# Achieving Optimal Traffic Engineering Using a Generalized Routing Framework

Ke Xu, *Senior Member, IEEE*, Meng Shen, *Member, IEEE*, Hongying Liu, Jiangchuan Liu, *Senior Member, IEEE*, Fan Li, *Member, IEEE*, and Tong Li

**Abstract—**The open shortest path first (OSPF) protocol has been widely applied to intra-domain routing in today's Internet. Since a router running OSPF distributes traffic *uniformly* over equal-cost multi-path (ECMP), the OSPF-based optimal traffic engineering (TE) problem (i.e., deriving optimal link weights for a given traffic demand) is computationally intractable for large-scale networks. Therefore, many studies resort to multi-protocol label switching (MPLS) based approaches to solve the optimal TE problem. In this paper we present a generalized routing framework to realize the optimal TE, which can be potentially implemented via OSPF- or MPLS-based approaches. We start with viewing the conventional optimal TE problem in a fresh way, i.e., optimally allocating the residual capacity to every link. Then we make a generalization of network utility maximization (NUM) to close this problem, where the network operator is associated with a utility function of the residual capacity to be maximized. We demonstrate that under this framework, the optimal routes resulting from the optimal TE are also the shortest paths in terms of a set of non-negative link weights that are explicitly determined by the optimal residual capacity and the objective function. The network entropy maximization theory is employed to enable routers to exponentially, instead of uniformly, split traffic over ECMP. The shortest-path penalizing exponential flow-splitting (SPEF) is designed as a link-state protocol with hop-by-hop forwarding to implement our theoretical findings. An alternative MPLS-based implementation is also discussed here. Numerical simulation results have demonstrated the effectiveness of the proposed framework as well as SPEF.

**Index Terms—**Traffic engineering, routing, OSPF, MPLS, utility, load balancing

## 1 Introduction

Internet traffic engineering (TE) addresses the performance optimization problem of operational networks [2]. The paramount objective of TE is to facilitate the transport of IP traffic through a given network in a possibly most efficient, reliable and expeditious manner. In this paper, we focus on TE in a single network domain (e.g., Autonomous System). Many TE solutions have been proposed in this field. From the perspective of TE enforcement mechanisms, they can be roughly classified as multiprotocol label switching (MPLS)-based and IP-based approaches.

The former approach relies on dedicated label switched paths (LSPs) for delivering encapsulated IP packets. Therefore, it enables explicit routing and arbitrary splitting of traffic, which is highly flexible for routing optimization. The major limitations are scalability and robustness for managing LSPs.

- *K. Xu and T. Li are with the Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science, Tsinghua University, Beijing, China. E-mail: xuke@tsinghua.edu.cn, litong12@mails.tsinghua.edu.cn.*
- *M. Shen and F. Li are with the School of Computer Science, Beijing Institute of Technology, Beijing, China. E-mail: {shenmeng, fli}@bit.edu.cn.*
- *H. Liu is with the School of Mathematics and Systems Science, Beihang University, Beijing, China. E-mail: liuhongying@buaa.edu.cn.*
- *J. Liu is with the School of Computing Science, Simon Fraser University, BC, Canada. E-mail: jcliu@cs.sfu.ca.*

*Manuscript received 4 Mar. 2014; revised 12 Dec. 2104; accepted 9 Jan. 2015. Date of publication 14 Jan. 2015; date of current version 16 Dec. 2015. Recommended for acceptance by S. Olariu.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TPDS.2015.2392760*

The basic idea of IP-based TE is to carefully manipulate link weights of interior gateway protocol (IGP). Open shortest path first (OSPF) [38], as a widely used IGP, has attracted many research attentions on achieving the optimal TE. In OSPF, equal-cost multipath (ECMP) directs an even splitting of traffic along multiple paths with equal OSPF weights. However, the even-splitting ECMP makes it computationally intractable to derive optimal link weights for large-scale networks [9]. In engineering practice, the state-of-the-art configurations remain largely intuitive and are lack of theoretical explanations, e.g., Cisco's InvCap [42] sets the weight of a link inversely proportional to its capacity.

It is highly desirable to design a *generalized* routing framework for the optimal intra-domain TE, with the following desired features. *First*, a mathematical optimization model should be involved as the theoretical foundation, which can produce certain solutions (e.g., routing and corresponding traffic distribution) with provable optimality. *Second*, this framework should be capable of capturing a variety of TE goals. As operators might be interested in different network performance indicators, several TE objective functions are proposed to meet individual TE goals, such as lowering the maximum link utilization (MLU) and minimizing the delay approximated by a piecewise-linear function of $M/M/1$ queue [11]. These objective functions are designed independent of one another, which brings difficulties in switching smoothly among various TE goals. *Last*, but not the least, this framework should also support feasible and flexible implementations of the solutions derived from the optimization model. Existing OSPF-based approaches leverage uneven splitting to realize the optimal routing, whereas they either sacrifice the optimality [28] or burden

routers in link weight computation [31]. Hence, it remains a challenge to realize the optimal TE following the destination-based hop-by-hop forwarding scheme.

In this paper, we develop a generalized routing framework by addressing the above challenges. We employ the network utility maximization (NUM) framework to model the optimal TE, which can be considered to be an *inversion* of NUM from an originally end-to-end scheme to a setting much better suited to intra-domain TE [17]. Originally, NUM is a flow-control framework, in which the network fixes routes and offers prices to end users, who in turn actively vary their traffic sending rates to maximize their own utilities. In the intra-domain TE, however, the situation is just the reverse: the amount of input traffic known as network-wide traffic demands is fixed and it is the operator, instead of end users, who has a utility function to maximize. NUM is non-particularly well-suited to this *rate adaptive multi-path routing* setting, and its adoption has been an open question so far [13]. We generalize NUM to the intra-domain TE context.

We further investigate a class of generic $(\boldsymbol{q}, \beta)$ load balancing utility functions that meet diverse interests of the ISPs by setting different parameter values. We prove that, with our framework, the optimal traffic distribution derived from any specific utility function, e.g., minimizing MLU or minimizing the function proposed in [11], can be emulated by setting an appropriate $\boldsymbol{q}$ in the $(\boldsymbol{q}, 1)$ load balancing utility function.

Finally, we show that the optimal routes that maximize a generalized objective function in the NUM framework are the shortest paths in terms of a set of non-negative link weights that are *explicitly* determined by the optimal traffic distribution and the objective function. By incorporating the network entropy maximization (NEM) theory [28] into the NUM framework, we design a link-state protocol named shortest-path penalizing exponential flow-splitting (SPEF) to achieve the optimal TE. Compared with original OSPF, SPEF maintains the destination-based hop-by-hop forwarding along the shortest paths, and needs only one more weight for each link, which makes it highly applicable in real networks. In addition, we also describe the MPLS-based implementation.

The remainder of this paper is outlined as follows. We first summarize the related work and highlight our novelty in Section 2. Then we put forward the NUM framework and theoretically prove the existence of the optimal link weights with general objective functions in Section 3, which is followed by the investigation of a class of generic utility functions for TE in Section 4. Implementation issues are described and discussed in Section 5. After performance evaluation in Section 6, we conclude the paper in Section 7. As this paper is extended from the conference version [1], the major differences are stated in Section 2.2.

## 2 Related Work

### 2.1 Brief Survey on Literatures

There have been many prior studies on the development of both single-domain and multi-domain optimal TE mechanisms for IP networks [11], [28], [30], [31], [32], [33], [34], [36]. Here we restrict ourselves to the most relevant ones to our work.

The optimal TE is usually formulated as minimizing a cost function under multi-commodity flow (MCF) constraints, where objective functions might be MLU [22], the piecewise-linear approximation of the $M/M/1$ delay formula [11], or a combinational function of user utility and congestion control [29]. Ben-Ameur et al. [3] and Gourdin and Klopfenstein [12] illustrate the superiority of the objective function of load balancing, which sheds a bright light on the development of operator-friendly objective functions with configurable load balancing criteria.

In OSPF-based TE, the optimal link weight computation with even splitting is known to be NP-hard [9], [34]. Therefore, many heuristics are proposed, including the local search approach [9] and algorithms with uneven splitting [23], [24], [26]. An important limitation of these uneven-splitting algorithms is that routers are unable to independently compute the traffic splitting fractions if only link weights are available. The authors in [37] put forward distributed adaption laws which enable each router to independently distribute traffic among any given set of next hops in an optimal way. PEFT proposed by Xu et al. [28] addresses the above limitation by incorporating the NEM framework. However, it fails to maintain the shortest paths in packet forwarding and thus sacrifices a key benefit of OSPF. Tso and Pezaros [32] implement PEFT in a cloud datacenter environment. Michael et al. [31] design HALO, an optimal link-state routing algorithm, where link weights can be calculated locally by routers. The distributed link weight calculation requires strict synchronous updates among routers.

In the field of MPLS-based approaches, many recent studies propose online TE algorithms to lower the LSPs management overhead. MIRA [18] minimizes the interference of a new LSP with existing routes that may be critical to satisfy future demands. COPE [25] combines the oblivious routing and prediction-based routing. REPLEX [8] employs the game theoretical rerouting policy given a set of fixed LSPs of each ingress-egress pair. Foteinos et al. [35] propose a TE framework, where the algorithm seeks for desired LSPs configurations according to customized high-level operational policies (e.g., load balancing and energy consumption).

The preceding literature overview reveals the lack of a routing framework that simultaneously supports diverse operational goals (i.e., objective functions), multiple potential implementations (i.e., OSPF- and MPLS-based), and the provable optimality. In this paper, we are dedicated to developing such a generalized framework for the optimal TE.

### 2.2 Elaboration of Novelty

In this paper, we generalize the framework in the preliminary work [1] from the following aspects: In order to capture various ISPs requirements in TE, we put forward a class of load balancing criteria, investigate the optimal link weights for two existing utility functions and the newly proposed load balancing utility functions, and quantify the efficiency of traffic distribution in terms of load balancing. We also depict the optimal link weights for OSPF-based

implementation, and describe an MPLS-based implementation. In addition, extensive packet-level NS2 simulations are conducted to evaluate the performance of the generalized framework.

Our OSPF-based implementation in this paper (i.e., SPEF) is closely related to PEFT [28]. Compared to PEFT, SPEF provably achieves the optimal TE without any optimality degradation. PEFT firstly employs NEM for exponential traffic splitting, however, theoretically, PEFT requires all possible paths of every ingress-egress pair to be involved as given information of NEM. To prevent loops and promote computational efficiency, the *downward* PEFT is proposed, which does not provably achieve the optimality. In SPEF, the first-set link weights are applied to get the shortest path(s) for every ingress-egress pair. Then NEM is employed to guide the traffic splitting over equal-cost shortest paths, which ensures loop-free routing. The second-set link weights also enable routers to *independently* calculate corresponding splitting functions.

## 3 Problem Definition and Formulation

### 3.1 Network Model

We consider a directed network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with a vertex set $\mathcal{N}$, edge set $\mathcal{E}$, and ingress-egress pair set $\mathcal{M}$. In the following, we use notations $N$, $E$ and $M$ to denote the cardinalities of set $\mathcal{N}$, $\mathcal{E}$ and $\mathcal{M}$, respectively. Each edge $(i,j) \in \mathcal{E}$ has a capacity $c_{ij}$. The traffic demand for an ingress-egress pair $(s_m, t_m) \in \mathcal{M}$ is denoted by $d_m$. The objective is to determine the routes for each ingress-egress pair so as to optimally make use of the network infrastructure.

It is known that routing in a network can be treated as multi-commodity flows [4], which is a network flow problem with multiple commodities flowing through the network with different source and sink nodes. A traffic distribution $\boldsymbol{f} = (f_{ij}, (i,j) \in \mathcal{E})$ is *feasible* if there exists $\boldsymbol{f}^{\mathcal{M}} = (\boldsymbol{f}^m, m \in \mathcal{M})$ satisfying the following constraints

$$f_{ij} = \sum_{m \in \mathcal{M}} f_{ij}^m \le c_{ij}, \ \forall (i,j) \in \mathcal{E} \tag{1a}$$

$$\sum_{i:(n,i)\in\mathcal{E}} f_{ni}^m - \sum_{j:(j,n)\in\mathcal{E}} f_{jn}^m = d_n^m, \ \forall m \in \mathcal{M}, \ \forall n \in \mathcal{N} \tag{1b}$$

$$f_{ij}^m \ge 0, \ \forall m \in \mathcal{M}, \ \forall (i,j) \in \mathcal{E}, \tag{1c}$$

where (1a) and (1b) are the capacity constraints and flow conservation constraints, respectively, and $d_n^m$ is the amount of traffic that node $n$ contributes to $(s_m, t_m)$, i.e.,

$$d_n^m = \begin{cases} d_m, & \text{if } n = s_m \\ -d_m, & \text{if } n = t_m \\ 0, & \text{otherwise.} \end{cases}$$

In a feasible traffic distribution $\boldsymbol{f}$, the total load and utilization of a link $(i,j)$ are $f_{ij}$ and $f_{ij}/c_{ij}$, respectively.

For the given traffic, TE deals with objective functions that potentially affect network congestion, such as the link-cost function $\Phi(\boldsymbol{f})$ in [11]. $\Phi(\boldsymbol{f})$ is a non-decreasing and convex function of $\boldsymbol{f}$, and enables quantitative comparisons between different routing solutions in terms of link load $f_{ij}$. The optimal TE [11] is to minimize the link-cost function $\Phi(\boldsymbol{f})$ under the multi-commodity flow constraints (1).

### 3.2 Universal Existence of Optimal Routes

In this paper, we view the optimal TE in a fresh way, which is optimally allocating the *residual capacity* to each link instead of distributing traffic load on each link. There are two reasons: 1) the average delay on link $(i,j)$ depends largely on its residual capacity $r_{ij} = c_{ij} - f_{ij}$ from the well-known Kleinrock independence approximation [4], and 2) when link failures occur, it is more convenient to reroute traffic demands along alternative links that have non-zero residual capacities [18].

We associate link $(i,j)$ with an operator. Assume that if a residual capacity $r_{ij}$ is maintained at link $(i,j)$, the operator will have utility $V(\boldsymbol{r})$, where $\boldsymbol{r} = (r_{ij}, (i,j) \in \mathcal{E})$ is the residual capacity vector, abbreviated to *residual capacity*. We assume that the utility $V(\boldsymbol{r})$ is a non-decreasing and concave function of $\boldsymbol{r}$ over the range $\boldsymbol{r} \ge \boldsymbol{0}$. These simple assumptions hold in several common cases (e.g., MLU and piecewise-linear approximation of the $M/M/1$ delay formula) and will be further discussed in Section 3.3.

Now the optimal TE can be formulated to maximize the utility under the MCF constraints (1), i.e.

$$\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V) \quad \underset{\boldsymbol{r}\ge\boldsymbol{0}, \boldsymbol{f}^m \ge \boldsymbol{0}}{\text{maximize}} \ V(\boldsymbol{r}) \tag{2a}$$

$$\text{subject to } \boldsymbol{r} + \sum_{m \in \mathcal{M}} \boldsymbol{f}^m = \boldsymbol{c}, \tag{2b}$$

$$\boldsymbol{A}\boldsymbol{f}^m = \boldsymbol{d}^m, \ \forall m \in \mathcal{M}, \tag{2c}$$

where $\boldsymbol{A}$, an $N \times E$ node-arc incidence matrix for network $\mathcal{G}$, is introduced to represent the multi-commodity flow constraints (1b). The column corresponding to link $(i,j)$ is $+1$ in row $i$, $-1$ in row $j$ or 0 otherwise.

From the general theory of convex optimization (e.g., [5], p. 279, Corollary 28.2.2), it follows that if $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$, then there exists a nonnegative[1] Lagrangian multiplier vector $\boldsymbol{w} = (w_{ij}, (i,j) \in \mathcal{E})$ such that $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves

$$\begin{aligned} &\underset{\boldsymbol{r}\ge\boldsymbol{0}, \boldsymbol{f}^m \ge \boldsymbol{0}}{\text{maximize}} \ V(\boldsymbol{r}) - \sum_{(i,j)\in\mathcal{E}} w_{ij} r_{ij} - \sum_{m \in \mathcal{M}} \boldsymbol{w}^\top \boldsymbol{f}^m + \boldsymbol{c}^\top \boldsymbol{w} \\ &\text{subject to } \boldsymbol{A}\boldsymbol{f}^m = \boldsymbol{d}^m, \ \forall m \in \mathcal{M}. \end{aligned} \tag{3}$$

It is a separable optimization problem since there is no coupling among variables, $\boldsymbol{r}$ and $\boldsymbol{f}^m$ for all $m \in \mathcal{M}$, in the objective function and constraints. Then $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$ can be decomposed into two subsidiary optimization problems, one for the ISP and the other for the network, by using price per unit residual capacity as a Lagrangian multiplier that mediates between two subproblems.

1. In Eq. (2), we have that (2b) is an equality condition and hence the corresponding Lagrangian multiplier $\boldsymbol{w}$ should be unrestricted in sign. Here the nonnegativity of $\boldsymbol{w}$ can be deviated from the concavity and non-decreasing of $V(\boldsymbol{r})$, see Theorem 4.

**Theorem 1.** *If $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves* $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$*, then there exists a Lagrangian multiplier vector $\boldsymbol{w} = (w_{ij}, (i,j) \in \mathcal{E})$ such that $\hat{\boldsymbol{r}}$ solves the* residual capacity planning problem

$$\mathrm{ISP}(V, \boldsymbol{w}) \underset{\boldsymbol{r} \geq \boldsymbol{0}}{\text{maximize}} \quad V(\boldsymbol{r}) - \sum_{(i,j)\in\mathcal{E}} w_{ij} r_{ij} \tag{4}$$

*and for each $m \in \mathcal{M}$, $\hat{\boldsymbol{f}}^m$ solves the* routing problem

$$\begin{aligned} \mathrm{SP}_m(\boldsymbol{w}) \qquad & \underset{\boldsymbol{f}^m \geq \boldsymbol{0}}{\text{minimize}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f_{ij}^m \\ & \text{subject to} \quad \boldsymbol{A}\boldsymbol{f}^m = \boldsymbol{d}^m. \end{aligned} \tag{5}$$

Here we give some engineering interpretations to $\mathrm{ISP}(V, \boldsymbol{w})$ and $\mathrm{SP}_m(\boldsymbol{w})$ for all $m \in \mathcal{M}$. First, $\mathrm{ISP}(V, \boldsymbol{w})$ can be interpreted as a residual capacity planning problem in which the ISP determines the possible residual capacity with the given link cost $w_{ij}$. Meanwhile, the network utility generated for residual capacity $\boldsymbol{r}$ is maximized. Then each ingress-egress pair finds a solution to the total cost minimization under the given $w_{ij}$.

A good property of the optimal routes is that they are the shortest paths. That is to say, the route for each ingress-egress pair is the shortest path under the link weight $w_{ij}$. Let $\boldsymbol{\pi}^t$ denote the optimal solution to the dual of $\mathrm{SP}_m(\boldsymbol{w})$. Based on the well-known *complementarity condition* in optimal conditions (See [5], p. 281, Theorem 28.3), we have

$$\pi_i^m - \pi_j^m = w_{ij}, \quad \text{if} \quad \mathrm{f}_{\mathrm{ij}}^{\mathrm{m}} > 0 \tag{6a}$$

$$\leq w_{ij}, \quad \text{if} \quad \mathrm{f}_{\mathrm{ij}}^{\mathrm{m}} = 0. \tag{6b}$$

Let $p : i_0 i_1 \cdots i_n$ be a possible path of the ingress-egress pair $(s_m, t_m)$, where $i_0 = s_m$ and $i_n = t_m$. For example, if $y_p = \min_{k=1,2,\ldots,n} \hat{f}_{i_{k-1} i_k}^m > 0$, we have $\sum_{(i,j)\in p} w_{ij} = \pi_{s_m}^m - \pi_{t_m}^m \leq \sum_{(i,j)\in\bar{p}} w_{ij}$ for any other path $\bar{p}$ that has the same ingress-egress nodes. Here the equality follows Eq. (6a) and the inequality follows Eq. (6b).

The Lagrangian multiplier $w_{ij}$ is the *shadow price* of the additional capacity at link $(i,j)$ [5], which can be viewed as the *generalized cost* of traffic through link $(i,j)$. Then the universal existence of the optimal link weight will lead to the optimal routes, which can be guaranteed by Theorem 1. Hereafter we refer to $\boldsymbol{w}$ as the *first*-set link weights, under which the optimal routes are the shortest paths.

**Remark 1.** Wang et al. [22] have shown that for any given set of routes, it is either shortest-path-reproducible or loopy. This is demonstrated through a conversion to a set of the shortest paths with respect to some link weights. Our results from convex optimization, however, *directly* show the universal existence of the optimal link weights. More importantly, we thoroughly reveal how the link weights *explicitly* depend on the utility function and the residual capacity in Theorems 2 and 3.

Theorem 1 implies that the Lagrangian multiplier vector $\boldsymbol{w}$ for $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$ provides link weights such that all the traffic is forwarded along the shortest paths. Meanwhile, the ISP achieves the maximum utility through retaining the residual capacity. Inversely, if there exists link weight $\boldsymbol{w}$ such that the solution to $\mathrm{ISP}(V, \boldsymbol{w})$ is consistent with the solution to $\mathrm{SP}_m(\boldsymbol{w})$ for all $m \in \mathcal{M}$, then Theorem 2 implies that those vectors also solve $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$.

**Theorem 2.** *If a weight vector $\boldsymbol{w} = (w_{ij}, (i,j) \in \mathcal{E})$ exists such that the solution $\hat{\boldsymbol{r}} = (r_{ij}, (i,j) \in \mathcal{E})$ to* $\mathrm{ISP}(V, \boldsymbol{w})$ *and the solution $\hat{\boldsymbol{f}}^m$ to* $\mathrm{SP}_m(\boldsymbol{w})$ *$(m \in \mathcal{M})$ are consistent, i.e.*

$$\hat{\boldsymbol{r}} + \sum_{m\in\mathcal{M}} \hat{\boldsymbol{f}}^m = \boldsymbol{c}, \tag{7}$$

*then $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves* $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$*. (See proof in the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2015.2392760.)*

### 3.3 Depicting the Optimal Link Weights

Let $f$ be a concave function within domain $C$. Let $\boldsymbol{p}$ be a supergradient of $f$ at $\boldsymbol{x} \in C$ if

$$f(\boldsymbol{x}) + \boldsymbol{p}^\top(\boldsymbol{y} - \boldsymbol{x}) \geq f(\boldsymbol{y}), \ \forall \boldsymbol{y} \in C.$$

Superdifferential of $f$ at $\boldsymbol{x}$ is the set of all the supergradients of $f$ at $\boldsymbol{x}$ and is denoted by $\partial f(\boldsymbol{x})$.[2] If $f$ is concave and differentiable at $\boldsymbol{x}$, then $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$.

With the optimality condition of $\mathrm{ISP}(V, \boldsymbol{w})$ (See [5], p. 281, Theorem 28.3) and the calculation rule of the superdifferential (See [15], p.183, Theorem 4.1.1), we get the property of the link weight as follows.

**Theorem 3.** *Let $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solve* $\mathrm{TE}(\mathcal{G}, \boldsymbol{D}, V)$ *and the vector $\boldsymbol{w}$ represent optimal link weight. Then there exists $\boldsymbol{p} \in \partial V(\hat{\boldsymbol{r}})$ such that for all $(i,j) \in \mathcal{E}$*

$$w_{ij} = p_{ij}, \quad \text{if } \hat{r}_{ij} > 0 \tag{8a}$$

$$w_{ij} \geq p_{ij}, \quad \text{if } \hat{r}_{ij} = 0. \tag{8b}$$

In the sequel, we refer to link $(i,j) \in \mathcal{E}$ as a *saturated* link if $r_{ij} = 0$; otherwise, an *unsaturated* link. Theorem 3 shows that the link weight is the component of the supergradient of the utility function for the optimal residual capacity at an unsaturated link.

**Lemma 1.** *If $\boldsymbol{0} \leq \hat{\boldsymbol{r}} < \boldsymbol{c}$, then it holds that $\boldsymbol{p} \geq \boldsymbol{0}$ for any $\boldsymbol{p} \in \partial V(\hat{\boldsymbol{r}})$.*

**Proof.** Given $\boldsymbol{p} \in \partial V(\hat{\boldsymbol{r}})$ and $(i,j) \in \mathcal{E}$, let $r_{ij} = \hat{r}_{ij} + \epsilon$ and $r_{kl} = \hat{r}_{kl}$ for all $(k,l) \in \mathcal{E}$ but $(k,l) \neq (i,j)$, where $\epsilon$ is positive and makes $\boldsymbol{r} \leq \boldsymbol{c}$. Then $V(\boldsymbol{r}) - V(\hat{\boldsymbol{r}})$ is nonnegative for the utility function $V(\boldsymbol{r})$ is non-decreasing. Furthermore, by the definition of supergradient, it holds that

$$V(\hat{\boldsymbol{r}}) + \epsilon p_{ij} \geq V(\boldsymbol{r}).$$

So $p_{ij} \geq 0$. We get $\boldsymbol{p} \geq \boldsymbol{0}$. □

Link costs possibly taking negative values then would cause serious problems in shortest path calculations and IGPs as well. By Theorem 3, the link weight vector satisfies

2. For convex $f$, $\partial f(\boldsymbol{x})$ denotes the set of subgradients and is called the subdifferential.

$\boldsymbol{w} \geq \boldsymbol{p}$ for some $\boldsymbol{p} \in \partial V(\hat{\boldsymbol{r}})$. Combining Lemma 1, we can solve the problem.

**Theorem 4.** *The link weight vector $\boldsymbol{w}$ in Theorem 1 is nonnegative.*

**Lemma 2 ([15], p.187, Corollary 4.3.2).** *Let $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ be $m$ concave and differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$ and define*

$$f(\boldsymbol{x}) := \min\{f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\}.$$

*Denoting by $\mathcal{I}(\boldsymbol{x}) := \{i : f_i(\boldsymbol{x}) = f(\boldsymbol{x}), i = 1, \ldots, m\}$ the active index-set, we have*

$$\partial f(\boldsymbol{x}) = \left\{ \sum_{i \in \mathcal{I}(\boldsymbol{x})} \theta_i \nabla f_i(\boldsymbol{x}) : \theta_i \geq 0 \text{ for } i \in \mathcal{I}(\boldsymbol{x}), \sum_{i \in \mathcal{I}(\boldsymbol{x})} \theta_i = 1 \right\}.$$

Based on Theorem 3, we leverage Lemma 2 to describe nice observations which illustrate the optimal link weights achieved for two common cost functions.

To minimize MLU, we have

$$V(\boldsymbol{r}) = \min_{(i,j) \in \mathcal{E}} \frac{r_{ij} - c_{ij}}{c_{ij}}.$$

Let $\mathcal{I}(\hat{\boldsymbol{r}})$ be the set of links with MLU ( i.e., bottleneck links)

$$\mathcal{I}(\hat{\boldsymbol{r}}) = \left\{ (i,j) \in \mathcal{E} : \frac{\hat{r}_{ij} - c_{ij}}{c_{ij}} = \min_{(i,j) \in \mathcal{E}} \frac{r_{ij} - c_{ij}}{c_{ij}} \right\}.$$

With Lemma 2, we have $\boldsymbol{p} \in \partial V(\hat{\boldsymbol{r}})$ if and only if $p_{ij} = a_{ij}/c_{ij}$ for $(i,j) \in \mathcal{I}(\hat{\boldsymbol{r}})$ and $a_{ij} \geq 0, \sum_{(i,j) \in \mathcal{I}(\hat{\boldsymbol{r}})} a_{ij} = 1$; otherwise, $p_{ij} = 0$. The results show that the routing that minimizes MLU is the shortest path routing for each ingress-egress pair, where the weights of the non-bottleneck links are all zero, whereas *only* the weights of bottleneck links are positive and inversely proportional to their own capacities.

The piecewise-linear approximation of the $M/M/1$ delay formula proposed by Fortz and Thorup [11] is based on discussions with the technicians in AT&T Lab. They assume utilities are additive, so that the aggregate utility of residual capacity $\boldsymbol{r}$ is $\sum_{(i,j) \in \mathcal{E}} V_{ij}(r_{ij})$ [21], where

$$V_{ij}(r_{ij}) = \begin{cases} r_{ij} - c_{ij}, & \frac{r_{ij}}{c_{ij}} \geq \frac{2}{3} \\ 3r_{ij} - \frac{7}{3} c_{ij}, & \frac{1}{3} \leq \frac{r_{ij}}{c_{ij}} < \frac{2}{3} \\ 10r_{ij} - \frac{14}{3} c_{ij}, & \frac{1}{10} \leq \frac{r_{ij}}{c_{ij}} < \frac{1}{3} \\ 70r_{ij} - \frac{32}{3} c_{ij}, & 0 \leq \frac{r_{ij}}{c_{ij}} < \frac{1}{10}. \end{cases} \tag{9}$$

Here we ignore the case where $r_{ij} < 0$ and reformulate it as

$$V_{ij}(r_{ij}) = \min_{k=1,\ldots,4} \varphi_k(r_{ij}),$$

where $\varphi_1(r_{ij}) = r_{ij} - c_{ij}, \varphi_2(r_{ij}) = 3r_{ij} - \frac{7}{3} c_{ij}, \varphi_3(r_{ij}) = 10r_{ij} - \frac{14}{3} c_{ij}$, $\varphi_4(r_{ij}) = 70r_{ij} - \frac{32}{3} c_{ij}$. With Lemma 2, we have $p_{ij} \in \partial V_{ij}(\hat{r}_{ij})$ if and only if

$$p_{ij} = \begin{cases} 1, & \frac{\hat{r}_{ij}}{c_{ij}} > \frac{2}{3} \\ 1 + 2a, & \frac{\hat{r}_{ij}}{c_{ij}} = \frac{2}{3} \\ 3, & \frac{1}{3} < \frac{\hat{r}_{ij}}{c_{ij}} < \frac{2}{3} \\ 3 + 7a & \frac{\hat{r}_{ij}}{c_{ij}} = \frac{1}{3} \\ 10, & \frac{1}{10} < \frac{\hat{r}_{ij}}{c_{ij}} < \frac{1}{3} \\ 10 + 60a, & \frac{\hat{r}_{ij}}{c_{ij}} = \frac{1}{10} \\ 70, & 0 < \frac{\hat{r}_{ij}}{c_{ij}} < \frac{1}{10}, \end{cases}$$

where $a \in [0,1]$ is a constant. The results show that the routing that minimizes piecewise-linear approximation of the $M/M/1$ delay formula is the shortest path routing for each ingress-egress pair, where the link weights are dependent on the link load. For example, the link weight is 1 for the link with utilization no greater than $1/3$, and is 70 for the link with utilization greater than $9/10$, and the link weight lies in the interval $[10, 70]$ for the link with utilization $9/10$.

## 4 A Class of Generic Utility Functions

In order to capture various TE requirements of network operators, we first exploit a class of $(\boldsymbol{q}, \beta)$ load balancing utility functions, and then investigate the relationship between *any* utility function and a $(\boldsymbol{q}, 1)$ load balancing utility function. To quantify the efficiency of each traffic distribution in terms of load balancing, we propose a novel definition of proportional load balancing.

### 4.1 $(\boldsymbol{q}, \beta)$ Load Balancing Utility Function

We assume that the operator has utility $V_{ij}(r_{ij})$ (e.g., (9)), if a residual capacity $r_{ij}$ is maintained at link $(i,j)$. Assume further utilities are additive, so that the aggregate utility of residual capacity $\boldsymbol{r}$ is $\sum_{(i,j) \in \mathcal{E}} V_{ij}(r_{ij})$.

To simplify the analysis, we assume the utility function is strictly concave and differentiable. In addition, $V_{ij}(r_{ij})$ tends to $-\infty$ as $r_{ij} \to 0$. Under those assumptions, for a residual capacity $\boldsymbol{r}$ (or the traffic distribution $\boldsymbol{f} = \boldsymbol{c} - \boldsymbol{r}$), there may be multiple flow vectors $\boldsymbol{f}^{\mathcal{M}}$ satisfying Eqs. (2b) and (2c). We say that $\hat{\boldsymbol{r}}$ solves TE$(V, \mathcal{G}, \boldsymbol{D})$ if there exists $\hat{\boldsymbol{f}}^{\mathcal{M}}$ such that $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves TE$(V, \mathcal{G}, \boldsymbol{D})$. We denote the corresponding link load with $\hat{\boldsymbol{f}} = \boldsymbol{c} - \hat{\boldsymbol{r}}$.

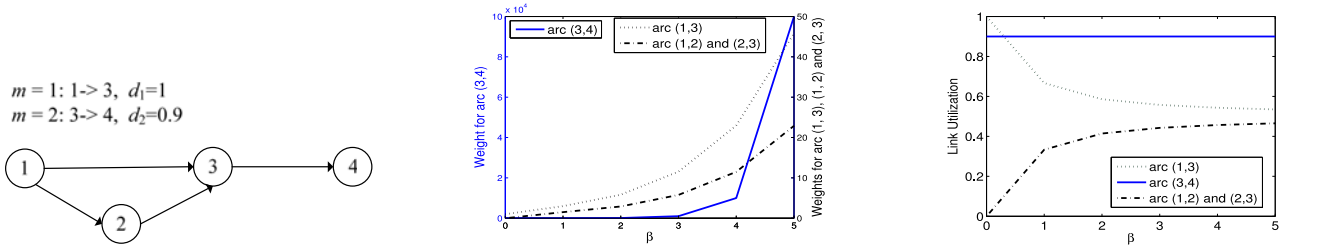
Let $q_{ij}$ be positive constants for all $(i,j) \in \mathcal{E}$ and $\beta$ be a positive constant. We consider a class of TE utility functions and refer to it as $(\boldsymbol{q}, \beta)$ load balancing utility function

$$V_{ij}(r_{ij}) = \begin{cases} q_{ij} \log r_{ij}, & \text{if } \beta = 1 \\ q_{ij}(1-\beta)^{-1}(r_{ij}/c_{ij})^{1-\beta}, & \text{if } \beta \neq 1. \end{cases} \tag{10}$$

Since $(\boldsymbol{q}, \beta)$ load balancing utility function is differentiable, we have $\partial V(\boldsymbol{r}) = \{\nabla V(\boldsymbol{r})\}$ and $\frac{\partial}{\partial r_{ij}} V(\boldsymbol{r}) = q_{ij} c_{ij}^{\beta-1} r_{ij}^{-\beta}$. Now, with Theorems 1 and 3, we give the physical meaning of the link weights for some specific $(\boldsymbol{q}, \beta)$ load balancing utility functions as follows.

**Example 1.** $(\mathbf{1}, 1)$ *load balancing utility function.*[3] Let the residual capacity $\hat{\boldsymbol{r}}$ solve TE$(\mathcal{G}, \boldsymbol{D}, V)$ with $V_{ij}(r_{ij}) = \log r_{ij}$. With (8a), we get the optimal link weight

3. In this paper, we use $\mathbf{1}$ to denote an all-one vector.




(a) The network topology (b) Link weight vs load balancing parameter $\beta$ (c) Link utilization vs load balancing parameter $\beta$

Fig. 1. An example illustrating the implication of $\beta$ in $(\mathbf{1}, \beta)$ load balancing criteria and the reason why minimizing MLU is not well-defined.

$$w_{ij} = \frac{1}{\hat{r}_{ij}} = \frac{1}{c_{ij} - \hat{f}_{ij}},$$

i.e. the average packet delay on link $(i, j)$ based on the $M/M/1$ queuing model [4]. Then if the path $\hat{p}$ for $(s, t)$ bears positive traffic, we have

$$\sum_{(i,j)\in\hat{p}} \frac{1}{c_{ij} - \hat{f}_{ij}} \leq \sum_{(i,j)\in p} \frac{1}{c_{ij} - f_{ij}}$$

for any other path $p$ for $(s, t)$. We show that the solution to TE$(\mathcal{G}, \boldsymbol{D}, V)$ with $(\mathbf{1}, 1)$ load balancing utility function minimizes the average packet queuing delay of $(s_m, t_m)$ for all $m \in \mathcal{M}$. If a network is running with low utilization, then $\hat{f}_{ij} \ll c_{ij}$, and therefore, the delay becomes $1/(c_{ij} - \hat{f}_{ij}) \approx 1/c_{ij}$. It is consistent with the Cisco's InvCap [42].

**Example 2.** $(\mathbf{1}, 2)$ *load balancing utility function.* Let the residual capacity $\hat{\boldsymbol{r}}$ solve TE$(\mathcal{G}, \boldsymbol{D}, V)$ with $V_{ij}(r_{ij}) = (-c_{ij})/r_{ij} = -1 - f_{ij}/(c_{ij} - f_{ij})$. In this case, the problem (2) tries to minimize the total average queuing delay with the $M/M/1$ queuing model and the optimal link weights $w_{ij} = c_{ij}/\hat{r}_{ij}^2$.

**Example 3.** $(\boldsymbol{q}, 0)$ *load balancing utility function.* Let $d_{ij}$ be the processing and propagation delay for unit traffic on link $(i, j)$. Set $q_{ij} = d_{ij}c_{ij}$. Let the residual capacity $\hat{\boldsymbol{r}}$ solve TE$(\mathcal{G}, \boldsymbol{D}, V)$ with $V_{ij}(r_{ij}) = d_{ij}r_{ij} = d_{ij}c_{ij} - d_{ij}f_{ij}$. In this case, TE$(\mathcal{G}, \boldsymbol{D}, V)$ tries to minimize the total processing and propagation delay, and the optimal link weight $w_{ij} = d_{ij}$ for the unsaturated link $(i, j)$ and $w_{ij} \geq d_{ij}$ for the saturated link $(i, j)$. If $d_{ij} = 1$, we have the minimum hop routing.

Consider the network in Fig. 1a. There are four links with capacities all being 1 unit. The non-zero demands are 1 and 0.9 units for ingress-egress pair $(1, 3)$ and $(3, 4)$, respectively. There are two paths for ingress-egress pair $(1, 3)$, i.e. 1-3 and 1-2-3, and a single path for ingress-egress pair $(3, 4)$, i.e. 3-4. Assume that $q_{ij}$ is 1 for each link $(i, j)$, the impact of load balancing parameter $\beta$ on link weight and link utilization is shown in Figs. 1b and 1c, respectively.

The difference between weights of links (1, 3) and (1, 2) increases as $\beta$ increases, as shown in Fig. 1b. The difference between utilizations of these links, however, decreases with the increase of $\beta$, as shown in Fig. 1c. These facts imply that we can balance the loads on different links by tuning the parameter $\beta$.

In order to have a deep insight into the links weights and traffic distribution when $\beta$ is 0 and 1, we list corresponding results in Table 1. In the case when $\beta = 0$ (i.e. the 2nd column), the optimization goal is equivalent to maximizing the network-wide residual bandwidth. Therefore, ingress-egress pair $(1, 3)$ only employs a single shortest path, i.e. 1-3, which results in the overall residual bandwidth of 2.1 units. This case also confirms the conclusion of *Example* 3 that, we will have the minimum hop routing if $\beta = 0$ and $d_{ij} = 1$.[4]

In another case when $\beta = 1$ (i.e. the 3rd column), there are two equal-cost shortest paths for ingress-egress pair $(1, 3)$, namely 1-3 and 1-2-3 with 2/3 and 1/3 of the total traffic load, respectively. It is easy to validate that the resulting optimal link weights follows the link weight assignment equation in *Example* 1. In Column 4, we also list the results with the objective function that minimizes piecewise-linear approximation of the M/M/1 delay formula [11]. We can find that the resulting optimal routing is the same as the case for $\beta = 1$, i.e. the same traffic slitting fractions for ingress-egress pair $(1, 3)$ over two equal-cost shortest paths, and the same link utilization.

**Remark 2.** Through numerical studies, Ben-Ameur et al. [3] and Gourdin and Klopfenstein [12] examine the impact of different objective functions, particularly those in (10) with $\beta = \infty$ and $\beta = 2$. Our work complements theirs by theoretically examining the optimal link weights associated with the $(\boldsymbol{q}, \beta)$ load balancing utility function.

### 4.2 General Utility versus $(q, 1)$ Load Balancing Utility

While the optimization problem TE$(\mathcal{G}, \boldsymbol{D}, V)$ is mathematically solvable, it involves the utility function $V$ that is unlikely to be known by the network. In this section, we highlight several observations with the $(\boldsymbol{q}, 1)$ load balancing utility, which helps operators to choose an appropriate utility function.

Assume the utility function is *strictly* concave and *differentiable*, we consider two simple problems. We regard the optimal link weight $w_{ij}$ as a charge per unit residual capacity for link $(i, j)$. If link $(i, j)$ can choose an amount to pay per unit time, $q_{ij}$, and receive in return a residual capacity $r_{ij}$ proportional to $q_{ij}$, say $r_{ij} = q_{ij}/w_{ij}$, the utility maximization problem for link $(i, j)$ becomes

4. As we assume that $q_{ij}$ and $c_{ij}$ are 1 in Fig. 1a, $d_{ij} = q_{ij}/c_{ij} = 1$.

TABLE 1
The Link Weight and Link Utilization for Example 1 in Optimal TE with Different Objective Functions

| Link | $\beta = 0$ | | $\beta = 1$ | | B. Fortz & M. Thorup [11] | | min-max | | MLU | [22] |
|---|---|---|---|---|---|---|---|---|---|---|
| | weights | utilizations | weights | utilizations | weights | utilizations | weights | utilizations | weights | utilizations |
| (1, 3) | 2 | 1.00 | 3 | 0.67 | 4.6 | 0.67 | 2 | 0.50 | 0 | a‡ |
| (3, 4) | 1 | 0.90 | 10 | 0.90 | 40.0 | 0.90 | 1 | 0.90 | 1 | 0.90 |
| (1, 2) | 1 | 0.00 | 1.5 | 0.33 | 2.3 | 0.33 | 1 | 0.50 | 0 | 1 − a |
| (2, 3) | 0 | 0.00 | 1.5 | 0.33 | 2.3 | 0.33 | 1 | 0.50 | 0 | 1 − a |

‡*a is a constant in interval* $[0.1, 0.9]$

$$\text{Link}_{ij}(V_{ij}, w_{ij}) \quad \underset{q_{ij}\geq 0}{\text{maximize}} \quad V_{ij}\left(\frac{q_{ij}}{w_{ij}}\right) - q_{ij}. \tag{11}$$

Next, suppose that the network knows the vector $\boldsymbol{q} = (q_{ij}, (i,j) \in \mathcal{E})$, and attempts to maximize the function $\sum_{(i,j)\in\mathcal{E}} q_{ij} \log r_{ij}$. The network's optimization problem is then as follows, i.e.

$$\begin{aligned} \text{Network}(\mathcal{G}, \boldsymbol{D}, \boldsymbol{q}) \quad & \underset{\boldsymbol{r}\geq \boldsymbol{0}, \boldsymbol{f}^m \geq \boldsymbol{0}}{\text{maximize}} \sum_{(i,j)\in\mathcal{E}} q_{ij} \log r_{ij} \\ & \text{subject to } \boldsymbol{r} + \sum_{m\in\mathcal{M}} \boldsymbol{f}^m = \boldsymbol{c} \\ & \boldsymbol{A}\boldsymbol{f}^m = \boldsymbol{d}^m, \ \forall m \in \mathcal{M}. \end{aligned} \tag{12}$$

We say $\hat{\boldsymbol{r}}$ solves Network$(\mathcal{G}, \boldsymbol{D}, \boldsymbol{q})$ if there exists $\hat{\boldsymbol{f}}^{\mathcal{M}}$ such that $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{f}}^{\mathcal{M}})$ solves the optimization problem (12).

**Theorem 5.** *There always exist vectors* $\boldsymbol{w} = (w_{ij}, (i,j) \in \mathcal{E})$, $\hat{\boldsymbol{q}} = (q_{ij}, (i,j) \in \mathcal{E})$, *and* $\hat{\boldsymbol{r}} = (r_{ij}, (i,j) \in \mathcal{E})$, *satisfying* $w_{ij} > 0$ *and* $\hat{q}_{ij} = w_{ij}\hat{r}_{ij}$ *for all* $(i,j) \in \mathcal{E}$, *such that* $\hat{q}_{ij}$ *solves* $Link_{ij}(V_{ij}, w_{ij})$ *for* $(i,j) \in \mathcal{E}$ *and* $\hat{\boldsymbol{r}}$ *solves* $Network(\mathcal{G}, \boldsymbol{D}, \hat{\boldsymbol{q}})$. *Further, given any such triple* $(\boldsymbol{w}, \hat{\boldsymbol{q}}, \hat{\boldsymbol{r}})$, $\hat{\boldsymbol{q}}$ *and* $\hat{\boldsymbol{r}}$ *are uniquely determined and* $\hat{\boldsymbol{r}}$ *is the unique solution to* $\text{TE}(\mathcal{G}, \boldsymbol{D}, V)$.

Theorem 5, proven in the supplementary file, available online, shows that the optimal traffic distribution derived from (2) with a general objective function is the same as the one from (12), in which the objective function is $(\boldsymbol{q}, 1)$ load balancing.

With $\beta = 1$, if $q_{ij}, (i,j) \in \mathcal{E}$ are all 1, the resulting capacity $\boldsymbol{r}$ is proportional load balancing, and if $q_{ij}, (i,j) \in \mathcal{E}$ are all integers, the resulting capacity $\boldsymbol{r}$ is weighted proportional load balancing (See Section 4.3). These typical observations can be used as baselines for network operators to facilitate utility function selection according to their load balancing goals.

### 4.3 Load Balancing versus Efficiency

A major concern in TE is load balancing of traffic repartition against traffic fluctuation, which can be partially addressed by the routing pattern that minimizes MLU.

Another consideration of an operator is routing efficiency, which means that the routing retains as much residual capacities as possible for all links after satisfying the current traffic demands. A routing pattern with high efficiency, however, does not necessarily achieve load balancing. For example, pursuing high efficiency might result in an unbalanced scenario, where the majority of links have a large amount of residual capacity while the rest little. The ideal situation, which enjoys high efficiency and load balancing simultaneously, is hard to realize due to the contradiction of these two indicators. Therefore, in certain cases, better load balancing is preferred even at the cost of reducing efficiency. In this paper, we apply load balancing criteria to characterize how traffic is distributed over links so as to achieve various trade-offs.

A residual capacity $\hat{\boldsymbol{r}}$ is *proportional load balancing* if it is feasible, and for any feasible residual capacity $\boldsymbol{r}$, the aggregation of proportional changes of residual capacity is zero or negative,

$$\sum_{(i,j)\in\mathcal{E}} \frac{r_{ij} - \hat{r}_{ij}}{\hat{r}_{ij}} \leq 0. \tag{13}$$

A residual capacity $\hat{\boldsymbol{r}}$ is *weighted proportional load balancing* if it is feasible, and for any other feasible residual allocations $\boldsymbol{r}$ and some positive constants $q_{ij}$,

$$\sum_{(i,j)\in\mathcal{E}} q_{ij} \frac{r_{ij} - \hat{r}_{ij}}{\hat{r}_{ij}} \leq 0. \tag{14}$$

The relationship between conditions (13) and (14) is well illustrated when $q_{ij}, (i,j) \in \mathcal{E}$, are all integers. For each $(i,j) \in \mathcal{E}$, replace the single link $(i,j)$ with $q_{ij}$ identical sub-links, construct the proportional load balancing residual capacity allocation over the obtained $\sum_{(i,j)\in\mathcal{E}} q_{ij}$ links, and allocate link $(i,j)$ with the aggregate residual capacity through its $q_{ij}$ sub-links; then the final traffic distribution is weighted proportional load balancing.

Here we can explain Theorem 5 with the proportional load balancing. Assume that we associate an agent with a link. Theorem 5 also shows that, if the agent of a link is able to choose a charge per unit time that is willing to pay, and if the network allocates residual capacities so that the residual capacity per unit charge is proportional load balancing, then a system optimum is achieved when the link agents' choices on charges and the network's choice on the amount and price (per unit) of allocated residual capacities are in equilibrium.

The following definition is a generalized proportional load balancing. A residual capacity $\hat{\boldsymbol{r}}$ is $(q, \beta)$ *proportional load balancing* if it is feasible and for any other feasible $\boldsymbol{r}$,

$$\sum_{(i,j)\in\mathcal{E}} q_{ij} c_{ij}^{\beta-1} \ \frac{r_{ij} - \hat{r}_{ij}}{(\hat{r}_{ij})^{\beta}} \leq 0, \tag{15}$$

where $\beta$ is a non-negative *load balancing parameter*. It is reduced to the proportional load balancing with $\beta = 1$.

**Theorem 6.** *A residual capacity $\hat{\boldsymbol{r}}$ is $(\boldsymbol{q}, \beta)$ proportional load balancing if and only if $\hat{\boldsymbol{r}}$ solves* TE$(\mathcal{G}, \boldsymbol{D}, V)$ *with $(\boldsymbol{q}, \beta)$ utility function.*

**Proof.** Plug $g(\boldsymbol{r}) = \sum_{(i,j)\in\mathcal{E}} V_{ij}(r_{ij})$ into (2a). For $g(\boldsymbol{r})$ is concave and continuously differentiable, it holds that $\hat{\boldsymbol{r}}$ solves (2) if and only if

$$(\nabla g(\hat{\boldsymbol{r}}))^{\top}(\boldsymbol{r} - \hat{\boldsymbol{r}}) \leq 0$$

for any feasible residual capacity $\boldsymbol{r}$ [5], [6]. By the definition of $g(\boldsymbol{r})$, we have $\partial g/\partial r_{ij} = q_{ij} c_{ij}^{\beta-1}/r_{ij}^{\beta}$. Then (15) holds for any feasible residual capacity $\boldsymbol{r}$. □

**Remark 3.** Based on Theorem 6, the solution to Network$(\mathcal{G}, \boldsymbol{D}, \mathbf{1})$ is proportional load balancing. If $q_{ij}, \forall(i,j)$ are all integers, the solution to Network$(\mathcal{G}, \boldsymbol{D}, \boldsymbol{q})$ can be constructed as follows: for each $(i,j)$, replace the single link $(i,j)$ with $q_{ij}$ identical sub-links, calculate the proportional load balancing allocation over the $\sum_{(i,j)\in\mathcal{E}} q_{ij}$ traffic, and then provide link $(i,j)$ with the aggregate residual capacity allocated to its $q_{ij}$ associated sub-links. The residual capacity *per unit charge* is then proportional load balancing.

Next we discuss the popular TE criterion of minimizing MLU, which is known to be oversensitive to individual bottleneck links [10]. It does not penalize solutions that force traffic to traverse very long paths, either. We first use an example to illustrate that minimizing MLU is *not* a well-defined load balancing criterion.

Considering the example shown in Fig. 1a, there are an infinite number of possible traffic distributions minimizing MLU, shown in the last column of Table 1. How to evaluate these optimal traffic distributions? Let us turn to the min-max load balancing rule first proposed in this paper. A residual capacity $\hat{\boldsymbol{r}}$ is considered as *min-max* load balancing if it is feasible, and for any feasible residual capacity $\boldsymbol{r}$, the following condition holds: if $r_{ij} > \hat{r}_{ij}$ for some $(i,j) \in \mathcal{E}$, then there exists $(u,v) \in \mathcal{E}$ such that $(\hat{r}_{uv}/c_{uv}) \leq (\hat{r}_{ij}/c_{ij})$ and $r_{uv} < \hat{r}_{uv}$.

We now show that a min-max load balancing residual capacity $\hat{\boldsymbol{r}}$ also minimizes MLU. Assume that $\hat{\boldsymbol{r}}$ is min-max load balancing and does not minimize MLU, which means there exists a feasible residual capacity $\boldsymbol{r}$ such that

$$\max_{(i,j)\in\mathcal{E}} \left(1 - \frac{r_{ij}}{c_{ij}}\right) < \max_{(i,j)\in\mathcal{E}} \left(1 - \frac{\hat{r}_{ij}}{c_{ij}}\right). \tag{16}$$

Let $(i,j) = \arg\max_{(i,j)\in\mathcal{E}}(1 - \hat{r}_{ij}/c_{ij})$. By (16), we have $(1 - r_{ij}/c_{ij}) < (1 - \hat{r}_{ij}/c_{ij})$. For $\hat{\boldsymbol{r}}$ is min-max load balancing, there exists $(u,v) \in \mathcal{E}$ such that $(1 - \hat{r}_{uv}/c_{uv}) \geq (1 - \hat{r}_{ij}/c_{ij})$ and $(1 - r_{uv}/c_{uv}) > (1 - \hat{r}_{uv}/c_{uv})$, which contradicts (16).

We show the min-max load balancing residual capacity with the example in Fig. 1a in the fifth column in Table 1, where the min-max load balancing rule reduces the second maximum link utilization to 50 percent. Yet similar to minimizing MLU, the min-max load balancing traffic distribution does not penalize solutions that force traffic to traverse very long paths, e.g. path 1-2-3 and path 1-3 for the ingress-egress pair $(1,3)$. If the link capacities are five times higher, then it would not be worthy to send the traffic from node 1 through a detour over node 2 to node 3. Because it does not really matter that we reduce the second maximum link utilization from 20 to 10 percent.

**Remark 4.** We can show that the min-max load balancing emerges with $\beta$ converging to infinity for $(\boldsymbol{q}, \beta)$ proportional load balancing.

## 5 Implementation Issues

The generalized optimal TE framework in (2) can be realized via link-state routing protocols or MPLS. In this section, we present potential ways to implement our theoretical achievements. We try to leverage existing protocols and mechanisms as much as possible for the benefits of compatibility and deployability.

### 5.1 Overview

In our previous work [1], we realize the optimal TE in a distributed manner, where each router independently computes link weights and forwards packets accordingly. Considering the overhead imposed on routers by such a distributed approach, in this paper, we resort to a centralized way as commonly-used in the conventional TE, i.e., a centralized controller is employed to make optimal routing decisions.

After collecting the necessary information (e.g., network topology and traffic demands), the controller solves the optimal TE problem to derive updated network configurations (e.g., OSPF link weights or MPLS tunnels), and then disseminates these configurations to corresponding routers. In practice, the controller can collect the topology and link load information from OSPF's link state advertisements (LSAs).

Since multiple shortest paths may exist to carry on the traffic of the same ingress-egress pair, a major challenge to optimal TE implementation arises, i.e., how to determine an appropriate splitting fraction on each of these paths. Inspired by the encouraging results by Xu et al. [28], the NEM theory is employed here to achieve an efficient and flexible splitting scheme, with a noticeable difference that we split traffic only among multiple *shortest* paths other than all *available* paths [28].

### 5.2 OSPF-Based Approach

By incorporating the NEM theory into the NUM framework in Section 3, we develop shortest-paths penalizing exponential flow-splitting (SPEF), a practical link-state routing protocol with hop-by-hop forwarding. SPEF requires two sets of link weights. The first-set link weights (see Section 3.2) are used to compute the shortest path(s) of every ingress-egress pair and weights in the second-set (introduced later) are used for routers to independently construct the traffic splitting function when ECMP exists.

*The traffic splitting function for SPEF.* Now we focus on leveraging the NEM theory to illustrate how to derive the second-set link weights and then construct the traffic splitting function.

Assume that we have obtained the first-set link weights $\boldsymbol{w}$ and the residual capacity $\hat{\boldsymbol{r}}$ by solving TE$(\mathcal{G}, \boldsymbol{D}, V)$. Accordingly, the set of all-pair shortest paths in terms of the

first-set link weights is denoted by $\mathrm{ON} = \{\mathrm{ON}_t : t \in \mathcal{N}\}$, where $\mathrm{ON}_t$ is the set of the shortest paths from every node $i \in \mathcal{N}$ to node $t$. When ECMP exists for a specific ingress-egress pair $(s_m, t_m)$, we adopt an exponential-weighted flow splitting scheme based on NEM, because it allows each router to independently compute the desired traffic splitting ratios using only alternative link weights [28]. As such, SPEF achieves the network-wide traffic engineering objective, yet keeps the simplicity and scalability of link-state routing protocols.

If path $r \in \mathrm{ON}$ is the shortest path from $s_m$ to $t_m$, we denote it by $r \in m$ for simplicity. Let the traffic split fraction of path $r$ be $p_r$, i.e. $\sum_{r:r\in m} p_r = 1$. Maximizing the relative entropy [6] of the traffic splitting vector can be formulated as

NEM$(\mathrm{ON}, \boldsymbol{D}, \hat{\boldsymbol{f}})$:

$$\underset{p_r}{\text{maximize}} \; -\sum_m d_m \sum_{r:r\in m} p_r \log p_r \tag{17a}$$

$$\text{subject to} \sum_m \sum_{r:r\in m,(i,j)\in r} d_m p_r \le \hat{f}_{ij}, \; \forall (i,j) \in \mathcal{E} \tag{17b}$$

$$\sum_{r:r\in m} p_r = 1, \; m = 1, \ldots, M \tag{17c}$$

where $\hat{\boldsymbol{f}} = \boldsymbol{c} - \hat{\boldsymbol{r}}$.

**Remark 5.** Compared with the earlier NEM problem for PEFT in [28], NEM$(\mathrm{ON}, \boldsymbol{D}, \hat{\boldsymbol{f}})$ only splits the traffic on the ECMPs determined by the first-set link weights, which enables the SPEF protocol to maintain the shortest-path nature of OSPF. On the other hand, PEFT adoptes the traffic-splitting function to get the total outgoing traffic flows (destined to $t_m$) traversing link $(i, j)$, which means we must split the traffic to each possible path for $(s_m, t_m)$.

We now show that the optimal solution to NEM$(\mathrm{ON}, \boldsymbol{D}, \hat{\boldsymbol{f}})$ can be realized in a hop-by-hop forwarding manner. Let $\hat{\boldsymbol{p}} = (\hat{p}_r, \forall r \in \mathrm{ON})$ be the optimal solution to NEM$(\mathrm{ON}, \boldsymbol{D}, \hat{\boldsymbol{f}})$ and $\boldsymbol{v}$ the Lagrangian multiplier vector associated with (17b). Hereafter we refer to $\boldsymbol{v}$ as the second-set link weights. Then by the convex optimization theory (See [5], p. 281, Theorem 28.3), $\hat{\boldsymbol{p}}$ solves

$$\begin{aligned} &\underset{p_r}{\text{maximize}} - \sum_m \sum_{r:r\in m} d_m (p_r \log p_r + q_r p_r) + \sum_{(i,j)\in\mathcal{E}} v_{ij} \hat{f}_{ij} \\ &\text{subject to} \sum_{r:r\in m} p_r = 1, \; m = 1, \ldots, M, \end{aligned} \tag{18}$$

where $q_r = \sum_{(i,j)\in r} v_{ij}$ is the length of path $r$ in terms of the second-set link weights $\boldsymbol{v}$. Note that the objective function in (18) is separable for a given $\boldsymbol{v}$. We get the solution to (18) by solving

$$\begin{aligned} &\underset{p_r}{\text{maximize}} - d_m \sum_{r:r\in m} \left(p_r \log p_r + q_r p_r\right) \\ &\text{subject to} \sum_{r:r\in m} p_r = 1 \end{aligned}$$

for each $m$ separately. Then there exists $\nu_m$ such that

$$d_m (1 + \log p_r + q_r) + \nu_m = 0$$

TABLE 2
Path Table for Node $i$ to Destination $t$ under SPEF

| Next-hop | Lengths of ECMPs through link $(i,$ Next-hop$)$ to $t$ in view of the second-set link weights |
|---|---|
| $j_1$ | $(q_{11}, \ldots, q_{1n_1})$ |
| $\vdots$ | $\vdots$ |
| $j_{l_i}$ | $(q_{l_i 1}, \ldots, q_{l_i n_{l_i}})$ |

and $\sum_{r:r\in m} p_r = 1$. Based on these above conditions, we get

$$\hat{p}_r = \frac{e^{-q_r}}{\sum_{r':r'\in m(r)} e^{-q_{r'}}}, \; \forall \; r, \tag{19}$$

where $m(r)$ is ingress-egress pair $(s_m, t_m)$ with $r \in m$.

The notion of the traffic splitting function was introduced in [27] to succinctly describe link-state routing protocols. Traffic-splitting function $\Gamma_t(i, j)$ indicates the amount of traffic that node $i$ forwards via outing link $(i, j)$ to $t$. Here we first need to establish a path table for node $i$ to destination $t$ as shown in Table 2. There are $l_i$ next-hops for node $i$ in $\mathrm{ON}_t$. Let $n_k$ denote the number of shortest paths from node $i$ across node $j_k$ to node $t$, and $q_{kh}$ the length of the $h$th path in terms of the second-set link weight $\boldsymbol{v}$ from node $i$ through node $j_k$ to node $t$. According to (19), the traffic splitting function is

$$\Gamma_t(i, j_k) = \frac{\sum_{h=1}^{n_k} e^{-q_{kh}}}{\sum_{k'=1}^{l_i} \sum_{h=1}^{n_{k'}} e^{-q_{k'h}}}, \; k = 1, \ldots, l_i. \tag{20}$$

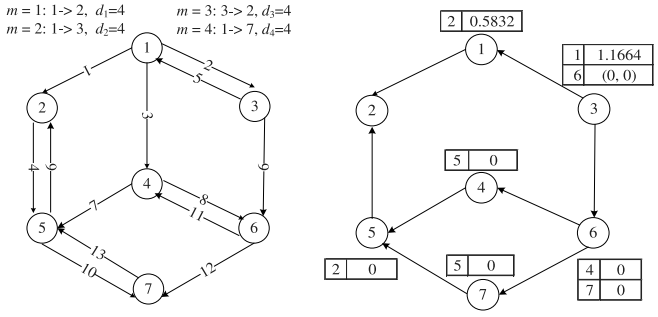
Note that if there is only one next hop $j$ for $i$ in $\mathrm{ON}_t$, we have $\Gamma_t(i, j) = 1$ according to (20).

*Routing algorithms for SPEF*. Now we are on the position to design algorithms for SPEF to realize the optimal TE. As stated earlier, we rely on a central controller to periodically calculate the up-to-date routing configurations. Therefore, a centralized algorithm for the controller is shown in Algorithm 1, which first takes the necessary input information (e.g., topology and traffic demands) and then calculates the first- and the second-set link weights by solving the TE and the NEM problem, respectively. These two convex optimization problems can be solved in a centralized manner in polynomial time. Alternative distributed solutions to these problems, as well as their convergence behaviors, can be found in [1]. Note that the operators can flexibly adjust their TE goals by changing the input utility function $V(\boldsymbol{r})$ accordingly.

**Algorithm 1.** *Centralized Computation*

**Input:** $\mathcal{G}, \boldsymbol{c}, \boldsymbol{D}, V$
1: Solve TE$(\mathcal{G}, \boldsymbol{D}, V)$ to obtain the first-set link weights $\boldsymbol{w}$ and residual capacity $\hat{\boldsymbol{r}}$.
2: **for** each destination node $t \in \mathcal{N}$ **do**
3: Run Dijkstra's algorithm with the first-set link weights to get the set of the shortest paths $\mathrm{ON} = \{\mathrm{ON}_t : t \in \mathcal{N}\}$.
4: **end**
5: Solve NEM$(\mathrm{ON}, \boldsymbol{D}, \hat{\boldsymbol{f}})$ to get the second-set weights $\boldsymbol{v}$.

A router learns the network topology and two sets of link weights from the flooding of LSAs (Section 5.4). Therefore, it




(a) Network topology and traffic demands

(b) The shortest path set ($\mathrm{ON}_2$) and the SPEF forwarding table for destination 2

Fig. 2. A simple example to illustrate the SPEF routing.

can independently construct the forwarding table for SPEF as shown in Algorithm 2. After getting the shortest paths in terms of the first-set link weights to all destinations (Line 1), it calculates how to split the traffic for every destination $t$ over its outgoing links (i.e., next-hops) (Lines 3-4). The time complexities to calculate the shortest paths with Dijkstra's algorithm (Line 1) and to create all entries (Lines 2-5) are $O(E + N\ \log N)$ and $O(N^2)$, respectively. Therefore, the total time complexity of Algorithm 2 is $O(N^2)$. Algorithms 1 and 2 enable SPEF to achieve the optimal TE with hop-by-hop forwarding.

**Algorithm 2.** *Constructing Forwarding Table for SPEF*

**Input:** $\boldsymbol{w}, \boldsymbol{v}$

1: Run Dijkstra's algorithm with the first-set link weights $\boldsymbol{w}$ to get the set of the shortest paths $\mathrm{ON}_t$ for all $t$.
2: **for** each destination node $t \in \mathcal{N}$ **do**
3: Calculate the length of each path in $\mathrm{ON}_t$ in terms of the second-set link weights $\boldsymbol{v}$ and construct Table 2 by classifying $\mathrm{ON}_t$ according to their next hops.
4: Get the traffic splitting by Eq. (20) and create a routing entry $< destination, next\_hop, splitting\_ratio >$.
5: **end**

*Case study: the SPEF routing*. We use a classical example for TE [20] as shown in Fig. 2a to illustrate SPEF routing, where each link has a capacity of 5 units. There are four ingress-egress pairs and each needs a bandwidth of 4 units. For simplicity, we omit the six links unused. The numbers on the links are the link indices.

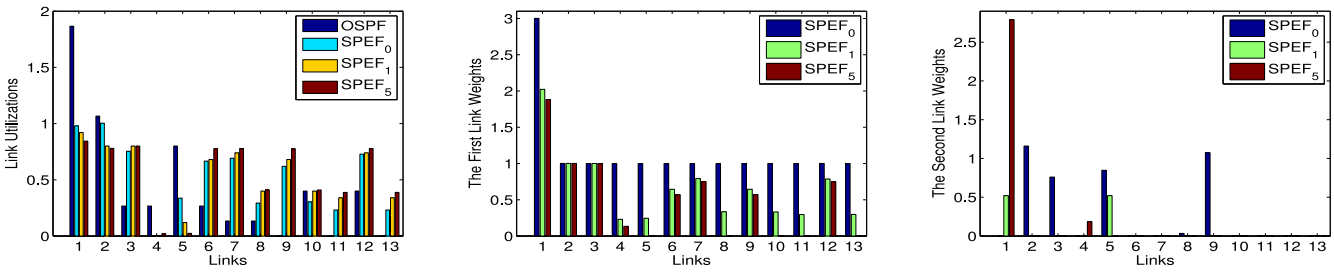
Considering Destination 2, Fig. 2b shows the shortest path set in view of the first-set link weights for SPEF with $\beta = 1$ and the forwarding tables of each node. We take Node 3 as an example. There are two next-hops, i.e., Nodes 1 and 6. Only one shortest path with a cost of 1.1664, in view of the second-set link weights, reaches Destination 2 via Node 1. Two shortest paths traverse Node 6 to Destination 2. Both of them have a cost of 0 in view of the second-set link weights. Then the router at Node 3 independently calculates the traffic splitting according to (20), i.e. $\Gamma_2(3,1) = e^{-1.1664}/(e^{-1.1664} + 2e^0) = 0.1348$ and $\Gamma_2(3,6) = 2e^0/(e^{-1.1664} + 2e^0) = 0.8652$.

Unless explicitly specified, hereafter we use OSPF to refer to a benchmark, which sets the weight of each link inversely proportional to its capacity and evenly splits traffic among the set of the next-hops in the ECMP. Fig. 3a shows the link utilizations for OSPF and SPEF with different parameters $\beta$, where Link 1 is the bottleneck link. The congestion occurs at Link 1 for OSPF. The utilization of Link 1 decreases with $\beta$ for SPEF. In Fig. 3b, the first-set link weight of Links 1 is 3 and other links' first-set link weights are all 1 when $\beta = 0$. For $\beta = 1$, we can see from Fig. 3c that all the second-set link weights are zero besides Links 1 and 5. The increase of the second-set weight of Link 1 with $\beta$ shows that we route less traffic through Link 1 with a larger $\beta$.

### 5.3 MPLS-Based Approach

An alternative way to achieve the optimal TE is the MPLS-based approach [39]. In MPLS, label-switched paths are set up using a signaling protocol for packet forwarding, e.g., Resource ReSerVation Protocol (RSVP) [41]. Packets are classified into different forwarding equivalence classes (FECs) at the edge router when they first enter an MPLS network, and then forwarded along corresponding LSPs. Multiple LSPs may serve the same FEC to support multi-path routing.

Besides the OSPF-based implementation (i.e., SPEF), our framework provides valid MPLS-based TE schemes as well. With a slight modification, Algorithm 1 can be used by the central controller to calculate the optimal MPLS tunnels. After solving the NEM in Line 5 of Algorithm 1, the controller should continue to get $\hat{p}_r$ for each path $r \in \mathrm{ON}$ using (19), which can be viewed as the ratio of the traffic routed along path $r$ to the total traffic demand with the same ingress-egress pair. Then, we take every ingress-egress pair $(s_m, t_m)$ as an FEC. Accordingly, each path $r \in \mathrm{ON}$ from $s_m$ to $t_m$ can be set up as an LSP by installing a label forwarding entry at routers along the LSP. The




(a) The link utilizations

(b) The first-set link weights

(c) The second-set link weights

Fig. 3. Results for the topology in Fig. 2a with different $\beta$ (we shrink each value in (b) by a fraction of $5^{\beta-1}$ to accommodate them in one plot).

TABLE 3
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 |

hashing mechanism [7] can be configured with different weights to support various splitting fractions over multiple LSPs for the same FEC.

### 5.4 Discussion

Since SPEF requires an additional set of link weights, we can separate the current 16-bit LSA message into two 8-bit spaces to represent the first- and the second-set link weights of each link, respectively. By this way, SPEF claims no change on the format of LSA messages, but does require a modification to the control plane of each router to recognize these weights and then carry on local computation. The major overhead arisen here comprises of the storage overhead of the path table in Table 2 and the computation overhead of the splitting ratios in Eq. (20). Assume that there are 256 edge nodes as destinations in a network (a reasonably large number given that an OSPF domain generally contains no more than 1,000 nodes [37]). For a node $i$, suppose the average degree (i.e., next-hops $l_i$) is 16 and each next-hop has on average eight ECMPs (i.e., $n_k$) in terms of the second link weights. If the length of each path, $q_{kh}$, is stored in 2 Bytes ($1 \sim 65{,}535$), then each entry in Table 2 takes up $2 \times 8 + 2$ (Bytes for the next-hop segment)=18 Bytes. So the total memory required is $18 \times 16 \times 256 = 0.074$ MB, which is very small. According to Eq. (20), the computation of splitting ratios for each destination at node $i$ consists of exponent arithmetic, addition and division, which can be done within several instructions in the processor. Considering parallel execution for multiple destinations, the computation overhead should not be a concern.

The online MPLS-based TE solutions (e.g., MIRA [18] and REPLEX [8]) are proposed to handle dynamic traffic demands without any priori information, whereas our MPLS-based implementation is likely to be categorized as an *offline* approach, which aims to globally optimize network-wide routes for given traffic demands. As the network traffic demands vary over time, the offline algorithm can be conducted periodically based on the collected traffic information. Intuitively, the choice of periodicity balances the trade-off between optimality and overhead, which will be further explored in Section 6.

## 6 Performance Evaluation

### 6.1 Network Setup

Two real backbone networks and several synthetic networks are involved in simulations, the properties of which are summarized in Table 3. The Abilene network has 28 10-Gbps links, while the Cernet2 [44] network has eight 10-Gbps links and 36 2.5-Gbps links. The traffic demands of Abilene network are available online [43], and those of Cernet2 network are generated using a gravity model. The link loads required by the gravity model are derived from the sample Netflow data collected from Jan. 10th to 16th, 2010.

In addition, we also generate synthetic networks using GT-ITM with similar configurations in [9]. Each two-level hierarchical network consists of two kinds of links: local access links and long distance links with 1- and 5-unit bandwidth, respectively. As to the random networks, the probability of having a link between two nodes is a constant, and all link bandwidths are 1 unit. Traffic demands for all synthetic networks are randomly generated using the gravity model.

### 6.2 Network Utility under Matlab-Based Simulations

We conduct Matlab-based simulations to explore the utility that SPEF achieves under different traffic demand scenarios, where the utility of OSPF is taken as a benchmark. PEFT is not involved for comparison because theoretically it achieves the same optimal utility with SPEF. Without losing generality, we use the utility function in (10) with $\boldsymbol{q} = \mathbf{1}$ and $\beta = 1$ to determine the first-set link weights for SPEF.

For each network, traffic demands increase proportionally to simulate different congestion levels. Fig. 4 shows the normalized utility when varying the network load for different topologies, where $x$-axis denotes the network load calculated as the ratio of total demands to the total link capacities. The normalized utility is $\sum_{(i,j)\in\mathcal{E}} \log\,(1 - u_{ij})$ ($u_{ij}$ is the link utilization) if MLU $< 1$, or $-\infty$ otherwise.

In Fig. 4, the utility difference between SPEF and OSPF becomes remarkable as network load increases (the lower the better). When MLU under OSPF is greater than 1, OSPF breaks down while SPEF still works well. This phenomenon is more obvious in large-scale synthetic networks, e.g., when the network load is greater than 0.09 in the Rand100 network in Fig. 4g. These facts indicate that OSPF is suitable for networks with relatively low network load, and SPEF routing is more robust than OSPF in sustaining heavier traffic workloads.

### 6.3 Performance under NS2-Based Simulations

In this section, we present a series of NS2 [45] simulations for performance comparison of SPEF, OSPF and PEFT, from the point of view of the link utilization and flow delay. In addition, we also explore the routing stability and configuration overhead of SPEF for the MPLS-based implementation.

Abilene network is used in NS2 simulations, because its real traffic matrices in every 5 minutes are available [43]. The original traffic matrices are considered as the *light* load. To emulate a *heavy* load, we scale up the original traffic matrices until the network load reaches 0.16, because a heavier load will make OSPF infeasible as shown in Fig. 4a. Each simulation lasts for 200 s and the propagation delay is 30 ms for all links.

For PEFT, as it is impossible to enumerate all possible paths, we select for each ingress-egress pair a candidate path set consisting of the top-three shortest paths in terms of link hops; then the path-based optimal solution to the

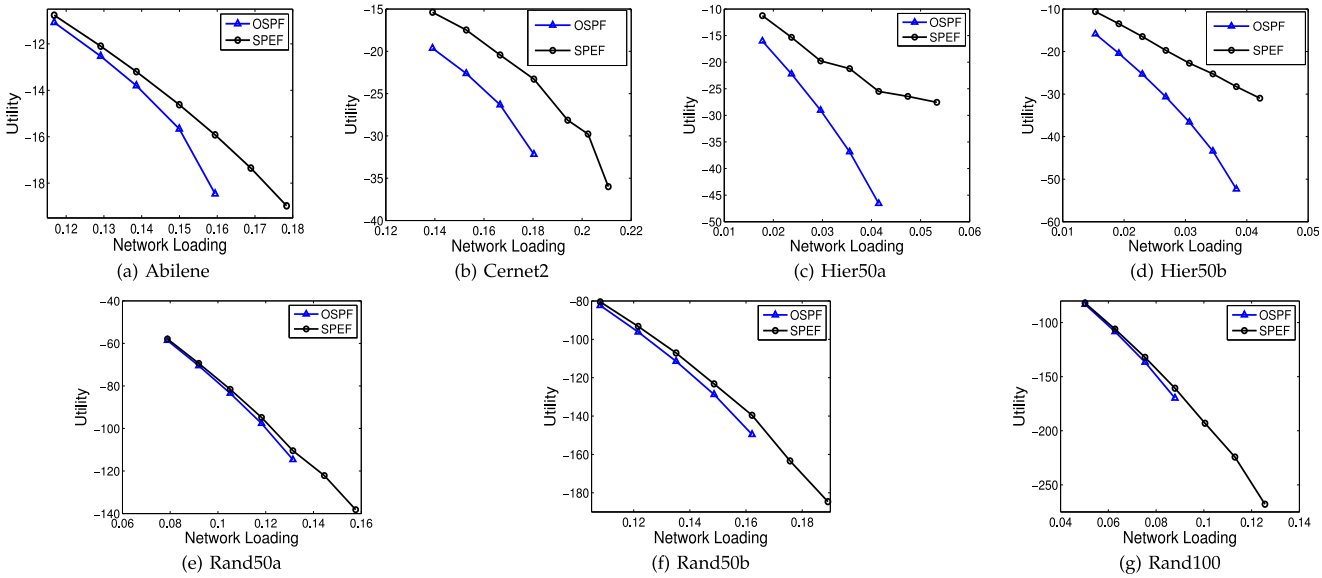



Fig. 4. Utility versus network loading for SPEF and OSPF.

multi-commodity problem with a piecewise-linear objective function is regarded as the optimal routes of PEFT. For OSPF, since each link in our simulation has the same capacity, Cisco's InvCap [42] results in the same weight for all links. Therefore, the shortest path(s) in terms of link weights are equivalent to the smallest hop-count paths. SPEF is expected to achieve better load balancing at an expense of enlarging path length in terms of hops for partial ingress-egress pairs.

*Link utilization*. To explore the link utilization for the three protocols, we select a series of traffic matrices within 6 hours (from 8 am to 2 pm) on Oct. 20th, 2004. The time-varying MLUs for three protocols are shown in Figs. 5 and 6, where y-axis represents the MLU of each 5-min interval. The major advantage that SPEF offers is the MLU improvement over the other two protocols, especially for the light load.

Then, we randomly select a time interval to have a deep look at how the traffic is distributed over all links for both light and heavy loads. Due to the space limitation, we only present the results of heavy load, as shown in Fig. 7. The case of light load has similar results. SPEF can slightly decrease the utilization of overloaded links by rerouting the traffic over links with much lower utilization. PEFT has similar performance as that of OSPF in the heavy traffic scenario, which is consistent with the results in Fig. 6.

*Flow delay*. Different traffic distributions may result in diverse routes for the same ingress-egress pair. Here we refer to the packets of the same ingress-egress pair as the same network *flow*. Using traffic matrices in the same time interval as in Fig. 7, we calculate the flow delay for each protocol as the total packet delay divided by the number of packets. Results for three protocols are exhibited in Figs. 8 and 9. In both cases, OSPF has the smallest average flow delay. SPEF is very close to OSPF and outperforms PEFT in both scenarios. This is because the network is actually not heavily congested under both loads, and the flow delay thus largely depends on the flow's path length in terms of link hops.

Now we attempt to verify that the flow delay is largely affected by the path length. Since multiple paths with evenly (i.e., OSPF) or exponentially splitting (i.e., PEFT and SPEF) ratios might be used by the same ingress-egress pair, we use the weighted average path length (WAPL) of each flow as a comparison metric, which is calculated by summing the product of the hop counts of each path and its splitting fraction. Therefore, WAPL of a flow reflects the path length in which the majority of traffic in this flow traverses. CDFs of WAPL for the three protocols in both light and heavy load cases are plotted in Figs. 10 and 11. As expected, the curves of three protocols have similar trends as shown in Figs. 8 and 9. The gap between two curves for OSPF and SPEF in Fig. 11 reveals that during the interval with heavy load, SPEF attempts to balance the overall traffic distribution (Fig. 7) while leading to longer paths for some ingress-egress pairs.

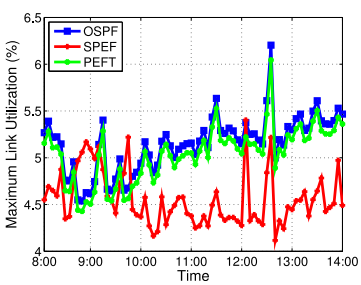



Fig. 5. Maximum link utilization over time with light load.

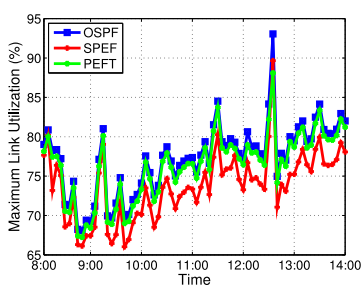



Fig. 6. Maximum link utilization over time with heavy load.

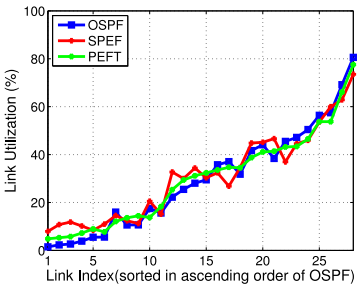



Fig. 7. Link utilization with heavy load (links are in ascending order under OSPF).

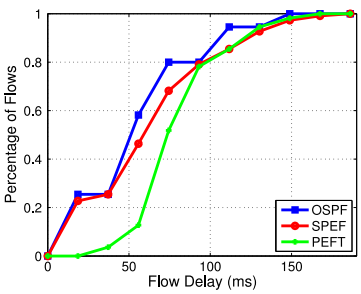



Fig. 8. CDF of the average flow delay with light load.

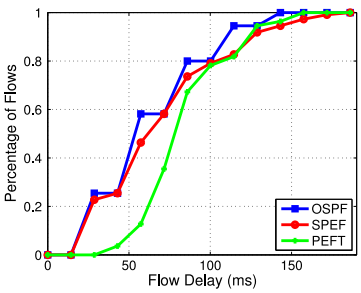



Fig. 9. CDF of the average flow delay with heavy load.

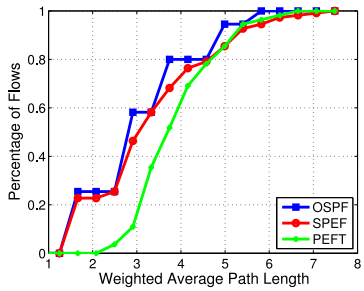



Fig. 10. CDF of the average path length with light load.

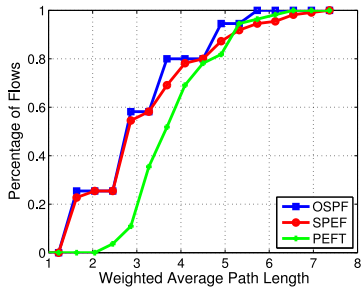



Fig. 11. CDF of the average path length with heavy load.

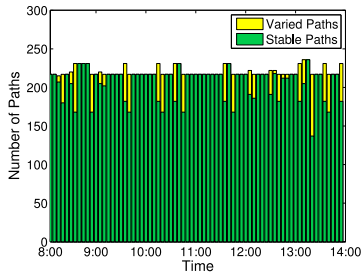



Fig. 12. Routing stability with heavy load.

*Reconfiguration overhead*. Assume that we compute and reconfigure the optimal paths (i.e., LSPs) every 5 minutes. Using the traffic matrices of heavy load during 6 hours of a day, we first explore the routing stability in terms of stable paths between two consecutive reconfigurations, as shown in Fig. 12. The bar in each time interval denotes the total number of paths for all ingress-egress pairs during that time interval, and the total number is divided into the number of stable paths (green) and that of varied paths (yellow). The ratio of the total number of varied paths over the total number of all ingress-egress paths is 6.67 percent. The routing derived from the generalized framework is quite stable even with the highest reconfiguration frequency.

Intuitively, the reconfiguration frequency is critical to balance the trade-off between the overhead and the optimality it can achieve, e.g., frequently reconfiguration makes routing react to the up-to-date traffic variation at an expense of high overhead. Several reconfiguration intervals are considered here, namely 5, 15, 30, 60, 90 and 120 mins. To get traffic matrices for each interval longer than 5 mins, we simply sum up the 5-min matrices during that interval, e.g., summing up three consecutive 5-min matrices as a single 15-min matrix.

Since establishing new LSPs and removing expired LSPs account for the majority of management overhead, we use the total number of varied paths during the 6 hours as an overhead metric, e.g., the overhead for the 5-min interval is calculated as the total number of varied paths in Fig. 12.

We refer to the link utilization with the 5-min reconfiguration interval as the optimal result, then the optimality gap of another reconfiguration interval is calculated as

$$Gap = \frac{1}{T}\sum_{t=1}^{T} \parallel \boldsymbol{u}(t) - \hat{\boldsymbol{u}}(t) \parallel_2, \tag{21}$$

where $\| \cdot \|_2$ denotes the $L_2$ norm. $\hat{u}(t)$ represents the optimal link utilization (i.e., with the 5-min interval) at time $t$, and $\boldsymbol{u}(t)$ represents the result at time $t$ with one of the other reconfiguration intervals, e.g., 15-min.

Fig. 13 exhibits the relationship between the optimality gap and the reconfiguration overhead, where the $x$-axis is the optimality gap calculated with Eq. (21). A curve is plotted by fitting all markers. The reconfiguration overhead drops dramatically from the 5-min to the 30-min. If the interval continues to increase, the optimality gap grows quickly without much decrease in reconfiguration overhead. Therefore, an operator can select an appropriate reconfiguration interval to balance the overhead and the optimality.

### 6.4 Summary of the Simulation Results

We summarize key observations from the simulations as follows. First, SPEF achieves optimal utilities under a

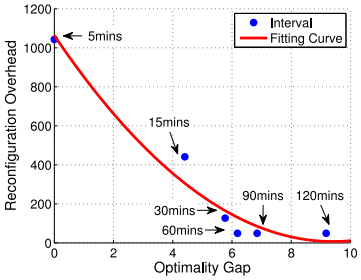



Fig. 13. Optimality versus reconfiguration overhead with heavy load.

wide range of network loads, particularly under the loads with which OSPF is incapable of dealing. Second, compared with OSPF and PEFT, SPEF leads to better load balancing in terms of network-wide link utilizations without introducing severe per-flow delays. Finally, the time-varying stability of the routing derived from our framework enables network providers to balance the recomputation overhead and the optimality.

## 7 Conclusions

In this paper, we successfully generalize the classic results of the fairness criteria in rate control to load balancing criteria in traffic engineering. The results obtained under this background are novel for TE and load balancing, though they may have corresponding versions for rate control. With the help of the framework NEM [28], a new routing protocol SPEF is proposed, which can be viewed as an application of the proposed general framework. In our opinion, SPEF can be considered as a perfect solution to the optimal TE based on OSPF. Our results possess the potential to be applied to many other protocols for TE. We are also interested in further optimizing the computational complexity of SPEF.

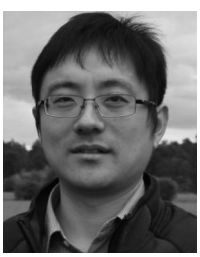
## Acknowledgments

This work has been supported in part by Canada NSERC Discovery Grant, NSFC Project (61170292, 61472212, 61172060, 61370192, 61432015), National Science and Technology Major Project (2012ZX03005001), 973 Project of China (2012CB315803), 863 Project of China (2013AA013302, 2015AA010203), EU MARIE CURIE ACTIONS EVANS (PIRSES-GA-2010-269323 and PIRSES-GA-2013-610524), and Beijing Institute of Technology Research Fund Program for Young Scholars. Dr. Meng Shen is the corresponding author.

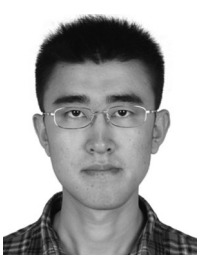
## References

[1] K. Xu, H. Liu, J. Liu, and M. Shen, "One more weight is enough: Toward the optimal traffic engineering with OSPF," in *Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst.*, 2011, pp. 836–846.
[2] D. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 42–47, Dec. 1999.
[3] W. Ben-Ameur, É. Gourdin, B. Liau, and N. Michel, "Routing strategies for IP networks," *Telektronikk Mag.*, vol. 97, no. 2/3, pp. 145–158, 2001.
[4] D. P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.
[5] R. Tyrrell Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.
[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[7] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing-based schemes for Internet load balancing," in *Proc. IEEE Conf. Comput. Commun.*, 2000, pp. 332–341.
[8] S. Fisher, N. Kammenhuber, and A. Feldmann, "REPLEX: Dynamic traffic engineering based on wardrop routing policies," in *Proc. ACM CoNEXT Conf.*, 2006, pp. 1–12.
[9] B. Fortz and M. Thorup, "Increasing Internet capacity using local search," *Comput. Optim. Appl.*, vol. 29, no. 1, pp. 13–48, 2004.
[10] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 118–124, 2002.
[11] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE Conf. Comput. Commun.*, 2000, pp. 519–528.
[12] E. Gourdin and O. Klopfenstein, "Comparison of different QoS-oriented objectives for multicommodity flow routing optimization," in *Proc. 13th Int. Conf. Telecommun.*, 2006.
[13] J. He, M. Chiang, and J. Rexford, "Towards robust multi-layer traffic engineering: Optimization of congestion control and routing," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 5, pp. 868–880, Jun. 2007.
[14] J. He, M. Suchara, and M. Chiang, "Rethinking Internet traffic management: From multiple decompositions to a practical protocol," *Proc. ACM CoNEXT Conf.*, 2007, pp. 1–12.
[15] J. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. New York, NY, USA: Springer-Verlag, 2001.
[16] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proc. Conf. Appli., Technol., Archit., Protocols Comput. Commun.*, 2005, pp. 253–264.
[17] F. P. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.
[18] M. Kodialam and T. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *Proc. IEEE Conf. Comput. Commun.*, 2000, pp. 884–893.
[19] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
[20] G. Rétvári, J. Bíró, and T. Cinkler, "On shortest path representation," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1293–1306, Dec. 2007.
[21] S. Shenker, "Fundamental design issues for the future Internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1176–1188, Sep. 1995.
[22] Z. Wang, Y. Wang, and L. Zhang, "Internet traffic engineering without full mesh overlaying," in *Proc. IEEE Conf. Comput. Commun.*, 2001, pp. 565–571.
[23] S. Srivastava, G. Agrawal, M. Pioro, and Medhi, "Determining link weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach," *IEEE Trans. Netw. Serv. Manage.*, vol. 2, no. 1, pp. 9–18, Nov. 2005.
[24] C. Villamizar. OSPF Optimized Multipath (OSPF-OMP). [Online]. Available: http://www.fictitious.org/omp, 1999.
[25] H. Wang, H. Xie and L. Qiu, "COPE: Traffic engineering in dynamic networks," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2006, pp. 99–110.
[26] A. Sridharan, R. Guerin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 234–247, Apr. 2005.
[27] D. Xu, M. Chiang, and J. Rexford, "DEFT: Distributed exponentially-weighted flow splitting," in *Proc. IEEE Conf. Comput. Commun.*, 2007, pp. 71–79.
[28] D. Xu, M. Chiang, and J. Rexford, "Link-state routing with hop-by-hop forwarding achieves optimal traffic engineering," in *Proc. IEEE Conf. Comput. Commun.*, 2008, pp. 1139–1147.
[29] K. Xu, H. Liu, J. Liu, and J. Zhang, "LBMP: A logarithm-barrier-based multipath protocol for Internet traffic management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 456–470, Mar. 2011.
[30] M. Wang, C. W. Tan, W. Xu, and A. Tang, "Cost of not splitting in routing: Characterization and estimation," *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1849–1859, Dec. 2011.
[31] N. Michael, A. Tang, and D. Xu, "Optimal link-state hop-by-hop routing," in *Proc. IEEE 21st Int. Conf. Netw. Protocols*, 2013, pp. 1–10.
[32] F. P. Tso and D. P. Pezaros, "Improving data center network utilization using near-optimal traffic engineering," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1139–1148, Jun. 2013.

[33] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, "Intra-data-center traffic engineering with ensemble routing," in *Proc. IEEE Conf. Comput. Commun.*, 2013, pp. 2148–2156.
[34] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multiPath: An algorithmic perspective," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 1590–1598.
[35] V. Foteinos, K. Tsagkaris, P. Peloso, L. Ciavaglia, and P. Demestichas, "Operator-friendly traffic engineering in IP/MPLS core networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 11, no. 3, pp. 333–349, Sep. 2014.
[36] W. Su, C. Liu, C. Lagoa, H. Che, K. Xu, and Y. Cui, "Integrated, distributed traffic control in multidomain networks," *IEEE Trans. Control Syst. Technol.*
[37] B. Movsichoff, C. Lagoa, and H. Che, "Decentralized optimal traffic engineering in connectionless networks, *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 293–303, Feb. 2005.
[38] J. Moy. OSPF Version 2 [Online]. Available: http://tools.ietf.org/rfc/rfc2328, 1998.
[39] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. [Online]. Available: http://www.ietf.org/rfc/rfc3031, 2001.
[40] G. Swallow, S. Bryant, and L. Andersson, Avoiding equal cost multipath treatment in MPLS networks. [Online]. Available: http://www.ietf.org/rfc/rfc4928, 2007.
[41] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource ReSerVation protocol (RSVP)–Version 1 functional specification. [Online]. Available: http://www.ietf.org/rfc/rfc2205, 1997.
[42] Cisco. Configuring OSPF. 1997.
[43] Yin Zhang's Abilene TM. [Online]. Available: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/, 2004.
[44] CERNET2. [Online]. Available: http://www.cernet2.edu.cn/index_en.htm, 2009.
[45] The network simulator. [Online]. Available: http://www.isi.edu/nsnam/ns/, 2011.



**Ke Xu** (M'02-SM'09) received the PhD degree from the Department of Computer Science, Tsinghua University, where he is currently a full professor. He has published more than 100 technical papers and held 20 patents in the research areas of next generation Internet, P2P systems, Internet of Things (IoT), network virtualization and optimization. He is a member of the ACM, and has guest edited several special issues in IEEE and Springer Journals. He is a senior member of the IEEE.



**Meng Shen** (M'14) received the BEng degree from Shandong University, Jinan, China, in 2009, and the PhD degree from Tsinghua University, Beijing, China, in 2014, both in computer science. He is currently an assistant professor at the Beijing Institute of Technology, Beijing, China. His research interests include network congestion control, traffic engineering and network virtualization. He is a member of the IEEE.

**Hongying Liu** received the PhD degree in mathematics from Xidian University, Xi'an, China, in 2000. She has been an associate professor of mathematics of systems at Beihang University, since 2003. Her research interests include optimization and applied probability focusing on applications in networks and statistical signal processing.

**Jiangchuan Liu** (S'01-M'03-SM'08) received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology, in 2003, both in computer science. He is currently a full professor at the School of Computing Science, Simon Fraser University, BC, Canada. He is an NSERC E.W.R. Steacie Memorial fellow, and an EMC-Endowed visiting chair professor of Tsinghua University, Beijing, China (2013-2016). From 2003 to 2004, he was an assistant professor at The Chinese University of Hong Kong. He is a co-recipient of ACM TOMCCAP Nicolas D. Georganas Best Paper Award 2013, ACM Multimedia Best Paper Award 2012, IEEE Globecom 2011 Best Paper Award, and IEEE Communications Society Best Paper Award on Multimedia Communications 2009. His research interests include multimedia systems and networks, cloud computing, social networking, online gaming, big data computing, crowdsourcing, wireless sensor networks, and peer-to-peer networks. He was on the editorial boards of *IEEE Transactions on Big Data, IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, IEEE Access, IEEE Internet of Things Journal*, Elsevier *Computer Communications*, and Wiley *Wireless Communications and Mobile Computing*. He is a senior member of the IEEE.

**Fan Li** received the PhD degree in computer science from the University of North Carolina at Charlotte, in 2008. She is currently an associate professor at the School of Computer Science, Beijing Institute of Technology, China. Her current research focuses on wireless networks, ad hoc and sensor networks, and mobile computing. She won Best Paper Awards from multiple conferences, such as IEEE MOBIHOC 2014, IEEE MASS 2013, and IEEE IPCCC 2013. She is a member of the IEEE and the ACM.

**Tong Li** received the BS degree in computer science from Wuhan University, China, in 2012. He is currently working toward the PhD degree at the Department of Computer Science, Tsinghua University. His research interests include network management and economics.