i,j) of an optimal path P: $(i,j) \in P \Rightarrow c_{ij} \le W_{ij} \le \sum_{(u,v)\in P} W_{uv} = \alpha_k$. ■

Thus, setting $c_{ij} > \alpha_k$ assures that any optimal path of session k circumvents (i,j). Links belonging the minimum cut(s) of high priority sessions are good candidates for tie breaking. Nevertheless, the exact method to select links subject to tie breaking is beyond the scope of this paper.

Finally, we present some simulation results to compare the performance of various OSPF TE techniques. In the case of traditional connectionless IP routing (where OSPF is extensively used) we found it very difficult to define a single measure of routing performance, which suitably captures the behavior of routing in itself. Rather, we used OSPF in conjunction with a connection-oriented infrastructure (e.g., MPLS) to dynamically route a pre-declared amount of offered traffic in the network and observed the resultant average call blocking ratio (ACBR). The ACBR is easy to measure and it has descriptive real-world interpretation. The results are averaged over a set of 40 random graphs each consisting of 15 nodes, 45 equal-capacity bidirectional links, and 4 sessions. Independent Poisson-processes were used to generate random-sized calls for the four sessions, one of which offers three times more traffic than the others. Table I summarizes the results for minimum hopcount routing [4], various forms of OSPF TE, which use the precise demand matrix: optimal routing (OPT1 [8]), optimal routing with piecewise linear cost function (OPT2 [4]), minimum link utilization routing (MINUTIL [8]), plus our proposed methods: MAX_TH (solving P_MAX_TH using identical preferred distance for each session), MAX_TH2 (setting preferred distances according to (15) to prioritize the high volume session) and MAX_TH3 (also with tie breaking over the minimum cut set). The table shows the ACBR of the high volume session (ACBR1), ACBR of all sessions (ACBR2), throughput relative to minhop routing (TH), the number of ECMP routes for the high volume session, the mean path length (MPL) and the average time taken by the optimization.

TABLE I
SIMULATION RESULTS FOR VARIOUS OSPF TE TECHNIQUES

	ACBR1	ACBR2	TH	#ECMP	MPL	Time [s]
MINHOP	43.4%	18.6%	1	2	1.66	–
OPT1	23%	11.6%	1.09	5	2.06	0.16
OPT2	21.7%	10.2%	1.09	14	2.65	0.17
MINUTIL	22.8%	9.2%	1.09	9	2.32	0.15
MAX_TH	24.5%	9.9%	1.1	10	2.43	0.16
MAX_TH2	19.5%	8.8%	1.1	17	2.65	0.17
MAX_TH3	18.1%	8.6%	1.11	16	2.64	0.19

Our most important conclusion is that, given the inherent uncertainty of ECMP load balancing, it is not worth precisely provisioning OSPF routing w.r.t. a given demand matrix. Instead, our priority-driven relative dimensioning scheme performs reasonably better. Furthermore, properly setting the preferred distances and using tie breaking may be highly beneficial. However, OSPF TE comes with its own drawbacks—the price one has to pay for higher throughput is the increased average delay caused by the significant growth in the average path length.

V. CONCLUSION

We have shown that OSPF link weights produced by the Primal Minimum Cost Maximum Throughput Problem supply a plausible basis to build a practical OSPF TE architecture onto. Such optimal link weights are guaranteed to exist, integer valued, upper-bounded, and produce a route set that maximizes the throughput of the network. For reasonable sized networks the corresponding ILP can be solved rapidly. Additionally, we introduced the notion of preferred distance to prioritize high-volume sessions and we proposed a scheme for tie breaking to avoid ambiguity in the shortest path representation. We believe that the framework presented in this letter is the first one that exhibits all the indispensable properties to make OSPF TE a good choice for "poor man's traffic engineering."

REFERENCES

[1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering,", RFC 3272, May 2002.
[2] J. Moy, "OSPF version 2,", RFC 2328, Apr. 1998.
[3] D. Katz, D. Yeung, and K. Kompella, "Traffic engineering extensions to OSPF version 2," in *Internet Draft*, Oct. 2002.
[4] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag.*, vol. 40, pp. 118–124, Oct. 2002.
[5] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 756–767, May 2002.
[6] ——, "Internet traffic engineering by optimizing OSPF weights," in *Proc. INFOCOM (2)*, 2000, pp. 519–528.
[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[8] Z. Wang, Y. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *Proc. INFOCOM*, Apr. 2001, pp. 565–571.
[9] A. Sridharan, C. Diot, and R. Guérin, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proc. INFOCOM*, Mar. 2003, pp. 1167–1177.
[10] M. Thorup, Avoiding Ties in Shortest Path First Routing, 2001. unpublished manuscript.
[11] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. New York: Wiley, Jan. 1990.