One More Weight is Enough: Toward the Optimal Traffic Engineering with OSPF

Ke Xu[†], Hongying Liu[‡], Jiangchuan Liu[§], Meng Shen[†]

Tsinghua University[†]   Beihang University[‡]   Simon Fraser University[§]

{xuke,shenmeng}@csnet1.cs.tsinghua.edu.cn   liuhongying@buaa.edu.cn   jcliu@cs.sfu.ca

Abstract—**Traffic Engineering (TE) leverages information of network traffic to generate a routing scheme optimizing the traffic distribution so as to advance network performance. However, optimize the link weights for OSPF to the offered traffic is an known NP-hard problem [16]. In this paper, motivated by the fairness concept of congestion control [10], we firstly propose a generic objective function, where various interests of providers can be extracted with different parameter settings. And then, we model the optimal TE as the utility maximization of multi-commodity flows with the generic objective function and theoretically show that any given set of optimal routes corresponding to a particular objective function can be converted to shortest paths with respect to a set of positive link weights. This can be directly configured on OSPF-based protocols. On these bases, we employ the Network Entropy Maximization(NEM) framework [20] and develop a new OSPF-based routing protocol, SPEF, to realize a flexible way to split traffic over shortest paths in a distributed fashion. Actually, comparing to OSPF, SPEF only needs one more weight for each link and provably achieves optimal TE. Numerical experiments have been done to compare SPEF with the current version of OSPF, showing the effectiveness of SPEF in terms of link utilization and network load distribution.**

Index Terms—**Traffic engineering, OSPF, Utility, Load balance, Routing**

I. INTRODUCTION

The primary role of Internet Service Providers (ISPs) is to guarantee service via deploying infrastructures, managing network connectivity and balancing traffic load inside their networks [12]. The goal of Traffic Engineering (TE) is to ensure efficient routing to minimize network congestion, so that users can experience low packet loss, high throughput, and low latency. Traffic Engineering leverages information from traffic entering and leaving the network to generate a routing scheme that optimizes network performance. In particular, an ISP solves the TE problem by adjusting the routing configuration to the prevailing traffic.

In this paper, we focus on traffic engineering within a single Autonomous Systems (AS), in which we assume that the egress point of each external destination is known and fixed. Traffic engineering thus depends on a set of performance objectives that guide path selection, as well as effective mechanisms for routers to select paths that satisfy these objectives [15].

Open Shortest Path First (OSPF) is a commonly used intra-domain routing protocol [24], which provides the network operators a way to control network routing by configuring OSPF link weights. The quality of OSPF-based traffic engineering depends largely on the choice of weights. Link weights can have a reasonable default configuration based on link capacity, *e.g.*, Cisco's InvCap [13] sets the weight of a link inversely proportional to its capacity, which can be explained by the $M/M/1$ queuing model. Although fairly intuitive and convenient, these setting approaches might lead to undesirable network load distribution, since they do not take the expected traffic demand into consideration. In practice, given network link capacities and expected traffic demands, the link weights can be optimized by ISPs according to a certain object function. However, computing the optimal link weights under the evenly traffic splitting scheme has been proven to be NP-complete [16].

Challenges. In this paper, we take an important step towards building an OSPF-based routing protocol that can achieve the optimal traffic engineering. Although this optimization problem has attracted a great research interest and been extensively studied (e.g., [18], [19], [23]), there are still several challenges to be further studied, including the following:

1. **Can we design a generic objective function to meet various providers' needs?** Network providers are usually interested in various indicators to improve the network performance in different ways, e.g.,some of them might prefer to lower the maximum link utilization, while others might try to minimize path lengths. Accordingly, various objective functions have been proposed to capture these demands. Unfortunately, a set of optimal link weights with one objective function does not necessarily perform well with another objective function, or may be even worse. Are there common features existed among these different objective functions? Can we design a generic objective function to meet providers' needs?

2. **Can we guarantee the universal existence of optimal link weights?** Preceding researches showed that the optimal link weights existed with a certain kind of object functions. Although this outcome is encouraging to some extent, we still prefer to ensure the universal existence of optimal link weights under various objective functions.

3. **Can we achieve the optimal TE for intra-domain IP networks based on OSPF?** As a distributed link-state routing protocol, OSPF uses the shortest path routing with destinations based hop-by-hop forwarding and Equal-Cost Multi-Path (ECMP) mechanism to evenly split the corresponding traffic over all available equal-cost paths. Many approaches

are proposed attempting to achieve the "optimal" routing based on OSPF. Wang et al. [19] and Srivastava et al. [18] proposed flexible solutions to efficiently split traffic over shortest paths, but these centralized solutions went against the distributed feature of OSPF. A new link-state protocol named PEFT, recently proposed by Xu et al. [20], successfully realized a flexible traffic splitting scheme in a distributed manner, whereas failed to maintain the shortest paths in packet forwarding thus sacrificing a key benefit of OSPF. Guaranteeing the crucial features of OSPF in terms of scalability and efficiency are thus a great challenge in achieving the optimal traffic engineering goals based on OSPF.

Our Approach and Contributions. Inspired by the fairness criterion of congestion control [10], we firstly propose a generic objective function named $(\mathbf{q}, \beta)$ proportional load balance to consider various load balance demands in TE. Then we model the optimal TE as the utility maximization of multi-commodity flows and propose a distributed dual decomposition method to compute the optimal link weights. Based on these, we develop a new OSPF-based protocol, Shortest paths Penalizing Exponential Flow-splitting (SPEF). It has been proved able to achieve the optimal TE.

Toward the optimal TE, in SPEF, we only need one more weight for each link. For simplicity, we hereafter refer to the optimal and the additional link weights as the first and second link weights, respectively. We use the Network Entropy Maximization (NEM) framework proposed in [20] to obtain the second link weights, aiming at maintaining the path diversity. In SPEF, packets forwarding is the same as OSPF: hop-by-hop along the shortest paths constructed based on destination according to the first link weights. When there are multiple shortest paths for some source destination pairs in view of the first link weights, the flow split ratio over the multiple shortest paths can be independently computed by the routers from the second link weights. In particular, we address the above challenges as follows:

1. To capture various operators' needs, we design a new generic objective function motivated by the fairness concept in congestion control [10], $(\mathbf{q}, \beta)$ proportional load balance, from which a family of load balance objective functions could be derived to meet various providers' needs. E.g., $(\mathbf{q}, \beta)$ proportional load balance could converge to min-max load balance with the increase of β, making the MLU minimized. It can also reduce to proportional load balance with $\beta = 1$, corresponding to maximizing the product of unused capacity in networks. The generic objective function provides a chance to make a trade-off by operators, since they can just simply vary parameter settings according to their special needs.

2. To ensure the existence of optimal link weights, we model the optimal TE as the utility maximization of multi-commodity flows with the generic objective function and theoretically show that any given set of optimal routes corresponding to a particular objective function can be converted to shortest paths with respect to a set of positive link weights, which can be explicitly formulated using the spare network capacity and objective function.

3. To achieve the optimal TE based on OSPF, we develop a new routing protocol, SPEF, proving that it can achieve the optimal TE for intra-domain IP networks. Although we leverage the NEM framework proposed in PEFT, the key difference between them is in that: SPEF realizes a flexible flow splitting over shortest paths in a distributed fashion, guaranteeing the crucial features of OSPF in terms of scalability and efficiency.

Paper Organization. The rest of the paper is organized as follows. We propose a new generic objective function of load balance in traffic engineering in Section II and theoretically prove the existence of the optimal link weights with the above generic function in Section III. The new OSPF-based protocol is developed in Section IV, following which is the performance evaluation in Section V. Related work is summarized in Section VI, before we conclude with the achievements and extensions in Section VII.

II. NETWORK MODEL AND LOAD BALANCE CRITERIA

The notion of load balance characterizes how traffic should be distributed to the links. In this section, we first give the network model and then propose some new definition for load balance, which is motivated by the notion of fairness [6], [10].

A. Network Model

We consider a directed network $\mathcal{G} = (\mathcal{N}, \mathcal{J})$ with vertex set $\mathcal{N}$, edge set $\mathcal{J}$, and R source-destination vertex pairs $\{s_1, t_1\}, \cdots, \{s_R, t_R\}$. Each edge (i, j) has a capacity c_{ij}, which is a measure for the amount of traffic flow it can take. A demand (traffic) for (s_r, t_r) is d_r, which denotes the average intensity of traffic entering the network at vertex s_r and exiting at vertex t_r. In the following, we use notations N, J to denote the cardinalities of sets $\mathcal{N}$ and $\mathcal{J}$ respectively and $\mathcal{R} = \{1, \cdots, R\}$ to denote the source-destination vertex pairs index set.

The multi-commodity flow problem is a network flow problem with multiple commodities (or goods) flowing through the network, with different source and sink nodes. The more customary way to treat routing in a network is to consider it as a multi-commodity flow problem. Denote the destination node set with $\mathcal{D} = \{t \in \mathcal{N} : \exists \, r \in \mathcal{R} \text{ s.t. } t_r = t\}$. The traffic flow to each destination $t \in \mathcal{D}$ can be regarded as a commodity. The flow of commodity t along edge (i, j) is f_{ij}^t. Find an assignment of flow satisfying the constraints:

$$f_{ij} \overset{\triangle}{=} \sum_{t \in \mathcal{D}} f_{ij}^t \leq c_{ij}, \quad \forall (i, j) \in \mathcal{J} \tag{1a}$$

$$\sum_{j:(s,j) \in \mathcal{J}} f_{sj}^t - \sum_{i:(i,s) \in \mathcal{J}} f_{is}^t = d_s^t, \forall t \in \mathcal{D}, \forall s \in \mathcal{N} \backslash \{t\}, \tag{1b}$$

$$f_{ij}^t \geq 0, \; \forall t \in \mathcal{D}, (i, j) \in \mathcal{J}, \tag{1c}$$

where (1a) and (1b) are the capacity constraints and flow conservation constraints, respectively, and $d_s^t \geq 0$ is the expected traffic entering the network at node s and destined to node t. Set $d_s^t = d_r$ if there exists $r \in \mathcal{R}$ such that $s_r = s$ and $t_r = t$, or set $d_s^t = 0$ otherwise.

We say a traffic distribution $\mathbf{f} = (f_{ij}, (i, j) \in \mathcal{J})$ is feasible if there exists $(\mathbf{f}^t, t \in \mathcal{D})$ such that $(\mathbf{f}, \mathbf{f}^t, t \in \mathcal{D})$ satisfies the
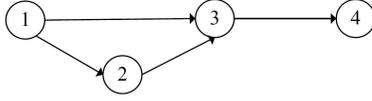
Fig. 1.   An example illustrating the notions of load balance for TE

multi-commodity flow constraints (1). If $\mathbf{f}$ is feasible, the total load on and the utilization of the link $(i,j) \in \mathcal{J}$ are f_{ij} and $\frac{f_{ij}}{c_{ij}}$ respectively, which depend on how the network decides to route the traffic. Now, one main task is to find a *appropriate* and *feasible* traffic distribution $\mathbf{f}$.

An objective function enables quantitative comparisons between different routing solutions in terms of load f_{ij} on the links. Traffic engineering usually considers a link-cost function $\Phi(\mathbf{f}, \mathbf{c})$ that is an increasing function of $\mathbf{f}$. *Optimal traffic engineering* [14] means that the TE cost function is minimized over multi-commodity flow constraints (1).

B. Load Balance Criteria

In order to use the network resources efficiently, spare resource (such as bandwidth) are made to ensure high probability of data arrival to its destinations. Now we will turn to discuss the load balance criteria based on spare link capacity for link (i,j), which is $s_{ij} = c_{ij} - f_{ij}$.

It is well known that minimizing MLU is over sensitive to individual bottleneck links that may be difficult to avoid [15]. In addition, the maximum link utilization function does not penalize solutions that force traffic to traverse very long paths. We first use an example to illustrate that MLU is not a well-defined objective function. Consider the topology in Fig. 1, there are four edges with capacities all being 1s. The nonzero demands are 1 for source-destination pair $(1,3)$ and 0.9 for source-destination pair $(3,4)$, respectively. There are two paths for source-destination pairs $(1,3)$, namely 1-3 and 1-2-3. There is a single path for source-destination pair $(3,4)$, i.e., 3-4. The link utilizations are shown in the last column of TABLE I. There is infinite optimal traffic distribution for minimizing MLU. How to evaluate these optimal traffic distribution? A formal definition is min-max load balance.

A traffic distribution $\mathbf{f}^*$ is said to be *min-max load balanced* if it is feasible and for any other feasible traffic distribution $\mathbf{f}$, the following condition holds: if $s_{ij} > s_{ij}^*$ for some $(i,j) \in \mathcal{J}$, then there exists $(u,v) \in \mathcal{J}$ such that $\frac{s_{uv}^*}{c_{uv}} \le \frac{s_{ij}^*}{c_{ij}}$ and $s_{uv} < s_{uv}^*$.

We first show that a min-max load balancing traffic distribution $\mathbf{f}^*$ makes MLU minimized. Minimizing MLU can be formulated with the spare capacity as

$$\text{minimize} \quad \max_{(i,j)\in\mathcal{J}} \left(1 - \frac{s_{ij}}{c_{ij}}\right) \qquad (2)$$

Assume that $\mathbf{f}^*$ is min-max load balanced and does not solve the problem (2). Then there exists a feasible traffic distribution $\mathbf{f}$ such that

$$\max_{(i,j)\in\mathcal{J}} \left(1 - \frac{s_{ij}}{c_{ij}}\right) < \max_{(i,j)\in J} \left(1 - \frac{s_{ij}^*}{c_{ij}}\right). \qquad (3)$$

Let $(i,j) = \arg\max_{(i,j)\in\mathcal{J}} \left(1 - \frac{s_{ij}^*}{c_{ij}}\right)$. By (3), we have $1 - \frac{s_{ij}}{c_{ij}} < 1 - \frac{s_{ij}^*}{c_{ij}}$. For $\mathbf{f}^*$ is min-max load balanced, there exists $(u,v) \in \mathcal{J}$ such that $1 - \frac{s_{uv}^*}{c_{uv}} \ge 1 - \frac{s_{ij}^*}{c_{ij}}$ and $1 - \frac{s_{uv}}{c_{uv}} > 1 - \frac{s_{uv}^*}{c_{uv}}$, which contradicts with (3).

The min-max load balancing traffic distribution for the topology in Fig. 1 is shown in TABLE I. It can be seen that the min-max load balance is not overly sensitive to individual bottleneck links that may be difficult to avoid. But similar with minimizing MLU, the min-max load balancing traffic distribution does not penalize solutions that force traffic to traverse very long paths. Consider path 1-2-3 and path 1-3 for the source-destination pair $(1,3)$. If the capacities in Fig. 1 are five times bigger, then it would not be worthwhile sending the traffic from 1 through a detour over 2 to 3. For it does not really matter that we reduce the second maximum link utilization from 20% to 10%.

A traffic distribution $\mathbf{f}^*$ is *proportional load balanced* if it is feasible and for any other feasible traffic distribution $\mathbf{f}$, the aggregate of proportional changes of spare capacity is zero or negative:

$$\sum_{(i,j)\in\mathcal{J}} \frac{s_{ij} - s_{ij}^*}{s_{ij}^*} \le 0,$$

where $s_{ij} = c_{ij} - f_{ij}$ is the spare capacity of link $(i,j) \in \mathcal{J}$ for a feasible traffic distribution $\mathbf{f}$.

A traffic distribution $\mathbf{f}^*$ is *weighting proportional load balanced* if it is feasible, and if for any other feasible traffic distribution $\mathbf{f}$

$$\sum_{(i,j)\in\mathcal{J}} q_{ij} \frac{s_{ij} - s_{ij}^*}{s_{ij}^*} \le 0,$$

where q_{ij} is a nonnegative constant for all $(i,j) \in \mathcal{J}$.

The following definition is a generalization of proportional load balance and min-max load balance. A traffic distribution $\mathbf{f}^*$ is $(\mathbf{q}, \beta)$ *proportional load balanced* if it is feasible and for any other feasible traffic distribution $\mathbf{f}$

$$\sum_{(i,j)\in\mathcal{J}} q_{ij} \frac{s_{ij} - s_{ij}^*}{(s_{ij}^*)^{\beta}} \le 0, \qquad (4)$$

where $s_{ij} = c_{ij} - f_{ij}$ and β is a nonnegative parameter.

The given definition reduces to that of proportional balance with $\beta = 1$. As β grows large, it converges to that of min-max load balance.

III. OPTIMAL WEIGHTS EXISTENCE

In this section, we first resort to the utility maximization of multi-commodity flows to model the optimal TE. And then we theoretically prove that the optimal link weights always exist under the generic objective functions with different parameters.

A. Utility Model of Traffic Engineering

For the offered traffic, TE changes routing to minimize network congestion. Here we use the utility maximization solution to route traffic which is equal to the multi-commodity

Link	$\beta = 0$		$\beta = 1$		B. Fortz & M. Thorup [14]		min-max		MLU [19]	
	weights	utilizations	weights	utilizations	weights	utilizations	weights	utilizations	weights	utilizations
$(1,3)$	2	1.00	3	0.67	4.6	0.67		0.50	0	$a^{\ddagger}$
$(3,4)$	1	0.90	10	0.90	40.0	0.90		0.90	1	0.90
$(1,2)$	1	0.00	1.5	0.33	2.3	0.33		0.50	0	$1-a$
$(2,3)$	0	0.00	1.5	0.33	2.3	0.33		0.50	0	$1-a$

$^{\ddagger}a$ is a constant in interval $[0.1, 0.9]$

flow solution. The reason is two-fold: (a) it is optimal, *i.e.*, it gives the routing with maximum spare capacity utility; (b) it can be realized by routing protocols that use MPLS tuneling, or in a distributed fashion by OSPF routing.

We associate link (i,j) with an operator, and assume that if a spare capacity s_{ij} is held by operator (i,j), which has utility $V_{ij}(s_{ij})$ to the operator. We assume that the utility $V_{ij}(s_{ij})$ is an increasing, concave and continuously differentiable function of s_{ij} over the range $s_{ij} \geq 0$, and $V'_{ij}(s_{ij}) > 0$ over the range $s_{ij} \geq 0$. Assume further that utilities are additive, so that the aggregate utility of spare capacity $\mathbf{s} = (s_{ij}, (i,j) \in \mathcal{J})$ is $\sum_{(i,j) \in \mathcal{J}} V_{ij}(s_{ij})$.

It is the concavity of the function V_{ij} that forces load balance among links. If V_{ij} is a convex increasing function instead of a concave, then maximize the aggregate utility. Larger spare capacity s_{ij} should be increased, since the rate of increase of $V_{ij}(s_{ij})$ is increasing in s_{ij}. When V_{ij} is linear, the rate of increase of V_{ij} is the same for all s_{ij}. When V_{ij} is concave, a smaller spare capacity s_{ij} is preferred, since $V'_{ij}(x) > V'_{ij}(y)$ if $x < y$.

Now the optimal traffic engineering can be formulated as maximizing the aggregated utility under the multi-commodity flow constraints (1).
TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$

$$\text{maximize}_{\mathbf{f}^t \geq \mathbf{0}} \sum_{(i,j) \in \mathcal{J}} V_{ij}(s_{ij}) \tag{5a}$$

$$\text{subject to} \quad \mathbf{c} - \sum_{t \in \mathcal{D}} \mathbf{f}^t = \mathbf{s} \geq \mathbf{0} \tag{5b}$$

$$\mathbf{Bf}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D}, \tag{5c}$$

where $\mathbf{B}$, an $N \times J$ node-arc incidence matrix for network $\mathcal{G}$, is introduced to represent the multi-commodity flow constraints (1). The j-th column of $\mathbf{B}$ corresponding to link $(u,v) \in \mathcal{J}$ is defined as

$$B_{ij} = \begin{cases} 1, & i = u \\ -1, & i = v \\ 0, & \text{otherwise,} \end{cases}$$

There is a unique optimum for the spare capacity vector $\mathbf{s}$, since the objective function (5a) is a strictly concave function of $\mathbf{s}$. But there may be many values of the flow vector $(\mathbf{f}^t, t \in \mathcal{D})$ satisfying relations (5b) and (5c). Say that $\mathbf{s}$ solves TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$ if there exists $(\mathbf{f}^t, t \in \mathcal{D})$ such that $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ solves the optimization problem (5).

From the general theory of constrained convex optimization [2]. It follows that $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ solves problem (5) if and only if there exist Lagrangian multiplier vectors $\mathbf{w}$ and $\boldsymbol{\nu}^t, t \in \mathcal{D}$ that satisfy

$$\mathbf{c} - \sum_{t \in \mathcal{D}} \mathbf{f}^t = \mathbf{s}, \quad \mathbf{Bf}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D} \tag{6a}$$

$$V'_{ij}(s_{ij}) - w_{ij} = 0, \quad \text{if} \ s_{ij} > 0 \tag{6b}$$

$$\leq 0, \quad \text{if} \ s_{ij} = 0 \tag{6c}$$

$$\nu^t_j - \nu^t_i - w_{ij} = 0, \quad \text{if} \ f^t_{ij} > 0 \tag{6d}$$

$$\leq 0, \quad \text{if} \ f^t_{ij} = 0. \tag{6e}$$

we define $c'_{ij} = c_{ij} - s_{ij}$ as the *target capacity* for each link, which is no greater than the actual capacity c_{ij} (a "virtual" capacity). This is also desirable since it leads to an empty equilibrium.

From $V'_{ij}(s_{ij}) > 0$, Eq. (6b) and (6c), we have $w_{ij} > 0$. The Lagrangian multiplier vectors $\mathbf{w}$ and $\boldsymbol{\nu}^t$ have several simple interpretations. Let $p : v_{j_0} v_{j_1} v_{j_2} \cdots v_{j_m} v_{j_{m+1}}$ be a possible path of source-destination pair (s,t), where $j_0 = s$ and $j_{m+1} = t$. For example, if $y_p = \min_{k=1,2,\cdots,m+1} f^t_{v_{j_{k-1}} v_{j_k}} > 0$, we have $\sum_{(i,j) \in p} w_{ij} = \nu^t_t - \nu^t_s \leq \sum_{(i,j) \in \bar{p}} w_{ij}$ for any other path $\bar{p}$ that connects the same source-destination pair (s,t) under the conditions (6d) and (6e). We may view w_{ij} as the implied cost of traffic through link (i,j). Alternatively, w_{ij} is the *shadow price* of additional capacity at link (i,j). We can also regard $\mathbf{w}$ as the weight set by the operator and $\boldsymbol{\nu}^t$ as the vector of node potentials $(\nu^t_i : i \in \mathcal{N})$ for destination t.

Let $\mathbf{w} = (w_{ij} : (i,j) \in \mathcal{J})$. And let $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ be a solution of (5). We have shown that $\mathbf{f}^t$ determines the shortest path for each source-destination pair (s,t) under the link weights $\mathbf{w}$, which is determined explicitly by the utility function V_{ij} and the spare capacity s_{ij} through Eq. (6b) and (6c).

If link (i,j) is charged price of per unit spare capacity, and is to freely vary the spare capacity s_{ij}, then the utility maximization problem for link (i,j) becomes
Link$_{ij}(V_{ij}; w_{ij})$

$$\begin{aligned} \text{maximize} \quad & V_{ij}(s_{ij}) - w_{ij} s_{ij} \\ \text{subject to} \quad & s_{ij} \geq 0. \end{aligned} \tag{7}$$

If the network receives a revenue w_{ij} per unit spare capacity from link (i,j), and is allowed to freely vary the spare capacity s_{ij}, then the revenue optimization problem for the network is

as follows.

Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$

$$\text{maximize}_{\mathbf{f}^t \geq \mathbf{0}} \quad \sum_{(i,j) \in \mathcal{J}} w_{ij} s_{ij}$$
$$\text{subject to} \quad \mathbf{c} - \sum_{t \in \mathcal{D}} \mathbf{f}^t = \mathbf{s} \geq \mathbf{0} \quad (8)$$
$$\mathbf{B} \mathbf{f}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D}.$$

$\mathbf{s}$ solves Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$ if there exists $(\mathbf{f}^t, t \in \mathcal{D})$ such that $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ solves the problem (8).

Remark 1: Reducing the spare capacity $\mathbf{s}$ from (8), we have that Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$ is a minimum cost multi-commodity flow problem [3], *i.e.*

$$\text{minimize}_{\mathbf{f}^t \geq \mathbf{0}} \quad \sum_{(i,j) \in \mathcal{J}} w_{ij} \sum_{t \in \mathcal{D}} f_{ij}^t$$
$$\text{subject to} \quad \sum_{t \in \mathcal{D}} f_{ij}^t \leq c_{ij}, \forall (i,j) \in \mathcal{J} \quad (9)$$
$$\mathbf{B} \mathbf{f}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D}.$$

B. Optimal Weights Existence

Theorem 3.1 (weight-setting): There exists a weight vector $\mathbf{w} = (w_{ij}, (i,j) \in \mathcal{J})$ such that the vector $\mathbf{s} = (s_{ij}, (i,j) \in \mathcal{J})$, formed from the unique solution s_{ij} to Link$_{ij}(V_{ij}; w_{ij})$, solves Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$. The vector $\mathbf{s}$ also solves TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$.

Proof: Let $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ be a solution of TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$, $\mathbf{w} = (w_{ij}, (i,j) \in \mathcal{J})$ and $(\boldsymbol{\nu}^t, t \in \mathcal{D})$ be the Lagrangian multiplier vectors, i.e., Eq. (6a)-(6e) are satisfied. We have that s_{ij} is the unique solution s_{ij} to Link$_{ij}(V_{ij}; w_{ij})$ for each $(i,j) \in \mathcal{J}$. It can be check that $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ is a KKT point of Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$ with Lagrangian multiplier vectors $(w_{ij}, (i,j) \in \mathcal{J})$ and $(\boldsymbol{\nu}^t, t \in D)$. $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ solves Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$, for which is a convex optimization problem.

In addition, let a vector $(w_{ij}, (i,j) \in \mathcal{J})$ such that the vector $\mathbf{s} = (s_{ij}, (i,j) \in \mathcal{J})$, formed from the unique solution s_{ij} to Link$_{ij}(V_{ij}; w_{ij})$ for each $(i,j) \in \mathcal{J}$, solves Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w})$. Then there exists flow vector $(\mathbf{f}^t, t \in \mathcal{D})$ and Lagrangian multiplier vector $(p_{ij}, (i,j) \in \mathcal{J})$ and $(\mathbf{q}^t, t \in \mathcal{D})$ such that

$$\mathbf{c} - \sum_{t \in \mathcal{D}} \mathbf{f}^t = \mathbf{s}, \quad \mathbf{B} \mathbf{f}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D} \quad (10a)$$
$$w_{ij} - p_{ij} = 0, \quad \text{if} \ s_{ij} > 0 \quad (10b)$$
$$\leq 0, \quad \text{if} \ s_{ij} = 0 \quad (10c)$$
$$q_j^t - q_i^t - p_{ij} = 0, \quad \text{if} \ f_{ij}^t > 0 \quad (10d)$$
$$\leq 0, \quad \text{if} \ f_{ij}^t = 0. \quad (10e)$$

Furthermore, by that s_{ij} solves Link$_{ij}(V_{ij}; w_{ij})$, we have

$$V_{ij}'(s_{ij}) = w_{ij}, \ \text{if} \ s_{ij} > 0$$

and

$$V_{ij}'(s_{ij}) \leq w_{ij}, \text{if} \ s_{ij} = 0.$$

Replacing Eq. (10b) and (10c) by this condition, we have that $(\mathbf{f}, \mathbf{s})$ satisfies conditions (6a)-(6e) by replacing $\mathbf{w}$ and $\boldsymbol{\nu}^t$ by $\mathbf{p}$ and $\mathbf{q}^t$ respectively. This establishes that $\mathbf{s}$ solves TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$, and hence the final part of the theorem. ∎
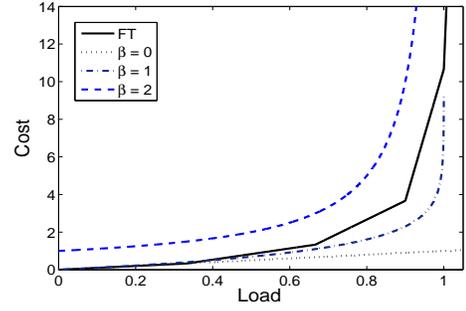

Fig. 2. Different link cost as a function of the load for a link capacity 1, where FT denotes the one proposed by Fortz and Thorp [14] and $q_{ij} = 1$ in (4)

We now examine the engineering implications of Theorem 3.1. It is true that, the Lagrangian multiplier vector $(w_{ij}, (i,j) \in \mathcal{J})$ gives link weights such that all the traffic flow will be forwarded along the minimum cost multi-commodity problem solution. Meanwhile the link (i,j) maximizes it's utility through retaining a proper spare capacity. Inversely, if there exists link weights $(w_{ij}, (i,j) \in \mathcal{J})$ such that the vector $\mathbf{s} = (s_{ij}, (i,j) \in \mathcal{J})$, formed from the unique solution s_{ij} to Link$_{ij}(V_{ij}; w_{ij})$ for each $(i,j) \in \mathcal{J}$, is the same with the solution of minimum cost multi-commodity problem (9), then $\{w_{ij}, (i,j) \in J\}$ is a set of link weights such that all the commodity flow will be forwarded along the shortest paths. Meanwhile, $\mathbf{s}$ solves traffic engineering problem TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$.

Lemma 3.2: Let $g(\mathbf{x})$ be continuously differentiable, i.e. $\frac{\partial g}{\partial x_i}$ exists and continues for all i. It holds that $g(\mathbf{x})$ is concave if and only if

$$g(\mathbf{x}) - g(\mathbf{x}^*) \leq \nabla g(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*)$$

holds for any $\mathbf{x}$ and $\mathbf{x}^*$ (see [2] and [4]).

Theorem 3.3: A traffic distribution $\mathbf{f}^*$ is $(\mathbf{q}, \beta)$ proportional load balanced if and only if $\mathbf{s}^* = \mathbf{c} - \mathbf{f}^*$ solves TE$(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$, where the objective function in (5a) is given by

$$V_{ij}(s_{ij}) = \begin{cases} q_{ij} \log s_{ij} & \text{if} \ \beta = 1 \\ q_{ij}(1-\beta)^{-1} s_{ij}^{1-\beta} & \text{if} \ \beta \neq 1. \end{cases} \quad (11)$$

Proof: Let $g(\mathbf{s}) = \sum_{(i,j) \in \mathcal{J}} V_{ij}(s_{ij})$ and $\mathbf{s}^*$ be a solution of (5) with objective $g(\mathbf{s})$. Let $\mathbf{f}^* = \mathbf{c} - \mathbf{s}^*$ and $(w_{ij}^*, (i,j) \in \mathcal{J})$ be a Lagrangian multiplier vector, i.e., satisfying (6a)-(6e). Let $\mathbf{f}$ be any feasible traffic distribution. Then $\mathbf{s} = \mathbf{c} - \mathbf{f} \geq \mathbf{0}$. Multiplying Eq. (6b) and (6c) by $s_{ij} - s_{ij}^*$ and summing over $(i,j) \in \mathcal{J}$, we find $(\nabla g(\mathbf{s}^*))^T (\mathbf{s} - \mathbf{s}^*) \leq \sum_{(i,j) \in \mathcal{J}} w_{ij}^* (s_{ij} - s_{ij}^*) \leq 0$ since $\mathbf{s}^*$ solves Network$(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{w}^*)$ (known in the proof of Theorem 3.1). Therefore,

$$(\nabla g(\mathbf{s}^*))^T (\mathbf{s} - \mathbf{s}^*) \leq 0. \quad (12)$$

By the definition of $g(\mathbf{s})$, we have $\frac{\partial g}{\partial s_{ij}} = \frac{q_{ij}}{s_{ij}^\beta}$. So Eq. (12) reduces to Eq. (4). We have shown that $\mathbf{f}^*$ is $(\mathbf{q}, \beta)$ proportional load balance.

(a) Weight vs load balance parameter β with $q_{ij} = 1$

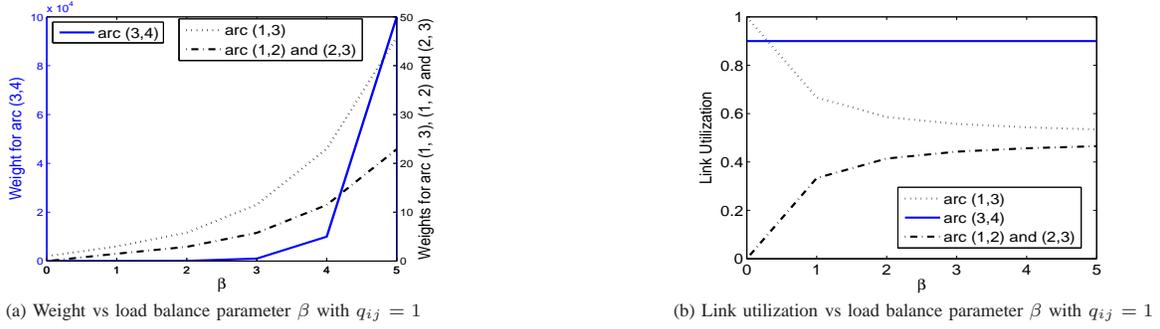(b) Link utilization vs load balance parameter β with $q_{ij} = 1$

Fig. 3. An example illustrating the notions of load balance for TE

For the converse, assume that $\mathbf{f}^*$ is $(\mathbf{q}, \beta)$-proportional load balance and $\mathbf{s}^* = \mathbf{c} - \mathbf{f}^*$. So Eq. (4) holds with $\mathbf{f}^*$, i.e., Eq. (12). For $g(\mathbf{s})$ is concave, we have $g(\mathbf{s}) - g(\mathbf{s}^*) \leq 0$ holds for any feasible traffic distribution $\mathbf{f}$ such that $\mathbf{s} = \mathbf{c} - \mathbf{f}$ by Lemma 3.2. It follows that $\mathbf{s}^*$ solves $TE(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$, where the objective function in Eq. (5a) is defined by Eq. (11). ∎

Based on Theorem 3.1 and Theorem 3.3, we can give the engineering interpretation of some specific $(\mathbf{q}, \beta)$ proportional load balance.

Example 1: Proportional load balance.

If a traffic distribution $\mathbf{f}$ is proportionally load balancing, then it solves the TE problem (5) with $V_{ij}(s_{ij}) = \log s_{ij}$. From Eq. (6b) and (6c), we can get $w_{ij} = \frac{1}{c_{ij} - f_{ij}}$, *i.e.* the average packet delay on link (i, j) is based on the $M/M/1$ queueing model [1], where $f_{ij} = \sum_{t \in \mathcal{D}} f_{ij}^t$. From the discussion above, we have that if path p^* for (s, t) bears positive traffic $y_{p^*} > 0$, then $\sum_{(i,j) \in p^*} \frac{1}{c_{ij} - f_{ij}} \leq \sum_{(i,j) \in p} \frac{1}{c_{ij} - f_{ij}}$ for any other path p for (s, t). The facts above show that the proportional balance vector $\mathbf{f}$ *not only* minimizes the average packet queueing delay of (s, t) for all $s, t \in \mathcal{N}$, *but also* minimizes the average delay over all the links.

If a network is running with low utilization, then $f_{ij} \ll c_{ij}$, and therefore, the delay $\frac{1}{c_{ij} - f_{ij}} \approx \frac{1}{c_{ij}}$. As such InvCap recommended by Cisco can be suitable. When the link load is non-negligible as compared to its link capacity, we should make some changes.

Example 2: $(\mathbf{c}, 2)$ proportional load balance.

If a traffic distribution $\mathbf{f}$ is $(\mathbf{c}, 2)$ proportionally load balancing, then it solves (5) with $V_{ij}(s_{ij}) = \frac{-c_{ij}}{c_{ij} - f_{ij}} = -1 - \frac{f_{ij}}{c_{ij} - f_{ij}}$. In this case, we can see that (5) tries to minimize the total average queueing delay by the $M/M/1$ queueing model with respect to optimal link weights $w_{ij} = \frac{c_{ij}}{(c_{ij} - f_{ij})^2}$ for $(i, j) \in \mathcal{J}$.

Example 3: $(\mathbf{d}, 0)$ proportional load balance.

Let d_{ij} be the processing and propagation delay on link (i, j). If a traffic distribution $\mathbf{f}$ is $(\mathbf{d}, 0)$ proportional load balance, then it solves the TE problem (5) with $V_{ij}(s_{ij}) = d_{ij}(c_{ij} - f_{ij}) = d_{ij}c_{ij} - d_{ij}f_{ij}$. In this case, we can see that (5) tries to minimize the total processing and propagation delay, and we have that the optimal link weights $w_{ij} = d_{ij}$ for unsaturated link $(i, j) \in \mathcal{J}$ and $w_{ij} \geq d_{ij}$ for saturated link (i, j). If $d_{ij} = 1$, we have the minimum hop routing.

We use the network topology in Fig. 1 to illustrate these

notions. In Fig.3, (a) and (b) are the link weights and the link utilization versus load balance parameter β, respectively. Detailed numerical results are shown in TABLE I.

Remark 2: Consider $TE(V, \mathcal{G}, \mathbf{c}, \mathbf{D})$ with V_{ij} defined in (11) with $q_{ij} = 1, \forall (i, j) \in \mathcal{J}$, we obtain: the minimum hop routing for each source-destination pair with $\beta = 0$; the shortest average packet delay routing for each source-destination pair with $\beta = 1$; the shortest paths for each source-destination pair which makes the traffic distribution be min-max load balance when $\beta \to \infty$.

C. Utility vs. Proportional Load Balance

In this subsection, we discuss the relation between the proportional load balance and different utility functions. If link (i, j) can choose an amount to pay per unit time, n_{ij}, and receive in return a spare capacity s_{ij} proportionally to n_{ij}, say $s_{ij} = \frac{n_{ij}}{w_{ij}}$, where w_{ij} could be regarded as a charge per unit flow for link (i, j), the utility maximization problem for link (i, j) becomes
$Link_{ij}(V_{ij}; w_{ij})$

$$
\begin{aligned}
\text{maximize} \quad & V_{ij}\left(\frac{n_{ij}}{w_{ij}}\right) - n_{ij} \\
\text{subject to} \quad & n_{ij} \geq 0.
\end{aligned} \tag{13}
$$

Let $\mathbf{n} = (n_{ij}, (i, j) \in \mathcal{J}), \mathcal{D}(\mathbf{n}) = \{(i, j) \in \mathcal{J} : n_{ij} > 0\}$. We define the optimization problem
$Network(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{n})$

$$
\begin{aligned}
\text{maximize}_{\mathbf{f}^t \geq \mathbf{0}} \quad & \sum_{(i,j) \in \mathcal{D}(\mathbf{n})} n_{ij} \log s_{ij} \\
\text{subject to} \quad & \mathbf{c} - \sum_{t \in \mathcal{D}} \mathbf{f}^t = \mathbf{s} \geq \mathbf{0} \\
& \mathbf{B}\mathbf{f}^t = \mathbf{d}^t, \ \forall t \in \mathcal{D}.
\end{aligned} \tag{14}
$$

Note that if $n_{ij} = 1$ for $(i, j) \in \mathcal{J}$, then the solution to $Network(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{n})$ is the proportional load balancing traffic allocation. If $n_{ij}, (i, j) \in \mathcal{J}$, are all integers, then the solution to $Network(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{n})$ can be constructed as follows. For each $(i, j) \in \mathcal{J}$, replace the single link (i, j) by n_{ij} identical sub-links, calculate the proportional load balance allocation over the resulting $\sum_{(i,j) \in \mathcal{J}} n_{ij}$ traffic, and then provide link (i, j) the aggregate spare capacity allocated to its n_{ij} associated sub-links. The load *per unit charge* are then proportional load balancing.

Say that $\mathbf{s}$ solves $Network(\mathcal{G}, \mathbf{c}, \mathbf{D}; \mathbf{n})$ if there exists $(\mathbf{f}^t, t \in \mathcal{D})$ such that $(\mathbf{s}, \mathbf{f}^t, t \in \mathcal{D})$ solves the optimization problem

(14). The corresponding Lagrangian is

$$L_{\text{Net}}(\mathbf{s},\mathbf{f}^t,t\in\mathcal{D};\mathbf{w},\boldsymbol{\nu}^t,t\in\mathcal{D})$$

$$= \sum_{(i,j)\in\mathcal{D}(\mathbf{n})} n_{ij}\log s_{ij} + \mathbf{w}^T(\mathbf{c} - \sum_{t\in\mathcal{D}}\mathbf{f}^t - \mathbf{s})$$
$$\quad - \sum_{t\in\mathcal{D}}\boldsymbol{\nu}^{t^T}(\mathbf{Bf}^t - \mathbf{d}^t)$$

$$= \sum_{(i,j)\in\mathcal{D}(\mathbf{n})}(n_{ij}\log s_{ij} - w_{ij}s_{ij}) - \sum_{(i,j)\notin\mathcal{D}(\mathbf{n})} w_{ij}s_{ij}$$
$$\quad + \sum_{t\in\mathcal{D}}\sum_{(i,j)\in\mathcal{J}}(\nu_j^t - \nu_i^t - w_{ij})f_{ij}^t$$
$$\quad + \sum_{(i,j)\in\mathcal{J}} w_{ij}c_{ij} + \sum_{t\in\mathcal{D}}\boldsymbol{\nu}^{t^T}\mathbf{d}^t.$$

For the general theory of constrained convex optimization [2], it follows that $(\mathbf{s},\mathbf{f}^t,t\in\mathcal{D})$ solves problem (14) if and only if there exist Lagrangian multiplier vectors $\boldsymbol{\nu}^t$ and $\mathbf{w}$ that satisfy:

$$\mathbf{c} - \sum_{t\in\mathcal{D}}\mathbf{f}^t = \mathbf{s}, \qquad \mathbf{Bf}^t = \mathbf{d}^t, \ \forall t\in\mathcal{D}$$
$$\frac{n_{ij}}{s_{ij}} - w_{ij} = 0, \text{ if } (i,j)\in\mathcal{D}(\mathbf{n})$$
$$-w_{ij} = 0, \text{ if } (i,j)\notin\mathcal{D}(\mathbf{n}), s_{ij} > 0$$
$$-w_{ij} \leq 0, \text{ if } (i,j)\notin\mathcal{D}(\mathbf{n}), s_{ij} = 0$$
$$\nu_j^t - \nu_i^t - w_{ij} = 0, \text{ if } f_{ij}^t > 0$$
$$\leq 0, \text{ if } f_{ij}^t = 0.$$

Theorem 3.4: There exist vectors $\mathbf{n} = (n_{ij},(i,j)\in\mathcal{J})$, $\mathbf{w} = (w_{ij},(i,j)\in\mathcal{J})$, and $\mathbf{s} = (s_{ij},(i,j)\in\mathcal{J})$ such that

i) $w_{ij} > 0$ and $n_{ij} = w_{ij}s_{ij}$, for $(i,j)\in\mathcal{J}$;
ii) n_{ij} solves $\text{Link}_{ij}(V_{ij};w_{ij})$ (13), for $(i,j)\in\mathcal{J}$;
iii) $\mathbf{s}$ solves $\text{Network}(\mathcal{G},\mathbf{c},\mathbf{D};\mathbf{n})$ (14).

Given any such triple $(\mathbf{n},\mathbf{w},\mathbf{s})$, the vectors $\mathbf{n}$ and $\mathbf{s}$ are uniquely determined, and $\mathbf{s}$ solves $\text{TE}(V,\mathcal{G},\mathbf{c},\mathbf{D})$.

Proof: Let $(\mathbf{s},\mathbf{f}^t,t\in\mathcal{D})$ be a solution of (5). Let $\mathbf{w}$ and $(\boldsymbol{\nu}^t,t\in\mathcal{D})$ be the Lagrangian multiplier vectors, i.e. the conditions of (6a)-(6e) are satisfied. For $V'(s_{ij}) > 0$, we have $w_{ij} \geq V'_{ij}(s_{ij}) > 0$ from (6b) and (6c). Let $n_{ij} = w_{ij}s_{ij},(i,j)\in\mathcal{J}$. Then i) of Theorem 3.4 holds.

As $w_{ij} > 0$ for all $(i,j)\in\mathcal{J}$, $\text{Link}_{ij}(V_{ij};n_{ij})$ is well defined. In addition, it is obvious that $n_{ij} = 0$ if and only if $s_{ij} = 0$. We have that n_{ij} solves $\text{Link}_{ij}(V_{ij};w_{ij})$, for $(i,j)\in\mathcal{J}$ by (6b) and (6c). Then ii) of Theorem 3.4 holds.

By i) of Theorem 3.4, we have the fact that $\mathcal{D}(\mathbf{n}^*) = \{(i,j)\in\mathcal{J}: n_{ij} > 0\} = \{(i,j)\in\mathcal{J}: s_{ij} > 0\}$. Then the condition (III-C) for $\text{Network}(\mathcal{G},\mathbf{c},\mathbf{D};\mathbf{n})$ is disappear. By the definition of $\mathbf{n}$ and $\mathbf{w}$, the conditions (III-C) and (III-C) hold. The other conditions follow from the conditions (6a), (6d) and (6e). We have the fact that $\mathbf{s}$ solves $\text{Network}(\mathcal{G},\mathbf{c},\mathbf{D};\mathbf{n})$. Then iii) of Theorem 3.4 holds.

Conversely, given $\mathbf{n},\mathbf{w}$ and $\mathbf{s}$ satisfying conditions i)-iii) of Theorem 3.4, then by i), $\text{Link}_{ij}(V_{ij};w_{ij})$ is well defined for all $(i,j)\in\mathcal{J}$. In addition, we have that $n_{ij} = 0$ if and only if $s_{ij} = 0$. By ii) and $s_{ij} = \frac{n_{ij}}{w_{ij}}$, the conditions (6b) and (6c) hold for given $\mathbf{s}$ and $\mathbf{w}$. By iii), there exist $\mathbf{f}^t$ and $\boldsymbol{\nu}^t$ for all $t\in\mathcal{D}$ satisfying conditions (III-C), (III-C) and (III-C). We can verify that $(\mathbf{s},\mathbf{f}^t,t\in\mathcal{D},\mathbf{w},\boldsymbol{\nu}^t,t\in\mathcal{D})$ satisfies (6a)-(6e). We conclude $\mathbf{s}$ solves $\text{TE}(V,\mathcal{G},\mathbf{c},\mathbf{D})$ and therefore $\mathbf{s}$ is uniquely determined. Since $w_{ij} > 0$ for all $(i,j)\in\mathcal{J}$, $\mathbf{n}$ is uniquely determined as well. ■

Since Theorem 3.4 is straightforward, here we do not present the detailed proof. It shows that if each link operator is able to choose a charge per unit time prepares to pay. And if the network allocates spare capacities so that the spare capacity per unit charge is proportional load balancing, then a system optimum is achieved when the link operator's choices of charges and the network's choice of allocated spare capacities are in equilibrium.

IV. A NEW ROUTING PROTOCOL: SPEF

We are now in a position to design a new routing protocol based on the above theoretical results. In the following, we first present the distributed algorithms to achieve the optimal link weights, also called the first link weight. And then we derive the second link weights from the conceptual framework Network Entropy Maximization [20].

In the Shortest paths Penalizing Exponential Flow-splitting (SPEF), each router can construct the shortest paths for each destination based on the first link weights and independently calculate the traffic split ratio among all equal-cost shortest paths using only the second link wights, *not only* achieves the optimal traffic engineering *but also* remains the path diversity.

A. Obtaining the First Link Weights

We now show a distributed algorithm to obtain the first link weights, which in fact is the sub-gradient projection method [4] applied to the dual of $\text{TE}(V,\mathcal{G},\mathbf{c},\mathbf{D})$. The algorithm comprises three parts: updating the weight vector, specifying the spare capacity and modifying the routing variables, as described in *Algorithm 1*.

Algorithm 1 *Dual decomposition for the first link weights*

Given tolerance *tol* and initial weight $\mathbf{w}^{(0)}$ (such as $w_{ij}^{(0)} = 1/c_{ij}$), $k = 0$;
for the given weight $\mathbf{w}^{(k)}$ **do**
  Each link (i,j) solves $\text{Link}_{ij}(V_{ij};w_{ij}^{(k)})$ to find
    the spare capacity $s_{ij}^{(k)}$;
  Each destination $t\in\mathcal{D}$ solves $\text{Route}_t(\mathbf{w}^{(k)};\mathbf{d}^t)$:

$$\begin{array}{cl} \text{minimize}_{\mathbf{f}^t\geq\mathbf{0}} & \sum_{(i,j)\in\mathcal{J}} w_{ij}^{(k)} f_{ij}^t \\ \text{subject to} & \mathbf{Bf}^t = \mathbf{d}^t \end{array} \quad (15)$$

    to find the routing variable $\mathbf{f}^{t\,(k)}$;
  Each link $(i,j)\in\mathcal{J}$ updates the link weight

$$w_{ij}^{(k+1)} = \left(w_{ij}^{(k)} - \gamma_k(c_{ij} - \sum_{t\in\mathcal{D}} f^{t\,(k)}_{ij} - s_{ij}^{(k)})\right)_+; \quad (16)$$

  $k \leftarrow k+1$;
  Until $\text{gap}(\mathbf{w}^{(k)},\mathbf{s}^{(k)},\mathbf{f}^{(k)}) < tol$.
end for

Given the link weight $\mathbf{w}$, the route problem (15) for each destination is a minimum-cost network flow problem [3]. In (16), γ_k is the step size and $(z)_+ = \max(0,z)$. And optimality measure is defined as the dual gap, *i.e.*

$$\text{gap}(\mathbf{w}^{(k)},\mathbf{s}^{(k)},\mathbf{f}^{(k)}) = \sum_{(i,j)\in\mathcal{J}} w_{ij}^{(k)}(\sum_{t\in\mathcal{D}} f^{t\,(k)}_{ij} + s_{ij}^{(k)} - c_{ij}).$$

Theorem 4.1: The link weight sequence $\{\mathbf{w}^{(k)}\}$ generated by Algorithm 1 converges to the first link weights $\mathbf{w}^*$ if $\sum_k \gamma_k = \infty$ and $\gamma_k \to 0$. Furthermore, if there are no saturated links, *i.e.* $s_{ij}^* > 0, \forall (i,j) \in \mathcal{J}$, the first link weights $\mathbf{w}^*$ is uniquely determined and the optimal traffic distribution is $\mathbf{f}^* = \mathbf{c} - \mathbf{s}^*$, where $s_{ij}^* = V_{ij}'^{-1}(w_{ij}^*)$.

We have proposed a link weight configuration method that can achieve the optimal traffic engineering. We can determine the set of shortest paths $\mathrm{ON} = \{\mathrm{ON}^t : t \in \mathcal{D}\}$ (*i.e.*, deciding which outgoing link should be chosen on the shortest path) based on the first link weights, where ON^t is the shortest path set for any node $s \in \mathcal{N}$ to destination $t \in \mathcal{D}$. Specifically, SP^r denotes the shortest path set for (s_r, t_r). Let $\mathrm{SP} = \{\mathrm{SP}^r : r \in \mathcal{R}\}$. When the first link weights generate multiple equal cost paths for a source-destination pair or next hops for a given destination routing prefix, we need to split the traffic among the multiple shortest paths or the next hops to keep paths diversity while achieving the optimal traffic engineering.

B. Obtaining the Second Link Weights

Motivated by PEFT [20], we propose an Exponential-weighted flow split in the presence of multiple equal cost paths for a given ingress and egress pair (s_r, t_r). The proposed method features that each router can *independently* compute the flow split only based on alternative link weights, where routers can direct traffic on the shortest paths determined by the first link weights. This method can achieve network-wide traffic engineering objective through OSPF which still keeps the simplicity and scalability of link-state routing protocols.

We maximize the relative entropy of the traffic split vector among the multipath in SP^r to maintain the path diversity. Maximizing the relative entropy [4] of the traffic split vector can be formulated as follows.

NEM($\mathrm{SP}, \mathbf{f}, \mathbf{D}$):

$$\text{maximize} - \sum_{r \in \mathcal{R}} d_r \sum_{k=1}^{n_r} p_k^r \log p_k^r \tag{17a}$$

$$\text{subject to} \sum_{r \in \mathcal{R}} \sum_{k:(i,j) \in \mathrm{SP}_k^r} d_r p_k^r \leq f_{ij}^*, \forall (i,j) \in \mathcal{J} \tag{17b}$$

$$\sum_{k=1}^{n_r} p_k^r = 1, \ \forall r \in \mathcal{R}, \tag{17c}$$

where n_r denotes the number of the shortest paths from s_r to t_r. SP_k^r denotes the k-th shortest path from s_r to t_r.

We will connect the characterization of optimal solution to NEM that is realizable with hop-by-hop forwarding to exponential penalty. Let $(p^r, r \in \mathcal{R})$ be a solution of (17). Then there exist Lagrangian multipliers vector $\mathbf{v} = (v_{ij}, (i,j) \in \mathcal{J})$ and $(\nu_r, r \in \mathcal{R})$ satisfying that $(1 + \log p_k^r) + \sum_{(i,j) \in \mathrm{SP}_k^r} v_{ij} + \frac{\nu_r}{d_r} = 0, \ \forall r \in \mathcal{R}, k$ and $\sum_{k=1}^{n_r} p_k^r = 1$. Under these conditions, we have

$$p_k^r = \frac{e^{-v_k^r}}{\sum_{i=1}^{n_r} e^{-v_i^r}}, \ \forall r \in \mathcal{R}, k, \tag{18}$$

where $v_k^r = \sum_{(i,j) \in \mathrm{SP}_k^r} v_{ij}$. In the following, we refer to Lagrange multipliers vector $\mathbf{v}$ as the second link weight. v_k^r is

the length of path SP_k^r with respect to the second weight $\mathbf{v}$.

Theorem 4.2: The optimal traffic engineering for a given traffic can be realized with the second link weights using exponential flow split (18).

To provide a foundation for the second link weight computation, we investigate the Lagrange dual problem of NEM($\mathrm{SP}, \mathbf{f}, \mathbf{D}$) and a dual-gradient-based solution. Denote the dual variables for constraints (17b) as v_{ij} for link (i,j) (or $\mathbf{v}$ as a vector). We first write the Lagrangian $L(\mathbf{p}, \mathbf{v})$ associated with problem NEM($\mathrm{SP}, \mathbf{f}, \mathbf{D}$) as

$$L(\mathbf{p}, \mathbf{v}) = -\sum_{r \in \mathcal{R}} d_r \sum_{k=1}^{n_r} (p_k^r \log p_k^r + v_k^r p_k^r) + \sum_{(i,j) \in \mathcal{J}} v_{ij} f_{ij}^*,$$

where $v_k^r = \sum_{(i,j) \in \mathrm{SP}_k^r} v_{ij}$. The Lagrange dual function is

$$d(\mathbf{v}) = \underset{}{\text{maximize}} \quad L(\mathbf{p}, \mathbf{v})$$
$$\text{subject to} \quad \sum_{k=1}^{n_r} p_k^r = 1, \ \forall r \in \mathcal{R}.$$

The dual problem is then formulated as

$$\text{minimize } d(\mathbf{v}) \text{ subject to } \mathbf{v} \geq \underline{0}. \tag{19}$$

To solve the dual problem, we first consider the maximization of the Lagrangian over $\mathbf{p}$. Note that, the $L(\mathbf{p}, \mathbf{v})$ is separable for a given dual variable $\mathbf{v}$, *i.e.*, the traffic split subproblem for each $r \in \mathcal{R}$ is independent of the others since they are not coupled together with link capacity constraint (17b). So we can solve a subproblem (20) below for each $r \in \mathcal{R}$ separately:

$$\text{maximize} \quad -d_r \sum_{k=1}^{n_r} \left(p_k^r \log p_k^r + v_k^r p_k^r \right)$$
$$\text{subject to} \quad \sum_{k=1}^{n_r} p_k^r = 1. \tag{20}$$

Then, the dual problem (19) can be solved by using the gradient projection method as follows for iterations indexed by k,

$$v_{ij}^{(k+1)} = \left(v_{ij}^{(k)} - \gamma (f_{ij}^* - \sum_{r \in \mathcal{R}} d_r \sum_{l:(i,j) \in \mathrm{SP}_l^r} p_l^{r \, (k)}) \right)_+$$
$$= \left(v_{ij}^{(k)} - \gamma (f_{ij}^* - f_{ij}^{(k)}) \right)_+ \tag{21}$$

where $\gamma > 0$ is a constant step size, $(p_1^{r \, (k)}, \cdots, p_{n_r}^{r \, (k)})$ are solutions of the traffic split subproblem (20) for $\mathbf{v}^{(k)}$, and $f_{ij}^{(k)}$ is the total flow on link $(i,j) \in \mathcal{J}$.

It is important to note, from (21) in iteration $k + 1$, the procedure of link weight updating needs $f_{ij}^{(k)}$, the aggregate bandwidth usage. We now show how to calculate it efficiently.

First, we need to establish the forward table for node s to destination t as shown in Table II, where n_k denotes the

number of shortest path from node s through node v_k to node t, $v_{kj}^{(s,t)}$ is the length of the j-th path from node s through node v_k to node t, and m_s denotes the number of next hop for s in ON^t. Then the traffic to destination t can be splited according to the following formula:

$$\Gamma^t(s, v_k) = \frac{\sum_{j=1}^{n_k} e^{-v_{kj}^{(s,t)}}}{\sum_{i=1}^{m_s} \sum_{j=1}^{n_i} e^{-v_{ij}^{(s,t)}}}, \quad k = 1, \cdots, m_s. \quad (22)$$

Finally, the formal algorithm for the second link weights can be described as follows, in which Algorithm 3 is needed to get the traffic distribution matching to the current second link weights $\mathbf{v}^{(k)}$.

Algorithm 2 *Dual decomposition for the second link weights*

Input the optimal traffic distribution $\mathbf{f}^*$ and tolerance ϵ;
Given the initial second link weights $\mathbf{v}^{(0)} = \mathbf{0}, k = 0$;
For the given weights $\mathbf{v}^{(k)}$, **do**
  Get the traffic distribution matching to $\mathbf{v}^{(k)}$, i.e.

$$\mathbf{f}^{(k)} \leftarrow \text{TrafficDistribution}(\mathbf{v}^{(k)}).$$

  Each link (i, j) updates the second link weights

$$v_{ij}^{(k+1)} \leftarrow \left(v_{ij}^{(k)} - \gamma(f_{ij}^* - f_{ij}^{(k)})\right)_+;$$

$k \leftarrow k + 1$;
Until $f_{ij}^{(k)} \leq f_{ij}^* + \epsilon$ for all $(i, j) \in \mathcal{J}$.

Algorithm 3 *TrafficDistribution($\mathbf{v}$)*

Input $\mathrm{ON} = \{\mathrm{ON}^t : t \in \mathcal{D}\}$;
Compute the path length for each path in ON
    in view of the second link weights $\mathbf{v}$;
Compute the traffic split $\Gamma^t(i, j)$ according to (22);
For each destination t **do**
  Do sorting on the distance of node s to t
    in view of the first link weights
  Each source $s \neq t$ in the decreasing distance order **do**
    $\bar{d}_{st} = d_{st} + \sum_{(j,s) \in \mathrm{ON}^t} f_{js}^t$;
  For all j such that $(s, j) \in \mathrm{ON}^t$
    $f_{sj}^t = \bar{d}_{st} \Gamma^t(s, j)$;
  end for
end for
$f_{ij} = \sum_{t \in \mathcal{D}} f_{ij}^t$ for all $(i, j) \in \mathcal{J}$;
Return /* set of $\mathbf{f}$ */

Here $\bar{d}_{st}$ denotes the total incoming flow destined to node t at node s (including traffic originating at s as well as any traffic arrived from other nodes).

The following result can be proved with standard convergence analysis for gradient projection algorithms [2]:

Theorem 4.3: Let $\{\mathbf{v}^{(i)}\}$ be the sequence generated by Algorithm 2. We have that $\{\mathbf{v}^{(i)}\}$ converges to the optimal dual solutions $\mathbf{v}^*$, and the corresponding primal variables $\mathbf{p}^*$ according to (18) are the globally optimal solution of (17).

We now present a new link-state routing with hop-by-hop forwarding, which can achieve the optimal traffic engineering.
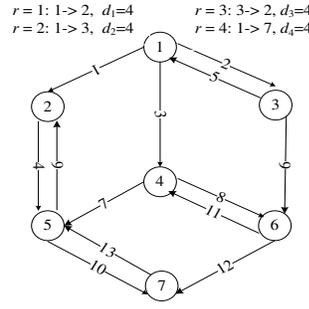

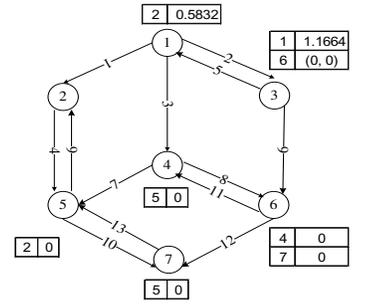Fig. 4. A simple network topology and traffic demands

Fig. 5. SPEF forwarding table for destination 2

Algorithm 4 *SPEF routing*

Running Algorithm 1 to obtain the first link weights $(w_{ij}, (i, j) \in \mathcal{J})$ and optimal traffic distribution $\mathbf{f}^*$.
For each destination node $t \in \mathcal{D}$ **do**
  Run Dijkstra's algorithm with the first link weights
    to get all the shortest paths $\mathrm{ON} = \{\mathrm{ON}^t : t \in \mathcal{D}\}$.
end **for**.
Running Algorithm 2 to obtain the second link weights $(v_{ij}, (i, j) \in \mathcal{J})$.
For each $t \in \mathcal{D}$ **do**
  For each source node s:
    Establish the forward routing table shown in Table II.
 **end For**
end For

V. PERFORMANCE EVALUATION

How well can the new routing protocol SPEF perform? In the first part, we will illustrate its performance with a simple example. In the second part, we demonstrate the performance of SPEF with numerical experiments over a real backbone network and several synthetic networks. Here we compare the results of SPEF with that of OSPF, which sets link weight inversely proportional to its capacity and evenly splits the traffic over multiple equal-cost shortest paths.

A. An Example

Fig. 4 shows a simple network topology, as used in [19]. Each link has a capacity of 5 units and each demand needs a bandwidth of 4 units. For simplicity, we omit six links unused. The numbers on the links are the link indices.

The link utilizations for optimal TE with a different parameter β are shown in Fig. 6. For the results of $\beta = 0$, link 1 is a bottle link. And the first link weight is 3. The first link weight of others are all 1. Considering link 1, the link utilization is decreasing in β. From Eq. (6b), the first weight of links 2 and 3 are the same when $\beta = 0, 1$ or 5, since all the spare capacities are equal to 1. For $\beta = 1$, from Fig.7 (b), it can be seen that all the second link weights are zero except for link 1 and link 5. The fact that the second weight of link 1 is increasing in β shows we route fewer traffic through link 1 with larger β.
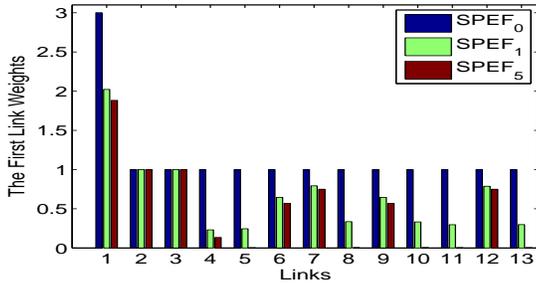
(a) The first link weights
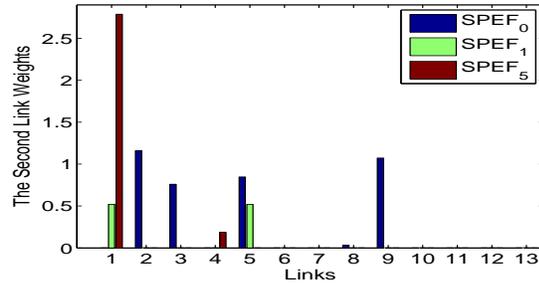
(b) The second link weights

Fig. 7. The first and second link weights for the network topology shown in Fig. 4 with different β
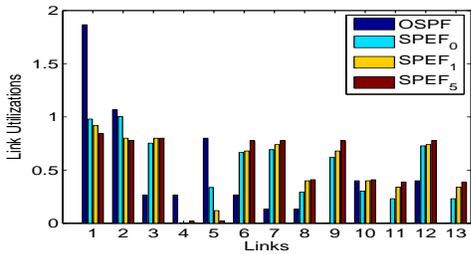

Fig. 6. The link utilization for the topology shown in Fig.4

TABLE III
PROPERTIES FOR DIFFERENT NETWORKS

Net. ID	Topology	Node #	Link #
Abilene	Backbone	11	28
Cernet2	Backbone	20	44
Hier50a	2-level	50	222
Hier50b	2-level	50	152
Rand50a	Random	50	242
Rand50b	Random	50	230
Rand100	Random	100	392

B. Simulation Environment

The properties of the networks used are summarized in TABLE III. The real backbone network, the Abilene network and Cernet2 network shown in Fig.8. The first network has 11 nodes and 28 directional links with 10Gbps capacity, and the latter has 20 nodes and 44 directional links with 10Gbps capacity for 4 backbone links and 2.5Gbps for others. The traffic demands for Abilene network is generated as those in Fortz and Thorup [16]. The traffic demands for Cernet2 network are generated by a gravity model with the link aggregated load extracted from the sample Netflow data, which was captured during 2010/1/10 to 2010/1/16. To simulate networks with different congestion levels, we create different test cases by uniformly increasing the traffic demands until the maximal link utilization almost reaches 100% with SPEF.

We also test the algorithms proposed in this paper on the same topologies and traffic matrices as in Fortz and Thorup [16]. The 2-level hierarchical networks were generated using GT-ITM, which consist of two kinds of links: local access links with 1 unit capacity and long distance links with 5-unit capacity. In the random topologies, the probability of having a link between two nodes is a constant parameter, and all link capacities are 1 unit. In these test cases, for each network, traffic demands are proportionally increased to simulate different congestion levels.

For SPEF, we employ the utility function with $\beta = 1$ to determine the first link weights. The utility is normalized, which means $\sum_{(i,j)\in\mathcal{J}} \log(1 - u_{ij})$, where u_{ij} is the link (i,j)'s utilization. The utility is $-\infty$ if MLU is greater than 1, which is not shown in Fig. 10.

C. Performance Comparison Against OSPF

The sorted link utilizations for Abilene network and Cernet2 network are shown in Fig. 9, where the network load is the ratio of total demand over the total capacity. Typical results for different topologies are shown in Fig.10.

From Fig. 9, it can be seen that some underutilized links in OSPF are used efficiently in SPEF. At the same time the traffic on the over-utilized links in OSPF is removed in SPEF. The results shown in Fig. 10 indicate that the utility difference between SPEF and OSPF becomes obvious with the increasing of network load. SPEF still works when MLU of OSPF is greater than 1.

D. Performance Comparison Against PEFT

In order to make a comparison between SPEF and PEFT in network environment, we resort to SSFnet, a highly efficient simulation tool, to explore the protocol behaviors in networks with different scales. In the simulation, both the simple network in Fig. 4 and Cernet2 backbone network in Fig. 8(b) are used, and the traffic demands are shown in TABLE IV. Accordingly, it can be divided into two parts.

First, SPEF and PEFT separately run for 400s on the simple network with each link capacity set to be 5Mb/s. The mean traffic load on each link is shown in Fig. 11(a), of which the X axis represents link index and the Y axis is the mean traffic load in kbps. The result shows that in PEFT altogether 8 links are used for carrying traffic, and the link loads vary severely from 1000 kbps to near 3000 kbps. Comparing to PEFT, 4 more links are involved in SPEF and the traffic load is more equally distributed among these links.
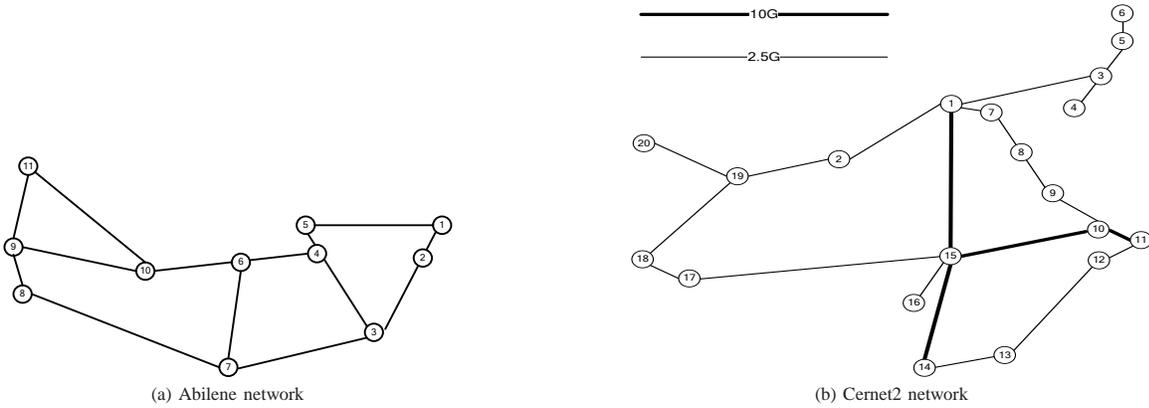
(a) Abilene network

(b) Cernet2 network
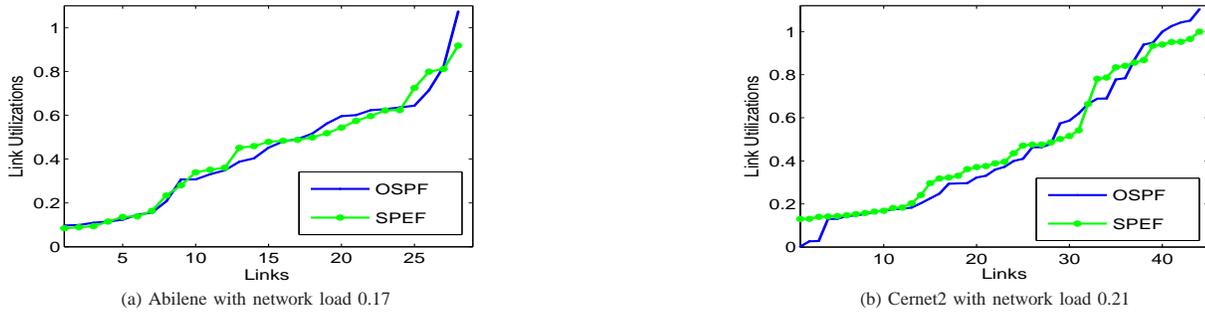
Fig. 8.   Backbone network topologies

(a) Abilene with network load 0.17

(b) Cernet2 with network load 0.21

Fig. 9.   Comparison of SPEF and OSPF in terms of the sorted link utilization

(a) Abilene

(b) Cernet2

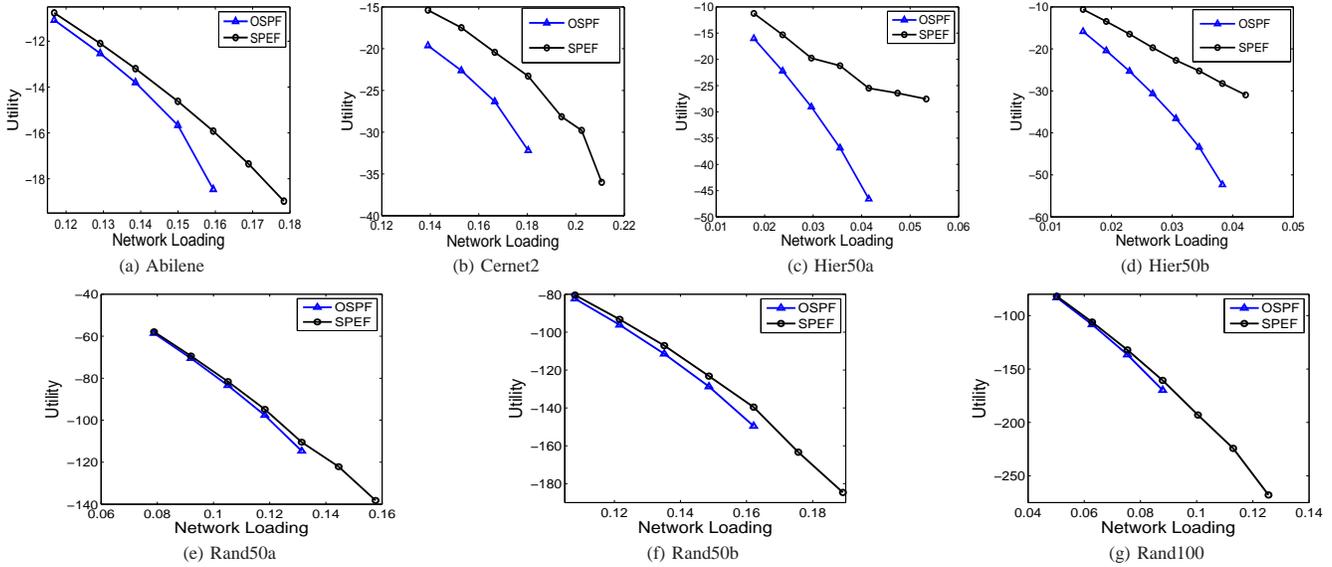(c) Hier50a

(d) Hier50b

(e) Rand50a

(f) Rand50b

(g) Rand100

Fig. 10.   Comparison of SPEF and OSPF in terms of utility

Second, we run SPEF and PEFT for 400s on the Cernet2 backbone network, where the capacity of 4 links marked with bold lines is 10Gbps, four times larger than that of the rest links. The mean traffic load on each link is shown in Fig. 11(b). The meaning of both X axis and Y axis are the same with those in Fig. 11(a), except that the link load is measured in Mbps. Three more links is used in SPEF than that in PEFT

and hence the variation of link load is lower.

In the above simulation cases, SPEF always leverages more links for packet delivery than PEFT, which can be explained by looking deeply into the forwarding tables of each protocol. Although in PEFT traffic can be intuitively split over all possible paths between the source and destination, the penalizing exponential flow-splitting mechanism prefers the shortest path

Net.ID.	Src.ID	Dst.ID	Demand #
Simple network in Fig. 4	1	2	4Mb
	1	3	4Mb
	3	2	4Mb
	1	7	4Mb
Cernet2 Backbone in Fig. 8(b)	11	1	3Gb
	11	2	2Gb
	11	20	2Gb
	13	6	1Gb
	14	1	4Gb
	14	8	2Gb

Routing	Network loading	n_1	n_2	n_3	n_4
OSPF	0.13, 0.17, 0.21	355	25	0	0
SPEF	0.13	330	48	0	2
	0.17	325	53	0	2
	0.21	321	54	3	2

while penalizing the longer paths. In SPEF, however, multiple equal-cost shortest paths for the same source-destination pair are constructed with a higher probability based on the first link weight, and then traffic is split over these paths according to the exponential ratios computed by the second link weight. The comparison result infers that SPEF outperforms PEFT in terms of load balance.

E. Equal Cost Paths

One of the key features of SPEF routing is the ability to balance traffic across multiple equal-cost paths. Intuitively, SPEF routing is more likely to use multiple paths to balance traffic at higher loads. Hence, we focus on a different utilization scenarios for Cernet2 network, for which we compute the number of equal cost paths used by SPEF routing. TABLE V shows the results, where n_i denotes the number of ingress-egress pairs that have i equal cost paths. It can be seen that the equal cost paths for some ingress-egress pairs are increasing with the increase of network load. But OSPF routing has not change with the network load.

F. Convergence Behavior

In Algorithm 1, the initial link weights $w_{ij}^{(0)} = \frac{1}{c_{ij}}$ for all link $(i,j) \in \mathcal{J}$ are a proper choose. The step sizes in Algorithm 1 can be constant or dynamically adjusted. We find that setting the step size in Algorithm 1 to the reciprocal of the maximum link capacity $\frac{1}{\max\{c_{ij}:(i,j)\in\mathcal{J}\}}$ performs well in practice. Fig. 12 (a) shows the evolution of dual objective value of TE obtained by Algorithm 1 with different step sizes, within the first 2000 iterations for Cernet2 network. It provides convergence behavior typically observed. The legends show the ratio of the step size over the default setting which is

$\frac{1}{\max\{c_{ij}:(i,j)\in\mathcal{J}\}}$. It demonstrates that Algorithm 1 developed for the SPEF routing convergence very fast with default setting. Algorithm 1 reduces the dual objective value of TE to -48 after 100 iterations and -49 after 500 iterations. In addition, increasing step size a little will speed up the convergency, and as expected, too large a step size (e.g., 2 for Algorithm 1 in the above example) would cause a little oscillation. Notice that there is a wide range of step size that can make convergence very fast.

In Algorithm 2, the initial link weights $v_{ij}^{(0)} = 0$ for all link $(i,j) \in \mathcal{J}$ are a proper choose. We find that setting the step size in Algorithm 2 to the reciprocal of the maximum optimal traffic distribution $\frac{1}{\max\{f_{ij}^*:(i,j)\in\mathcal{J}\}}$ performs well in practice. Fig. 12 (b) shows evolution of dual objective value of NEM obtained by Algorithm 2 with different step sizes for Cernet2 network. It provides convergence behavior typically observed. The legends show the ratio of the step size over the default setting which is $\frac{1}{\max\{f_{ij}^*:(i,j)\in\mathcal{J}\}}$. It demonstrates that the initial link weights for Algorithm 2 are a good approximation solution for the dual problem of NEM. And Algorithm 2 developed for the SPEF routing also convergence very fast with default setting. Algorithm reduces the dual objective value of NEM to 0.6695 after 100 iterations and 0.66945 after 300 iterations. In addition, increasing step size a little will speed up the convergency. Notice that there is a wide range of step size that can make convergence very fast.

G. Noninteger Link Weights

We should point out that Algorithm 1 is not guaranteed to yield exact integer solutions for the first link weights. In practice, routing protocols link OSPF and IS-IS have a finite field width for link weight information. To guarantee the weight for the link with maximum spare capacity is 1, we get the integer weights as following

$$w_{ij}' = \text{round} \quad w_{ij}\left(\max\{s_{ij}:(i,j)\in J\}\right), \forall(i,j)\in\mathcal{J},$$

where $\text{round}(x)$ means rounding x to the nearest integer and w_{ij} is the first link weight obtained with Algorithm 1. This is because the modified link weights can result in routings that are different from the optimal routing. Hence it is important to study how errors in link weights influence performance.

There are two factors that can introduce inaccuracies in link weights. First of course is the precision in rounding off link weights. The second factor is the nonzero tolerance required by Dijkstra's algorithm. This implies that the optimal link weights (and path cost) are accurate only within a certain tolerance. For example, two path costs are treated to be equal by Dijkstra's algorithm if the difference in costs is less than the specified tolerance. We specify the tolerances for the Dijkstra's algorithm are 0.3 and 1 for noninteger weights and integer weights respectively.

Fig.13 shows the impact of integer weights on utility for Abilene network and Cernet2 network. Observe that the integer weights has little impact on utility for the low network loading. At higher network loadings, errors due to integer
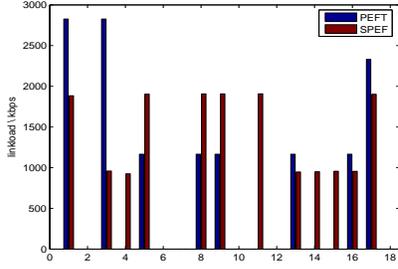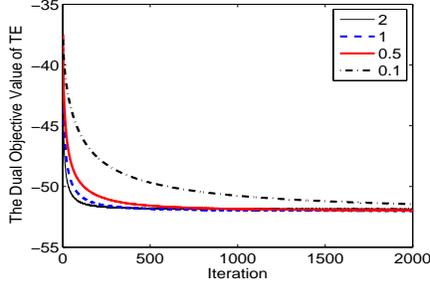
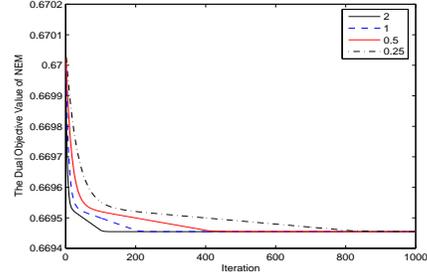(a) Link load distribution over the simple network in Fig. 4
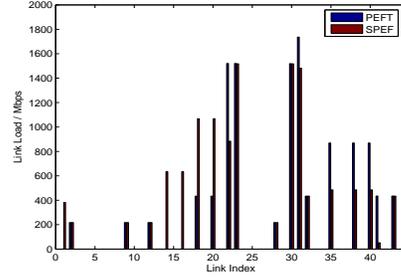
(b) Link load distribution over the Cernet2 network in Fig. 8(b)

Fig. 11. Simulation results of SPEF and PEFT using SSFnet over different networks

(a) Evolution of dual objective value of Algorithm 1

(b) Evolution of dual objective value of Algorithm 2

Fig. 12. Evolution of dual objective value obtained by Algorithm 1 and Algorithm 2 with different step sizes for Cernet2 network

tolerances comes into play so that the utility starts to deviate significantly. This is because, the first link weights will be increasing with the link load while the tolerance for the Dijkstra's algorithm is specified. In order to avoid problems due to such errors, we use the different tolerance for the Dijkstra's algorithm with the different network loading.

VI. RELATED WORK

Among the papers focused on TE, MLU [19] and piecewise-linear approximation of the $M/M/1$ delay formula [14] are two frequently-used cost functions. Minimizing MLU ensures that the traffic is moved away from congested hot spots to less utilized parts of the network. The latter formula proposed by Fortz et al. [14] is based on discussions with the technicians in AT&T Lab. A description of the general infrastructure behind this kind of traffic engineering is given in [15]. Srivastava et al. [18] constructed a composite cost function which was a positive linear combination of the used capacity and MLU, and then proposed a heuristic hybrid method combining the sub-gradient projected method and a genetic algorithm to determine the link weight system. Di Yuan [23] proposed an approach for robust OSPF routing using an artificial objective function embedded into a local search algorithm.

In the research group of congestion control, researchers are mainly concerned with fairness and efficiency. Network utility maximization (NUM) [5], especially the proportionally fair, is a trade-off objective for this aim. In addition, to design the end-to-end algorithms for joint routing and rate control, many following researches replace capacity constraints with barrier functions that specify the congestion cost at the link (e.g., [5],

[7] and [8]). Generally, a function $\Phi_{ij}(f_{ij}) = \int_0^{f_{ij}} p_{ij}(u)\mathrm{d}u$ is defined, which can be regarded as a penalty function that describes the rate at which the cost is incurred at resource (i, j) with capacity $c(i, j)$ when the load through it is f_{ij}. He et al. [11] choose $\Phi(f_{ij}, c_{ij}) = e^{\frac{f_{ij}}{c_{ij}}}$ to model $M/M/1$ queuing delay, which is related to the price function $p_{ij}(f_{ij}) = \frac{1}{c_{ij}}e^{\frac{f_{ij}}{c_{ij}}}$. Xu et al. in [21] defined $\Phi_{ij}(f_{ij}) = -q_{ij}\ln(c_{ij} - f_{ij})$, which is related to the price function $p_{ij}(f_{ij}) = \frac{q_{ij}}{c_{ij}-f_{ij}}$.

Based on the results derived from linear programming, Wang et al. [19] proved that any arbitrary set of routes can be converted to shortest-paths with respect to some set of positive link weights. This implies that the shortest path limitation is in itself not a major hurdle. Fortz et al. [16] showed that optimizing the link weights for OSPF with evenly split over ECMP to the offered traffic is an NP-hard problem and proposed a local search heuristic. Sridharan et al. [17] used a centralized greedy computation to select the subset of next-hops for each prefix to attain load balance much better than even splitting among the shortest paths. But these solutions fail to enable routers to independently compute the flow-splitting ratios only using link weights. PEFT, recently proposed by Xu et al. [20], is a promising link-state routing protocol splitting traffic over multiple paths with an exponential penalty on longer paths. In order to prevent loops and promote computational efficiency, PEFT used *Downward PEFT* for traffic splitting, which does not provably achieve optimal TE [20]. PEFT shed a new light for further studies on developing an OSPF-based protocol, which enables routers to independently make traffic split decisions while maintaining the shortest paths.
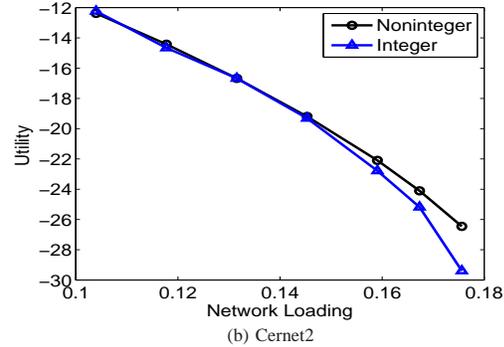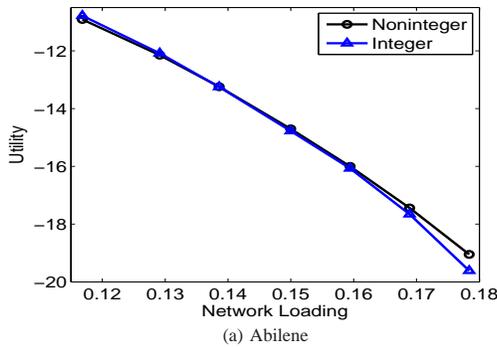
| (a) Abilene | (b) Cernet2 |

Fig. 13.   Impact of integer weights on performance

VII. CONCLUSIONS

In this paper, we explore the problem of achieving the optimal traffic engineering in intra-domain IP networks. Firstly, we propose a new generic objective function, where various interests of providers can be extracted with different parameter settings. And then, we model the optimal TE as the utility maximization of multi-commodity flows and theoretically show that any given set of optimal routes corresponding to a particular objective function can be converted to shortest paths with respect to a set of positive link weights, which can be directly configured on OSPF-based protocols. On these bases, we develop a new OSPF-based routing protocol, SPEF, to realize a flexible way that splits traffic over shortest paths in a distributed fashion. The inspiring fact lies that comparing to OSPF, SPEF only needs one more weight for each link and provably achieves optimal TE. Numerical experiments have been done to compare SPEF with the current version of OSPF, showing the effectiveness of SPEF in terms of link utilization and network load distribution.

A direction for further studies is that we should analyze the computational complexity in network environment with OSPF as well as other existing approaches including PEFT.

REFERENCES

[1] D. P. Bertsekas and R. Gallager. *Data Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] D. P. Bertsekas. *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[3] J. Z. Zhang and S. J. Xu. *Linear programming*, Beijing, P.R. China:Science Publishing House, 1990.

[4] S. Boyd, L. Vandenberghe. *Convex Optimization*, Cambridge University Press: Cambridge,2004.

[5] F. Kelly, A. Maulloo and D. Tan. "Rate control for communication networks: shadow prices, proportional fairness and stability", *J. Operations Res. Soc.*, vol. 49, No.3, pp.237-252, 1998.

[6] F. P. Kelly. "Charging and rate control for elastic traffic (with correction)", *Eur Trans. on Telecommun*, vol.8, pp.33-37, 1997.

[7] F. P. Kelly and T. Voice. "Stability of end-to-end algorithms for joint routing and rate control", *Computer Communication review*, vol. 35, no. 2, pp. 5-12, 2004.

[8] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and Don Towsley. "Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet", *IEEE/ACM Trans. on Networking*, vol. 14, no. 6, pp.1260-1271, Dec. 2006.

[9] S. Low. "A duality model of TCP and queue management algorithms", *IEEE/ACM Trans. Networking*, vol. 11, no.4, pp.525-536, Aug. 2003.

[10] J. Mo, J. Walrand. "Fair end-to-end window-based congestion control", *IEEE/ACM Trans. on Networking*, vol.8, no.5, pp.556-567, Oct. 2000.

[11] J.He, M. Suchara and M. Chiang. "Rethinking Internet traffic management: from multiple decompositions to a practical protocol", In *CoNEXT'07*, New York:2007.

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

[13] Cisco, Configuring OSPF, 1997.

[14] B. Fortz and M. Thorup. "Internet traffic engineering by optimizing OSPF weights", *Proc. IEEE INFOCOM, 2000*, Tel Aviv, Israel, pp. 519 - 528.

[15] B. Fortz, J. Rexford, and M. Thorup. "Traffic engineering with traditional IP routing protocols", *IEEE Communications Magazine*, vol. 40, no. 10, pp. 118-124, Oct. 2002.

[16] B. Fortz and M. Thorup. "Increasing Internet Capacity Using Local Search", *Computational Optimization and Applications*, vol. 29, pp. 13-48, 2004.

[17] A. Sridharan, R. Guerin and C. Diot. "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks", *IEEE/ACM Trans. on Networking*, vol.13, No. 2, pp. 234-247, Apr. 2005.

[18] Srivastava, S., Agrawal, G., Pioro, M., and Medhi, D. 2005. Determining link weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach. *IEEE e-Trans. on Network and Service Management*, vol. 2, No.1, pp.9-18.

[19] Z. Wang, Y. Wang and L. Zhang. "Internet traffic engineering without full mesh overlaying", *Proc. IEEE INFOCOM, 2001*, Anchorage, AK.

[20] D. Xu, M. Chiang, and J. Rexford. "Link-state routing with hop-by-hop forwarding achieves optimal traffic engineering", *Proc. IEEE INFOCOM, 2008*, Phoenix, USA.

[21] K. Xu, H. Liu, J. Liu and J. Zhang. "LBMP: A Logarithm-Barrier-based Multipath Protocol for Internet Traffic Management", accepted for *IEEE Trans. on Parallel and Distributed Systems*. .

[22] H. Wang, H. Xie and L. Qiu. "COPE: traffic engineering in dynamic networks". *Proc. SIGCOMM'06*, Sep. 11-15, 2006, Pisa, Italy.

[23] Di Yuan. "A Bi-Criteria Optimization Approach for Robust OSPF Routing". *Proc. 6th IEEE Workshop on IP Operation Management (IPOM 2003)*, 2003.

[24] RFC 2328, http://tools.ietf.org/html/rfc2328.