J. Parallel Distrib. Comput. 74 (2014) 1848–1857

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Multi-path utility maximization and multi-path TCP design

Phuong Luu Vo [a], Tuan Anh Le [b,a], Sungwon Lee [a], Choong Seon Hong [a,*], Byeongsik Kim [c], Hoyoung Song [c]

[a] Department of Computer Engineering, Kyung Hee University, Republic of Korea
[b] Department of Information Technology, Thudaumot University, Viet Nam
[c] Next Generation Communication Research Laboratory, ETRI, Daejeon, Republic of Korea

HIGHLIGHTS

- Propose a modified multi-path NUM model.
- Solve it by successive approximation approach in fluid model.
- Design a series of multi-path TCPs.
- Conduct experiments on mReno and compare it to the current MPTCP.

ARTICLE INFO

Article history:
Received 7 September 2011
Received in revised form
12 June 2013
Accepted 29 July 2013
Available online 7 August 2013

Keywords:
Multi-path NUM
Multi-path TCP
Successive approximation

ABSTRACT

The canonical multi-path network utility maximization (NUM) model which is extended directly from the single-path NUM has been studied widely in the literature. Most of the previous approaches do not specify the case of subflows on paths with different characteristics. Moreover, the transport protocol derived from the canonical multi-path NUM exhibits flappiness in the subflows because of the non-strictly convexity of the optimization problem.

This paper introduces a modified multi-path NUM model and proposes a novel approach to overcome the mentioned issues. Using Jensen's inequality, the multi-path NUM is approximated to a strictly convex and separable problem which can be solved efficiently by dual-based decomposition method. The algorithm successively solving a sequence of approximation problems is proven to converge at the global optimum of the original problem. Moreover, considering the separable form of the approximation utility and the dual-based nature of the proposed algorithm, the reverse engineering frameworks of the current TCPs are used to develop a series of multi-path TCPs that are compatible with corresponding regular single-path TCPs.

1. Introduction

In a multi-path environment with sets of sources $\mathcal{N}$ and links $\mathcal{L}$, each source can transmit data on several paths. In this work, $\boldsymbol{s}$ indicates the set of paths/subflows from source s and $\mathcal{R}_{s,i}$ is the i-th path of source s. The canonical multi-path network utility maximization (NUM) problem, which is directly extended from the single-path NUM, has the following form:

$$\text{Max.} \quad \sum_{s\in\mathcal{N}} U_s\left(\sum_{i\in\boldsymbol{s}} x_{s,i}\right)$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{N}} \sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} \le c_l, \quad \forall l \in \mathcal{L},$$

where U_s is the utility function associated with source s and $x_{s,i}$ is the allocated rate on i-th path of source s. We assume $x_{s,i} \in [m, M]$ for all $i \in \boldsymbol{s}$ and $s \in \mathcal{N}$.

Although U_s is usually assumed as a strictly concave function with respect to each variable $x_{s,i}$ in the literature, the function $U_s(\sum_{i\in\boldsymbol{s}} x_{s,i})$ remains a non-strictly concave function in terms of vector $\boldsymbol{x}_s = [x_{s,1}, \ldots, x_{s,|\boldsymbol{s}|}]^T$.[1] The function is also non-separable in

* Corresponding author.
E-mail addresses: phuongvo@khu.ac.kr (P.L. Vo), letuanh@tdmu.edu.vn, letuanh@khu.ac.kr (T.A. Le), drsungwon@khu.ac.kr (S. Lee), cshong@khu.ac.kr, cshong@networking.khu.ac.kr (C.S. Hong), bskim25@etri.re.kr (B. Kim), hsong@etri.re.kr (H. Song).

[1] A function $f(\boldsymbol{x})$ is called *(non-)strictly convex* on $\mathbb{D}$ if its Hessian matrix is positive (semi)definite on $\mathbb{D}$, i.e., $\nabla^2 f(\boldsymbol{x}) \succ 0$, $\forall \boldsymbol{x} \in \mathbb{D}$ in the case of strictly convex or $\nabla^2 f(\boldsymbol{x}) \succeq 0$, $\forall \boldsymbol{x} \in \mathbb{D}$ in the case of non-strictly convex. A convex optimization problem with the cost be a (non-)strictly convex function is called *(non-)strictly convex optimization problem.* In this paper, maximizing the concave objective, the NUM is called *(non-)strictly convex problem* if the objective is a (non-)strictly concave function.

0743-7315/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jpdc.2013.07.005

which the variables $x_{s,i}$, $\forall i \in \boldsymbol{s}$ are regulated by only one function. In many current works on canonical multi-path NUM [8,5,19,9,6], the authors performed either subtraction or addition of a strictly convex or concave function from or to the objective in order to transform the multi-path NUM into a strictly convex problem. This new problem is distributively solved either by primal or dual approach. The results are globally optimal solutions. However, the new strictly convex problems remain non-separable. As such, they do not fully model the case of a multi-path user having paths with different characteristics, namely, different round-trip-times (RTT). On the other hand, current TCPs are window-based update protocols, whereas the algorithms in [8,5,19,9,6] are rate-based updates. Hence, deploying them to current Internet is challenging. Authors in [20,21] recently showed that the transport protocol derived from the canonical multi-path NUM has flappiness in the subflows in packet-level simulation caused by the multiple solutions of the non-strictly convex optimization problem.

In this paper, we propose a novel approach to overcome the above issues. We introduce a modified strictly concave utility function for the multi-path user. The multi-path NUM is solved afterward using the successive approximation method. The sequence of solutions to approximation problems generated the algorithm converges to the unique global optimal solution of the multi-path NUM. Each approximation problem is solved for a solution using dual-based gradient approach [11].

In going beyond the previous works, we establish a connection between the theoretical model and the practical design of multi-path TCPs. Utilizing the separability of the approximation problem and the dual-based approach of the algorithm, a series of multi-path TCPs is designed based on the reverse engineering frameworks of the current TCPs [10]. The proposed multi-path TCPs are completely compatible with current TCPs. Running our proposed multi-path TCP in a single-path environment is exactly like running a regular single-path TCP.

The successive approximation approach is introduced in [13], and is usually applied to geometric programming in power control problems, such as in [3,15,17], to approximate non-convex capacity constraints. In this paper, we utilize this method by approximating the non-separable objective into a new strictly concave and separable one. Our previous work also utilizes this method to approximate both objective and power constraints of NUM to jointly control the power and rate in multi-hop wireless networks supporting multi-class traffic [18].

The paper is organized as follows. Section 2 introduces the multi-path NUM model, and applies the successive approximation approach to solve the problem. In Section 3, we design multi-path Vegas, multi-path Reno, and general form of multi-path TCP based on the theoretical analysis. Multi-path Reno experiments and conclusions are presented in Sections 4 and 5, respectively.

Notations: in the paper, we use italic characters to denote variables and bold characters to denote vectors. For example, $\boldsymbol{x}_s \triangleq [x_{s,1}, x_{s,2}, \ldots, x_{s,|\boldsymbol{s}|}]^T$ is the rate vector of all paths from source s, and $\boldsymbol{x} \triangleq [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_{|\mathcal{N}|}^T]^T$ is the rate vector of all paths from all sources. Similarly, $\theta_{s,i}$ is the auxiliary variable associated with path i of source s, $\boldsymbol{\theta}_s \triangleq [\theta_{s,1}, \theta_{s,2}, \ldots, \theta_{s,|\boldsymbol{s}|}]^T$, and $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \ldots, \boldsymbol{\theta}_{|\mathcal{N}|}^T]^T$.

2. Analysis

2.1. Multi-path NUM model

We consider the following multi-path NUM problem in this paper:

Main Problem:

$$\begin{aligned} \text{Max.}\quad & (1-\epsilon)\sum_{s\in\mathcal{N}} U_s\Big(\sum_{i\in\boldsymbol{s}} x_{s,i}\Big) + \epsilon \sum_{s\in\mathcal{N}}\sum_{i\in\boldsymbol{s}} U_s(x_{s,i}) \\ \text{s.t.}\quad & \sum_{s\in\mathcal{N}}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} \le c_l, \quad \forall l \in \mathcal{L}, \end{aligned}$$

where ϵ is a small strictly positive number, and $\epsilon < 1$. In the Main Problem, each source s is associated with a new utility function $V_s(\boldsymbol{x}_s) = (1-\epsilon)U_s\big(\sum_{i\in\boldsymbol{s}} x_{s,i}\big) + \epsilon\sum_{i\in\boldsymbol{s}} U_s(x_{s,i})$. This utility is exactly the single-path utility if the source has only one subflow. Hence, this model is compatible with canonical NUM for single-path network environment. The multi-path utility function $V_s(\boldsymbol{x}_s)$ has two terms, namely, non-separable and separable terms. When $\epsilon = 0$, the multi-path NUM is completely non-separable, and is the canonical multi-path NUM addressed in [19,9,6]. Meanwhile, when $\epsilon = 1$, the multi-path NUM is separable. Each subflow is treated as a separate flow with its own utility function. The former is also called *fully coupled* model, whereas the later is called *uncoupled* model [20,21,7].

The following conditions are assumed in this paper:

A1 (utility function): on the interval $[m, M] \subset \mathbb{R}$, the function $U_s(x)$ is increasing, strictly concave, and continuously differentiable. The curvature of $U_s(x)$ is bounded away from zero on $[m, M]$, i.e., $-U_s''(x) \ge 1/\alpha_s > 0 \forall x \in [m, M]$ for all $s \in \mathcal{N}$.

A2 (Slater's condition): there exists a strictly feasible point $\boldsymbol{x}$, i.e., $\sum_{s\in\mathcal{N}}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} < c_l$, $\forall l \in \mathcal{L}$ and $x_{s,i} \in [m, M]$, $\forall i \in \boldsymbol{s}$, $s \in \mathcal{N}$.[2]

2.2. Approximation problem

It has been known that if $f(.)$ is a concave function, Jensen's inequality, $f(\sum_{i\in\boldsymbol{s}} \theta_{s,i} z_i) \ge \sum_{i\in\boldsymbol{s}} \theta_{s,i} f(z_i)$, holds for all $\boldsymbol{\theta}_s \succ \boldsymbol{0}$ and $\boldsymbol{1}^T\boldsymbol{\theta}_s = 1$.[3] After replacing $x_{s,i} = \theta_{s,i} z_i$, the following inequality is obtained

$$U_s\Big(\sum_{i\in\boldsymbol{s}} x_{s,i}\Big) \ge \sum_{i\in\boldsymbol{s}} \theta_{s,i} U_s\Big(\frac{x_{s,i}}{\theta_{s,i}}\Big). \tag{1}$$

The equality of (1) holds if and only if

$$\theta_{s,i} = \frac{x_{s,i}}{\sum_{j\in\boldsymbol{s}} x_{s,j}}, \quad \forall i \in \boldsymbol{s}, s \in \mathcal{N}. \tag{2}$$

By denoting

$$\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) \triangleq (1-\epsilon)\theta_{s,i} U_s\Big(\frac{x_{s,i}}{\theta_{s,i}}\Big) + \epsilon U_s(x_{s,i}), \quad \forall s \in \mathcal{N}, \tag{3}$$

as a function of $x_{s,i}$ parameterized by $\theta_{s,i}$, the approximation of Main Problem is as follows

Approximation Problem:

$$\begin{aligned} \text{Max.}\quad & \sum_{s\in\mathcal{N}}\sum_{i\in\boldsymbol{s}} \tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) \\ \text{s.t.}\quad & \sum_{s\in\mathcal{N}}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} \le c_l, \quad \forall l \in \mathcal{L}. \end{aligned}$$

Approximation Problem is exactly the basic NUM problem in which a new separable and strictly concave utility is associated with each subflow. Therefore, given $\boldsymbol{\theta}$, the network treats each subflow as a separate flow. Approximation Problem is solved by the dual-based decomposition method as described in [11].

The dual function is given by

$$\begin{aligned} D(\boldsymbol{\lambda}) &= \max_{\boldsymbol{x}}\Big(\sum_{s\in\mathcal{N}}\sum_{i\in\boldsymbol{s}} \tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) - \sum_{l\in\mathcal{L}} \lambda_l\Big(\sum_{s\in\mathcal{N}}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} - c_l\Big)\Big) \\ &= \sum_{s\in\mathcal{N}}\sum_{i\in\boldsymbol{s}} \max_{x_{s,i}}\Big(\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) - \Big(\sum_{l\in\mathcal{R}_{s,i}} \lambda_l\Big)x_{s,i}\Big) + \sum_{l\in\mathcal{L}} c_l\lambda_l, \end{aligned}$$

and the dual problem is $\min_{\boldsymbol{\lambda}\succeq 0} D(\boldsymbol{\lambda})$.

[2] Since Main Problem only has affine constraints, this condition can be weakened: there exists a feasible point of Main Problem (see [2, page 226]).

[3] For vectors, the notations $\succ$, $\succeq$, $\prec$, $\preceq$ mean component-wise larger than, larger than or equal, less than, less than or equal, respectively.

Let $q_{s,i}(t) \triangleq \sum_{l \in \mathcal{R}_{s,i}} \lambda_l(t)$ be the congestion price of i-th path from source s. Solving the subproblem $\max_{x_{s,i}} \Big(\tilde{U}_{s,i}(x_{s,i}; \theta_{s,i}) - (\sum_{l \in \mathcal{R}_{s,i}} \lambda_l) x_{s,i} \Big)$ yields the rate update for each subflow

$$x_{s,i}(t+1) = \left[\tilde{U}_{s,i}^{'-1}(q_{s,i}(t); \theta_{s,i}) \right]_m^M, \tag{4}$$

where $[a]_c^b = \max(\min(a,b), c)$.

The congestion price update of each link is obtained by applying the projected gradient descent algorithm for the dual problem:

$$\lambda_l(t+1) = \left[\lambda_l(t) + \kappa \left(\sum_{s \in \mathcal{N}} \sum_{i: l \in \mathcal{R}_{s,i}} x_{s,i}(t) - c_l \right) \right]^+, \quad \forall l \in \mathcal{L}, \tag{5}$$

where step-size κ is sufficiently small for the convergence of the algorithm, and $[a]^+ = \max(a, 0)$.

2.3. Successive approximation algorithm

Algorithm 1. Initialize from any feasible point, in the τ-th iteration,

1. Each source updates $\boldsymbol{\theta}_s$ according to (2) with $\boldsymbol{x}_s^\infty(\tau-1)$;
2. Given $\boldsymbol{\theta}_s(\tau)$, source s updates the transmission rates on its paths according to (4), and the links update their prices using (5) until converging to the stationary point $\boldsymbol{x}^\infty(\tau)$;
3. Increase τ and go back to step 1.

Let define the parameter $\bar{\alpha} \triangleq \max_{s \in \mathcal{N}} \alpha_s$, the length of the longest path used by flows/subflows $\bar{L} \triangleq \max_{i \in \boldsymbol{s}, s \in \mathcal{N}} |\mathcal{R}_{s,i}|$, and the most congested link used by flows/subflows $\bar{S} \triangleq \max_{l \in \mathcal{L}} |\mathcal{N}(l)|$. We have the following theorem.

Theorem 1. *Suppose that* $0 < \epsilon < 1$. *If the step-size satisfies* $0 < \kappa < \frac{2\epsilon}{\bar{\alpha}\bar{L}\bar{S}}$, *then the sequence* $\{\boldsymbol{x}^\infty(\tau)\}$, $\tau = 1, 2, \ldots$ *generated by Algorithm 1 converges to an accumulation point, which is the unique optimal solution to Main Problem.*

Proof. The following parameters are defined for a convenient presentation:

- $\boldsymbol{x}^o(\tau)$: initial point of step τ;
- $\boldsymbol{x}^\infty(\tau)$: stationary point of step τ;
- $G(\boldsymbol{x}) \triangleq (1-\epsilon) \sum_{s \in \mathcal{N}} U_s\big(\sum_{i \in \boldsymbol{s}} x_{s,i}\big) + \epsilon \sum_{s \in \mathcal{N}} \sum_{i \in \boldsymbol{s}} U_s(x_{s,i})$: objective of Main Problem;
- $\tilde{G}(\boldsymbol{x}; \boldsymbol{\theta}) \triangleq \sum_{s \in \mathcal{N}} \sum_{i \in \boldsymbol{s}} \tilde{U}_{s,i}\big(x_{s,i}; \theta_{s,i}\big)$: objective of the approximation problem given $\boldsymbol{\theta}$.

We apply Zangwill's Global Convergence Theorem [22] on the convergence proof. Function $G(\boldsymbol{x})$ is used as the ascent function, and the solution set is then defined as the set of the optimal solutions of Main Problem.

The solution to Approximation Problem indeed increases the objective of Main Problem per iteration due to the following equality and inequalities:

$$\begin{aligned} G(\boldsymbol{x}^\infty(\tau-1)) &= \tilde{G}(\boldsymbol{x}^o(\tau); \boldsymbol{\theta}(\tau)) && (6) \\ &\le \tilde{G}(\boldsymbol{x}^\infty(\tau); \boldsymbol{\theta}(\tau)) && (7) \\ &\le G(\boldsymbol{x}^\infty(\tau)). && (8) \end{aligned}$$

Equality (6) is obtained from replacing $\theta_{s,i}(\tau)$ by $\frac{x_{s,i}^\infty(\tau-1)}{\sum_{j \in \boldsymbol{s}} x_{s,i}^\infty(\tau-1)}$ and $\boldsymbol{x}^o(\tau)$ by $\boldsymbol{x}^\infty(\tau-1)$. The assumptions A1-2 guarantee that the approximation utility $\tilde{U}_{s,i}(x_{s,i})$ satisfies conditions C1-2 in [11], particularly, $-\tilde{U}''_{s,i}(x) \ge \epsilon/\alpha_s > 0$ for all $x \in [m, M]$ and $s \in \mathcal{N}$. As a result, the dual-based algorithm converges to the optimal value of the primal problem, i.e., the approximation problem. Hence, (7) is always satisfied since $\boldsymbol{x}^\infty(\tau)$ is the optimal solution of Approximation Problem given $\boldsymbol{\theta}(\tau)$. Finally, inequality (8) is satisfied because of (1). Therefore, the sequence $\{G(\boldsymbol{x}^\infty(\tau))\}$, $\tau = 0, 1, \ldots$ increases in every iteration.

The convergence conditions according to Zangwill's Global Convergence Theorem are now verified point-by-point:

1. the feasible regions of Approximation Problems and Main Problem are identical and are the compact sets.
2. if $\boldsymbol{x}^\infty(\tau-1)$ is not a solution to Main Problem, we prove that $G(\boldsymbol{x}^\infty(\tau-1)) < G(\boldsymbol{x}^\infty(\tau))$ by contradiction. If $G(\boldsymbol{x}^\infty(\tau-1)) < G(\boldsymbol{x}^\infty(\tau))$ is not true, then $G(\boldsymbol{x}^\infty(\tau-1)) = G(\boldsymbol{x}^\infty(\tau))$ as we always have $G(\boldsymbol{x}^\infty(\tau-1)) \le G(\boldsymbol{x}^\infty(\tau))$ according to the equality/inequalities (6)–(8). The equality of Jensen's inequality (8) holds. Thus, $\theta_{s,i}(\tau)$ which is assigned by $\frac{x_{s,i}^\infty(\tau-1)}{\sum_{j \in \boldsymbol{s}} x_{s,j}^\infty(\tau-1)}$ at the beginning of the iteration is equal to $\frac{x_{s,i}^\infty(\tau)}{\sum_{j \in \boldsymbol{s}} x_{s,j}^\infty(\tau)}$, i.e., $\theta_{s,i}(\tau) = \frac{x_{s,i}^\infty(\tau)}{\sum_{j \in \boldsymbol{s}} x_{s,j}^\infty(\tau)}$ for all $s \in \mathcal{N}$, $i \in \boldsymbol{s}$.
   On the other hand, $\boldsymbol{x}^\infty(\tau)$ and the corresponding dual value $\boldsymbol{\lambda}^\infty(\tau)$ are the primal–dual optimal solutions to Approximation Problem according to [11]. Thus, they satisfy the Karush–Kuhn–Tucker (KKT) conditions of Approximation Problem:

$$\nabla \tilde{G}(\mathbf{x}^\infty(\tau); \boldsymbol{\theta}(\tau)) - \boldsymbol{q}^\infty(\tau) = 0, \tag{9}$$

$$\lambda_l^\infty(\tau) \left(\sum_{s \in \mathcal{N}} \sum_{i: l \in \mathcal{R}_{s,i}} x_{s,i}^\infty(\tau) - c_l \right) = 0, \quad \forall l \in \mathcal{L}, \tag{10}$$

$$\sum_{s \in \mathcal{N}} \sum_{i: l \in \mathcal{R}_{s,i}} x_{s,i}^\infty(\tau) \le c_l, \quad \forall l \in \mathcal{L}, \tag{11}$$

$$\lambda_l^\infty(\tau) \ge 0, \quad \forall l \in \mathcal{L}, \tag{12}$$

where $q_{s,i}^\infty(\tau) = \sum_{l \in \mathcal{R}_{s,i}} \lambda_l^\infty(\tau)$.

With $\theta_{s,i}(\tau) = \frac{x_{s,i}^\infty(\tau)}{\sum_{j \in \boldsymbol{s}} x_{s,j}^\infty(\tau)}$ for all $s \in \mathcal{N}$ and $i \in \boldsymbol{s}$, we prove that $\boldsymbol{x}^\infty(\tau)$ is a global optimal solution to Main Problem. The following equalities are obtained

$$\begin{aligned} \left. \frac{\partial U_s\left(\sum_{i \in \boldsymbol{s}} x_{s,i}\right)}{\partial x_{s,1}} \right|_{\boldsymbol{x}_s = \boldsymbol{x}_s^\infty(\tau)} &= \left. \frac{\partial U_s\left(x_{s,1} + \sum_{\substack{i \in \boldsymbol{s} \\ i \neq 1}} x_{s,i}^\infty(\tau)\right)}{\partial\left(x_{s,1} + \sum_{\substack{i \in \boldsymbol{s} \\ i \neq 1}} x_{s,i}^\infty(\tau)\right)} \right|_{x_{s,1} = x_{s,1}^\infty(\tau)} \\ &= \left. \frac{\partial U_s(x)}{\partial x} \right|_{x = \sum_{i \in \boldsymbol{s}} x_{s,i}^\infty(\tau) = \frac{x_{s,1}^\infty(\tau)}{\theta_{s,1}(\tau)}} \\ &= \left. \frac{\partial U_s\left(\frac{x_{s,1}}{\theta_{s,1}(\tau)}\right)}{\partial\left(\frac{x_{s,1}}{\theta_{s,1}(\tau)}\right)} \right|_{x_{s,1} = x_{s,1}^\infty(\tau)} \\ &= \theta_{s,1}(\tau) \left. \frac{\partial U_s\left(\frac{x_{s,1}}{\theta_{s,1}(\tau)}\right)}{\partial x_{s,1}} \right|_{x_{s,1} = x_{s,1}^\infty(\tau)}. \end{aligned}$$

Therefore,

$$\begin{aligned} \left. \frac{\partial G(\boldsymbol{x})}{\partial x_{s,1}} \right|_{\boldsymbol{x} = \boldsymbol{x}^\infty(\tau)} = (1-\epsilon) &\left. \frac{\partial U_s\left(\sum_{i \in \boldsymbol{s}} x_{s,i}\right)}{\partial x_{s,1}} \right|_{\boldsymbol{x}_s = \boldsymbol{x}_s^\infty(\tau)} \\ &+ \epsilon \left. \frac{\partial U_s(x_{s,1})}{\partial x_{s,1}} \right|_{x_{s,1} = x_{s,1}^\infty(\tau)} \end{aligned}$$

$$= (1-\epsilon)\theta_{s,1}(\tau)\left.\frac{\partial U_s\left(\frac{x_{s,1}}{\theta_{s,1}(\tau)}\right)}{\partial x_{s,1}}\right|_{x_{s,1}=x^{\infty}_{s,1}(\tau)} + \epsilon \left.\frac{\partial U_s(x_{s,1})}{\partial x_{s,1}}\right|_{x_{s,1}=x^{\infty}_{s,1}(\tau)} = \left.\frac{\partial \tilde{G}(\boldsymbol{x};\boldsymbol{\theta}(\tau))}{\partial x_{s,1}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{\infty}(\tau)}.$$

Proofs of all the other partial differential equations are similar. Hence, $\nabla\tilde{G}(\boldsymbol{x}^{\infty}(\tau);\boldsymbol{\theta}(\tau)) - \mathbf{q}^{\infty}(\tau) = \nabla G(\boldsymbol{x}^{\infty}(\tau)) - \mathbf{q}^{\infty}(\tau) = 0$. The remaining conditions (10)–(12) are the same. Combined with Slater's assumption, the KKT conditions are also sufficient for $\boldsymbol{x}^{\infty}(\tau)$ to be a global optimal solution to Main Problem [2]. Since $G(\boldsymbol{x}^{\infty}(\tau-1)) = G(\boldsymbol{x}^{\infty}(\tau))$, $\boldsymbol{x}^{\infty}(\tau-1)$ is also a solution to Main Problem, which is a contradiction.

3. If $\boldsymbol{x}^{\infty}(\tau-1)$ is a solution, then $G(\boldsymbol{x}^{\infty}(\tau-1)) \le G(\boldsymbol{x}^{\infty}(\tau))$.
4. Each iteration in Algorithm 1 is a closed point-to-set mapping [4].

According to Zangwill's convergence theorem [22], any accumulation point of the sequence $\{\boldsymbol{x}^{\infty}(\tau)\}$, $\tau = 1, 2, \ldots$ is a solution of Main Problem (recall that the solution set is defined as the set of solutions to Main Problem).

Main Problem is a strictly convex optimization problem as $\epsilon > 0$. The optimal point is unique [1]. Since the sequence $\{\boldsymbol{x}^{\infty}(\tau)\}$, $\tau = 1, 2, \ldots$ is bounded and has a unique accumulation point, it converges to the unique optimal solution of Main Problem [12]. □

Notice that Algorithm 1 has two levels of convergence. The outer-iterations update $\boldsymbol{\theta}$ and the inner-iterations are indeed the standard dual-based algorithm solving the NUM with a new utility function. Although only local information is required to update $\boldsymbol{\theta}_s$ for each source, the network theoretically needs to be stable before updating $\boldsymbol{\theta}$ which is the information from all the other sources.

Recognizing network stationary in each outer-iteration is difficult in the practical implementation, especially in time-varying environment. We apply a heuristic algorithm by fixing the number of inner-iterations in every outer-iteration, i.e., 50 inner-iterations in the experiments. The dual-based gradient algorithm monotonically decreases the dual function in every step given a sufficiently small step-size (see [11]). However, the monotonic increase of the objective of the primal problem is not guaranteed. Hence, the heuristic algorithm does not theoretically guarantee an increase in aggregate utility in every outer-step. Nevertheless, the heuristic algorithm still converges even with a small number of inner-iterations in many experiments.

3. Multi-path TCPs

This section discusses the design of multi-path TCPs (MTCPs) based on the theoretical analysis in Section 2. We focus on three main targets in designing MTCPs:

1. MTCPs must be compatible with current single-path TCPs, i.e., they perform exactly the same as the corresponding single-path TCPs if they are used in a single-path environment.
2. the model must address the mismatch parameters between paths from one source, such as different backlog packets in Vegas or different round-trip-times (RTT) in Reno.
3. protocols can be implemented online.

It has been known that current single-path TCPs are implicit solutions to the NUM problems with particular utility functions [10]. For example, TCP Vegas has utility $U_s(x_s) = \alpha_s d_s \log(x_s)$, where d_s is the propagation delay of the path associated with source s, and $\alpha_s d_s$ is the number of backlog packets on the path; TCP Reno has the utility $U_s(x_s) = -\frac{3/2}{D_s^2 x_s}$, where D_s is the RTT of the path. Such concave functions satisfy assumptions A1-2, rendering the framework in Section 2 applicable.

However, applying the framework raises two questions. *How to choose the appropriate utility for the multi-path user* is the first question. In the single-path case, each utility function of a single-path flow has path-specific parameters. For example, the single-path utility of TCP Reno $U_s(x_s) = -\frac{3/2}{D_s^2 x_s}$ uses the RTT of the path, D_s. This parameter also serves as the weight in the sum of utilities in the maximization problem. In TCP Reno, the less bandwidth is allocated for the flow with higher RTT. As the utility for the multi-path user is given by $V_s(\boldsymbol{x}_s) = (1-\epsilon)U_s\left(\sum_{i\in\boldsymbol{s}} x_{s,i}\right) + \epsilon\sum_{i\in\boldsymbol{s}} U_s(x_{s,i})$, there are two terms in the multi-path utility, namely, the non-separable and separable terms. Choosing the second term $U_s(x_{s,i})$ as the corresponding utility function with the parameter specifying subflow (s,i) is straightforward. However, the non-separable term is quite complex. To be compatible with the single-path environment, the multi-path utility must become exactly the single-path utility when the user has only one subflow. For example, in TCP Reno, $-\frac{3/2}{D_s^{\min 2}\sum_{i\in\boldsymbol{s}} x_{s,i}}$ and $-\frac{3/2}{D_s^{\max 2}\sum_{i\in\boldsymbol{s}} x_{s,i}}$ where $D_s^{\min 2} = \min_{i\in\boldsymbol{s}} D_{s,i}$ and $D_s^{\max 2} = \max_{i\in\boldsymbol{s}} D_{s,i}$ are some candidates for the non-separable term of the multi-path utility.

The second question is *how to deploy the appropriate approximation inequality*. As the utility of each subflow in Approximation Problem is $\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) = (1-\epsilon)\theta_{s,i}U_s\left(\frac{x_{s,i}}{\theta_{s,i}}\right) + \epsilon U_s(x_{s,i})$, the function $\tilde{U}_{s,i}(x_{s,i};\theta_{s,i})$ should also have the format specified for the path (s,i) to address target 2. The equilibrium equation should be similar to that of the single-path TCP to easily forward engineering based on the current framework for single-path TCPs. On the other hand, to satisfy target 1, $\tilde{U}_{s,i}(x_{s,i};\theta_{s,i})$ must also become exactly the canonical single-path utility function in the single-path case. Hence, if we apply Jensen's inequality (1) directly, such as using the function $-\frac{3/2}{D_s^{\min 2}\sum_{i\in\boldsymbol{s}} x_{s,i}}$ for the non-separable term of the multi-path utility in Reno, then the approximation utility $\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) = -(1-\epsilon)\theta_{s,i}\frac{3/2}{D_s^{\min 2} x_{s,i}} - \epsilon\frac{3/2}{D_{s,i}^2 x_{s,i}}$ is difficult to use to forward engineering. We will use the parameter $\hat{\boldsymbol{\theta}}$ instead of $\boldsymbol{\theta}$ in the deployments.

The number of inner-iterations is fixed in N iterations for online implementation. $\hat{\boldsymbol{\theta}}$ is updated after each N iterations. The following subsections are three examples of deploying our theoretical framework to design three new multi-path TCPs, which are compatible with their corresponding TCPs, i.e., Vegas, Reno, and general Reno.

3.1. Multi-path Vegas (mVegas)

Single-path TCP Vegas has the implicit utility function $U_s(x_s) = \alpha_s d_s \log(x_s)$, where d_s is the propagation delay of the path associated with source s, and $\alpha_s d_s$ is the number of backlog packets on the path. Denote $b_s \triangleq \alpha_s d_s$. We choose the utility function for the multi-path users as follows

$$V_s(\boldsymbol{x}_s) = (1-\epsilon)b_s^{\min}\log\left(\sum_{i\in\boldsymbol{s}} x_{s,i}\right) + \epsilon\sum_{i\in\boldsymbol{s}} b_{s,i}\log(x_{s,i}),$$

where $b_s^{\min} = \min_i\{\alpha_{s,i}d_{s,i}\}$.

The approximation inequality is modified

$$b_s^{\min}\log\left(\sum_{i\in\boldsymbol{s}} x_{s,i}\right) \ge b_s^{\min}\sum_{i\in\boldsymbol{s}}\theta_{s,i}\log\left(\frac{x_{s,i}}{\theta_{s,i}}\right) = \sum_{i\in\boldsymbol{s}} b_{s,i}\hat{\theta}_{s,i}\log\left(\frac{b_s^{\min}}{b_{s,i}\hat{\theta}_{s,i}}x_{s,i}\right),$$

where $b_{s,i} = \alpha_{s,i} d_{s,i}$ is the number of backlog packets on ith path of source s. The equality holds as

$$\hat{\theta}_{s,i} \triangleq \frac{b_s^{\min}}{b_{s,i}}\theta_{s,i} = \frac{b_s^{\min}}{b_{s,i}}\frac{x_{s,i}}{\sum\limits_{i\in\boldsymbol{s}} x_{s,i}}. \tag{13}$$

Hence,

$$\begin{aligned}\tilde{U}_{s,i}(x_{s,i};\hat{\theta}_{s,i}) &= ((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)b_{s,i}\log(x_{s,i}) \\ &\quad + (1-\epsilon)\hat{\theta}_{s,i}b_{s,i}\log\left(\frac{b_s^{\min}}{b_{s,i}\hat{\theta}_{s,i}}\right).\end{aligned}$$

Note that if mVegas has the same number of backlog packets on subflows from one source, then we still have $\hat{\theta}_{s,i} = \theta_{s,i} = \frac{x_{s,i}}{\sum_{i\in\boldsymbol{s}} x_{s,i}}$ as normal.

From (4), we have

$$x_{s,i} = ((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\frac{b_{s,i}}{q_{s,i}} = ((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\frac{\alpha_{s,i}d_{s,i}}{q_{s,i}}$$

at the equilibrium point. As a result, the window-size and the rate updates of multi-path Vegas in one time-slot are as follows

$$w_{s,i}(t+1) = \begin{cases} w_{s,i}(t) + \dfrac{1}{D_{s,i}} & \text{if } \dfrac{w_{s,i}(t)}{d_{s,i}} - \dfrac{w_{s,i}(t)}{D_{s,i}} \\ \qquad < ((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\alpha_{s,i}, & \\ w_{s,i}(t) - \dfrac{1}{D_{s,i}} & \text{if } \dfrac{w_{s,i}(t)}{d_{s,i}} - \dfrac{w_{s,i}(t)}{D_{s,i}} \\ \qquad > ((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\alpha_{s,i}, & \\ w_{s,i}(t) \quad \text{otherwise}, & \end{cases} \tag{14}$$

$$\begin{aligned} x_{s,i}(t+1) &= \Bigg[x_{s,i}(t) + \frac{1}{D_{s,i}^2} \\ &\quad \times \mathbf{1}\left(((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\frac{\alpha_{s,i}d_{s,i}}{q_{s,i}(t)} - x_{s,i}(t)\right)\Bigg]^+, \end{aligned} \tag{15}$$

where $\mathbf{1}(z)$ equals 1 if $z > 0$, -1 if $z < 0$, and 0 if $z = 0$. $D_{s,i}$ is the RTT, $d_{s,i}$ is the total propagation delay, and $q_{s,i}$ is the queueing delay of path i. We can see a slight difference between the updates of the subflow in mVegas and that in the single-path TCP Vegas. In each RTT, mVegas adjusts the window-size by 1 when comparing the throughput with $((1-\epsilon)\hat{\theta}_{s,i}+\epsilon)\alpha_{s,i}$ instead of α_s as in the single-path Vegas. $\hat{\theta}$ is always 1 for a single-path user, hence, (15) and (14) become exactly the rate and window-size updates of (single-path) TCP Vegas, respectively. Therefore, mVegas can coexist with the current Vegas. The link prices, i.e., the queue sizes, are implicitly updated. The total congestion price of the entire path is feed-backed to the source in acknowledgment packets.

3.2. Multi-path Reno (mReno)

With a small loss rate, the implicit utility function of TCP Reno for single-path user is given by $U_s(x_s) = -\frac{3/2}{D_s^2 x_s}$, where D_s is the RTT of the path. The utility function for the multi-path user is utilized as follows.

$$V_s(\boldsymbol{x}_s) = -(1-\epsilon)\frac{3/2}{D_s^{\min 2}\left(\sum\limits_{i\in\boldsymbol{s}} x_{s,i}\right)} - \epsilon\sum_{i\in\boldsymbol{s}}\frac{3/2}{D_{s,i}^2 x_{s,i}},$$

where $D_s^{\min}$ is the minimum RTT over all paths from source s. In order for $\tilde{U}(.)$ to have the similar form as the single-path utility, $\theta_{s,i}$ is selected such that

$$\theta_{s,i} = \frac{x_{s,i}}{\sum\limits_{i\in\boldsymbol{s}} x_{s,i}} = \frac{D_s^{\min}}{D_{s,i}}\hat{\theta}_{s,i}$$

or

$$\hat{\theta}_{s,i} = \frac{D_{s,i}}{D_s^{\min}}\frac{x_{s,i}}{\sum\limits_{i\in\boldsymbol{s}} x_{s,i}}. \tag{16}$$

The approximation inequality becomes

$$\begin{aligned} -\frac{3/2}{D_s^{\min 2}\left(\sum\limits_{i\in\boldsymbol{s}} x_{s,i}\right)} &\geq -\sum_{i\in\boldsymbol{s}}\theta_{s,i}\frac{3/2}{D_s^{\min 2}\left(\frac{x_{s,i}}{\theta_{s,i}}\right)} \\ &= -\sum_{i\in\boldsymbol{s}}\hat{\theta}_{s,i}^2\frac{3/2}{D_{s,i}^2 x_{s,i}}. \end{aligned}$$

Hence,

$$\tilde{U}_{s,i}(x_{s,i};\hat{\theta}_{s,i}) = -((1-\epsilon)\hat{\theta}_{s,i}^2+\epsilon)\frac{3/2}{D_{s,i}^2 x_{s,i}}.$$

The equation $q_{s,i} = ((1-\epsilon)\hat{\theta}_{s,i}^2+\epsilon)\frac{3/2}{D_{s,i}^2 x_{s,i}^2}$ is obtained at the equilibrium point. Thus, we construct the rate update of mReno in one time-slot as follows

$$\begin{aligned} &x_{s,i}(t+1) \\ &= \left[x_{s,i}(t) + ((1-\epsilon)\hat{\theta}_{s,i}^2+\epsilon)\frac{1}{D_{s,i}^2} - \frac{2}{3}q_{s,i}(t)x_{s,i}^2(t)\right]^+. \end{aligned} \tag{17}$$

Since $x = \frac{w}{D}$, the window-size update in 1 time-slot is given by

$$((1-\epsilon)\hat{\theta}_{s,i}^2+\epsilon)\frac{1}{D_{s,i}(t)} - \frac{2}{3}q_{s,i}(t)x_{s,i}^2(t). \tag{18}$$

Finally, mReno operates as follows: *the window-size of each subflow increases by* $((1-\epsilon)\hat{\theta}_{s,i}^2+\epsilon)$ *in every RTT and decreases by half whenever a packet loss is detected.*[4]

Remark 1. In the literature, the function $\frac{\sqrt{3/2}}{D_s}\tan^{-1}\left(\sqrt{\frac{2}{3}}D_s x_s\right)$ is also used as the implicit utility for TCP Reno. Therefore, the rate update has the form

$$x_s(t+1) = \left[x_s(t) + \frac{1-q_s(t)}{D_s^2} - \frac{2}{3}q_s(t)x_s^2(t)\right]^+.$$

With small loss-rate assumption, $1 - q_s \approx 1$. Hence,

$$x_s(t+1) = \left[x_s(t) + \frac{1}{D_s^2} - \frac{2}{3}q_s(t)x_s^2(t)\right]^+,$$

which leads to the utility $-\frac{3/2}{D_s^2 x_s}$. (With a small loss rate, $\frac{2x_s^2 D_s^2}{3} \gg 1$, the equation $q_s = \frac{3}{2D_s^2 x_s^2+3}$ is approximated to $q_s = \frac{3}{2D_s^2 x_s^2}$ at the equilibrium point, which also leads to the utility function $-\frac{3/2}{D_s^2 x_s}$.) There is a subtle difference between the increase parts of the above two updates. The protocol is implemented "in every RTT, the window-size increases by 1 if the sender receives an ACK" in the former, whereas "the window-size increases by 1 in every RTT" with the latter. Two schemes obtain similar results under the small packet-loss assumption, and both are used commonly [10,20,21].

3.3. General multi-path TCP (GMTCP)

General TCP algorithm increases the rate by $A_s(x_s(t))$ with each positive acknowledgment, and decreases it by $B_s(x_s(t))$ with each negative acknowledgment. Therefore, the utility function derived from the reverse engineering model is given by

[4] The factor 1/2 is usually replaced by 2/3 when describing the TCP behavior in the mathematical model.

$\int_0^{x_s} \frac{A_s(x)}{A_s(x)+B_s(x)} \mathrm{d}x$ [10]. In the cases of $A_s(x_s) = \frac{a}{x_s^{k+1} D_s^{k+2}}$ and $B_s(x_s) = b x_s^l D_s^{l-1}$, we have Binomial TCP with

$$U_s(x_s) = \int_0^{x_s} \frac{1}{1 + \frac{b}{a}(D_s x)^{k+l+1}} \mathrm{d}x$$

as the utility function. The series of TCP-friendly protocols is the case in which $k + l = 1$ and $\frac{a}{b} = \frac{3}{2}$. For example, with $(k, l) = (0, 1)$, Binomial TCP becomes TCP Reno.

We also construct GMTCP based on the Binomial TCP with an analogous framework used in mReno. The utility function for the multi-path user is

$$V_s(\boldsymbol{x}_s) = \int_0^{\sum_{i \in \boldsymbol{s}} x_{s,i}} \frac{1-\epsilon}{1 + \frac{b}{a}(D_s^{\min} x)^{k+l+1}} \mathrm{d}x + \sum_{i \in \boldsymbol{s}} \int_0^{x_{s,i}} \frac{\epsilon}{1 + \frac{b}{a}(D_{s,i} x)^{k+l+1}} \mathrm{d}x.$$

Using the small loss-rate assumption yields

$$\tilde{U}_{s,i}(x_{s,i}; \theta_{s,i}) = (1-\epsilon) \int_0^{x_{s,i}} \frac{a}{b\left(D_{s,i} \frac{x}{\hat{\theta}_{s,i}}\right)^{k+l+1}} \mathrm{d}x + \epsilon \int_0^{x_{s,i}} \frac{a}{b(D_{s,i} x)^{k+l+1}} \mathrm{d}x, \tag{19}$$

where $\hat{\theta}_{s,i} = \frac{D_{s,i}}{D_s^{\min}} \frac{x_{s,i}}{\sum_{i \in \boldsymbol{s}} x_{s,i}}$. (The first term of (19) is obtained by changing the variable x to $\frac{D_s^{\min} \theta_{s,i}}{D_{s,i}} x$.) Finally, the rate update of GMTCP is given by

$$x_{s,i}(t+1) = \left[x_{s,i}(t) + ((1-\epsilon)\hat{\theta}_{s,i}^{k+l+1} + \epsilon) \frac{a}{D_{s,i}^{k+2} x_{s,i}^k(t)} - b q_{s,i}(t) D_{s,i}^{l-1} x_{s,i}^{l+1}(t) \right]^+. \tag{20}$$

Remark 2. In mReno as well as GMTCP, we can choose $D_s^{\max}$ or D_s^{aver} instead of $D_s^{\min}$. Whichever value we choose, it plays the compatibility role for the multi-path TCP to the single-path flows. Therefore, the single-path TCP does not need to change. Using $D_s^{\min}$ implies that the network will treat the multi-path user as a single-path user on the path with the minimum RTT, which is logical in the multi-path routing where the multi-path user should inherit the good characteristic of its best path.

4. mReno experiments

The experiments verify the theoretical result as well as evaluate the performance of mReno in the packet-level simulation. In theoretical experiments, we observe the convergence of the algorithm, and obtain the optimal solution. The RTT is approximated to be double the propagation delay in the calculation. The constant step-size $\kappa = 10^{-5}$ is used. The algorithm is considered as convergence if the difference between two consecutive values of the aggregate utility is less than an error tolerance, i.e., $|\sum_s V_s(\tau) - \sum_s V_s(\tau - 1)| < 10^{-5}$.

In packet-level experiments, we build our code based on the open source code of IETF's MPTCP [14]. Single-path users use normal TCP Reno whereas multi-path users use mReno. The marking probability scheme RED is used. We choose the number of inner-iterations $N = 50$ in all experiments except that in Section 4.1. The $\boldsymbol{\theta}$ is calculated by averaging the last 20 inner-iterations in each outer-step. All the experiments are run in five times to obtain mean values. Samples are taken every 2 ms.

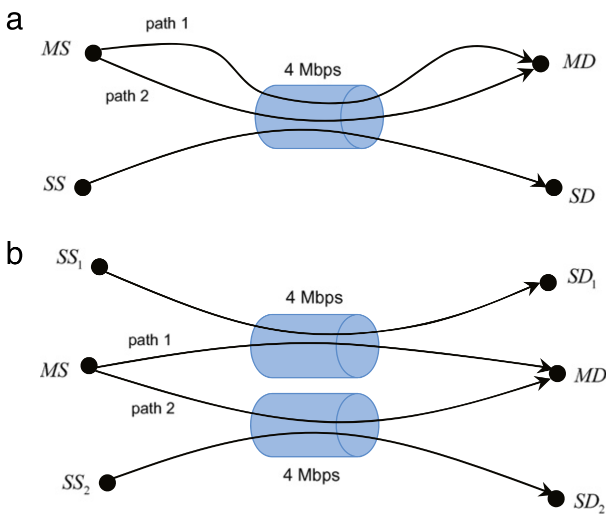

Fig. 1. Network topologies.

4.1. Convergence of the algorithm

In this experiment, we observe the impact of the number of inner-iterations in each outer-step on the convergence time of Algorithm 1. Consider the two-bottleneck-link network as described in Fig. 1(b). The RTTs of two links are 100 and 400 ms.

Fig. 2(a) shows the rapid decrease of the number of outer-iterations to converge when we increase the number of inner-iterations. However, the total number of iterations in all the outer-steps, i.e., the actual iterations, tends to increase as the number of inner-iterations increases as shown in Fig. 2(b). The larger number of inner-iterations results in less calculation but longer convergence time.

4.2. One-bottleneck-link topology

This experiment aimed to demonstrate the effect of parameter ϵ on the convergence of the protocol. Consider a network with two flows (a multi-path flow and a single-path flow) on one bottleneck link (see Fig. 1(a)). The link capacity is 4 Mbps. We assume that the two paths of the multi-path user have the same RTTs. When $\epsilon = 0$, the multi-path and single-path flows are stable and fairly share the link's bandwidth in the packet-level simulation (Fig. 3(b)). However, flappiness occurs in the subflows of the multi-path user as shown in Fig. 3(a). Actually, the flappiness of the multi-path protocol directly derived from the canonical NUM model has been shown in [21,7]. The reason is from the multiple optimal solutions of Main Problem as $\epsilon = 0$. In this case, any subflow rate allocation with a total that equals to 2 Mbps is an optimal solution.

When ϵ is strictly positive, i.e., $\epsilon = 0.05$, Main Problem has a unique solution. The subflow rates are no longer flappy (see Fig. 3(c)). When ϵ increases, Jain's fairness index between two flows decreases.[5] In Fig. 4, a larger ϵ worsens the fairness. When $\epsilon = 1$, the network treats each subflow as an independent flow. The first term of the multi-path utility is zero. Hence, the multi-path flow with two subflows is actually two single-path flows. Its rate doubles the single-path user's rate.

[5] Jain's fairness index is calculated by $f = (\sum_s x_s)^2/(|N| \sum_s x_s^2)$. $f = 1$ indicates the best fairness.

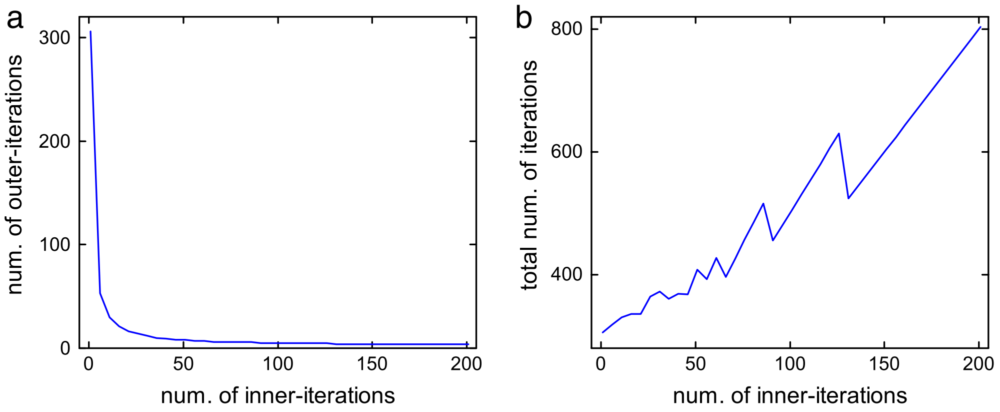

Fig. 2. Number of outer-iterations (a) and total number of iterations (b) until the convergence of Algorithm 1 when the number of inner-iterations in each outer-step increases.

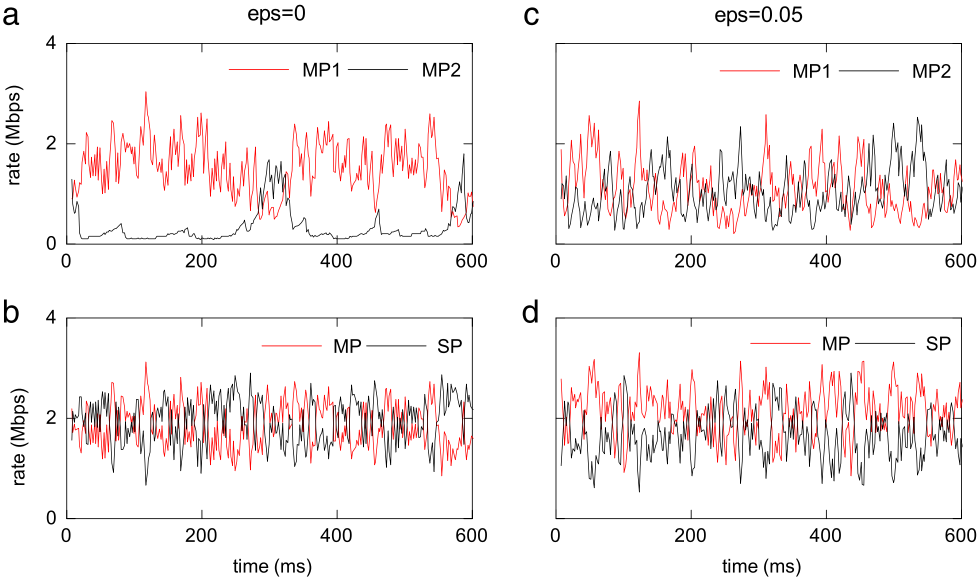

Fig. 3. Rate of flows and subflows in one-bottleneck-link topology. Flows are always stable (b, d), but subflows are flappy when $\epsilon = 0$ (a) and are not flappy when $\epsilon = 0.05$ (c).

4.3. Two-bottleneck-link topology

The two-bottleneck-link topology with three users, namely, one multi-path user and two single-path users, is considered (see Fig. 1(b)). The network is monitored in 1000 s (500 samples). The flows are on/off in the following pattern:

1. 0–250 s: only multi-path user runs,
2. 250–500 s: multi-path user and one single-path user run,
3. 500–750 s: multi-path user and both single-path users run, and
4. 750–1000 s: only multi-path user runs.

Fig. 5 show the rate evolutions of the flows and subflows in four phases. With the absence of the single-path flows in phases 1 and 4, two subflows of the multi-path user take the entire bandwidth of the two links for transmission. In phase 2, the network only has the multi-path user and one single-path user on the shorter-RTT link. Subflow rate on path 1 decreases, and the multi-path user shifts its main traffic to path 2 to reserve the bandwidth on link 1 for single-path user 1 in both same and different RTT cases. In phase 3, when two links have the same RTTs, two subflow rates of the multi-path user are close together, and the two single-path users are too (Fig. 5(a)–(b), Table 1). When two links have different RTTs, the total rate of the multi-path flow is slightly higher than the rate of single-path flow 1 on the shorter-RTT link. However, it is much higher than that of single-path flow 2 on the longer-RTT link (Fig. 5(c)–(d), Table 1). In all of four phases, same RTTs as well as

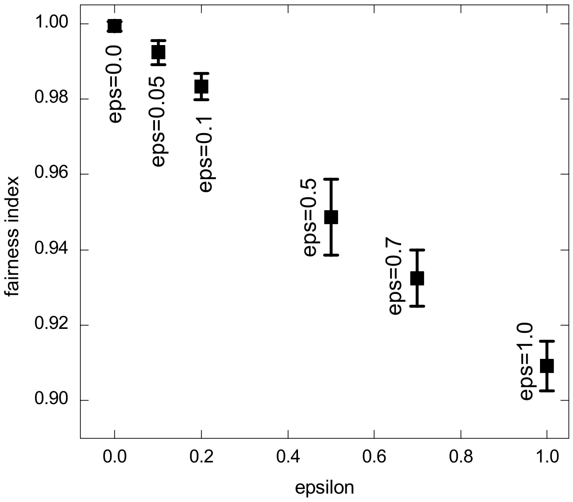

Fig. 4. Fairness index between the multi-path and single-path user rates in one-bottleneck-link topology when ϵ is gradually increased.

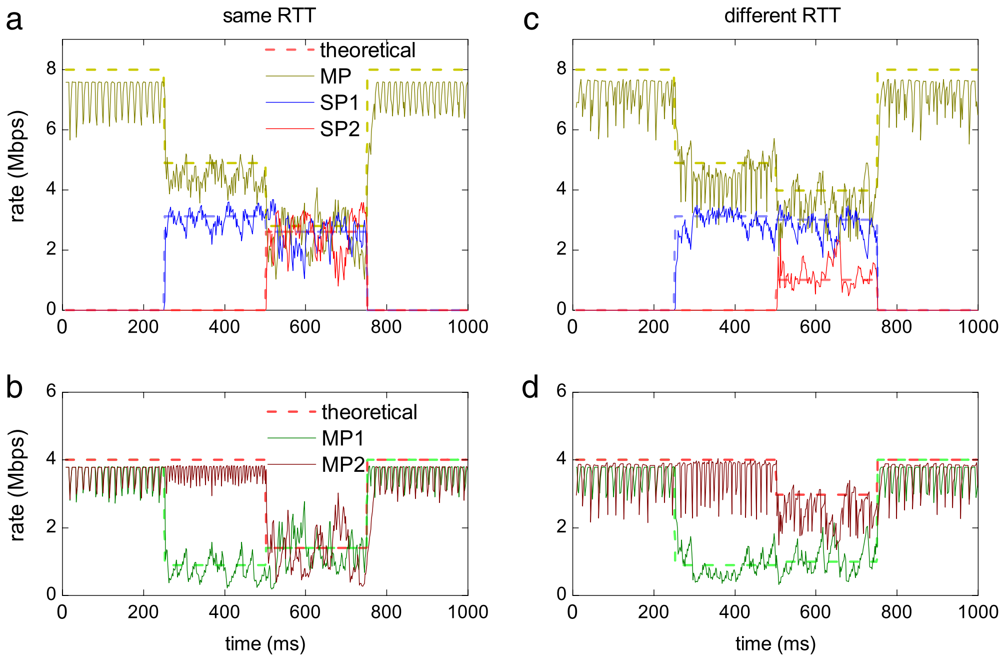

Fig. 5. Rate of flows and subflows in two-bottleneck-link network. MP1, MP2, MP, SP1, and SP2 are the rates of subflow 1, subflow 2, multi-path flow (MP1 + MP2), single-path flow 1, and single-path flow 2, respectively. (a)–(b) propagation delays of both links 1 and 2 are both 50 ms, (c)–(d) propagation delays of links 1 and 2 are 50 and 200 ms, respectively.

Table 1
Theoretical and practical values of (unbound) mReno, two-bottleneck-link topology (phase 3). MP1, MP2, MP, SP1, and SP2 are the rates of subflow 1, subflow 2, multi-path flow (MP1 + MP2), single-path flow 1, and single-path flow 2, respectively. Values on the second row of every cell are theoretical values.

	MP1	MP2	MP	SP1	SP2
Same RTT					
Phase 1	3.59 ± 0.00 4	3.62 ± 0.00 4	7.20 ± 0.00 8	0 0	0 0
Phase 2	0.95 ± 0.09 0.89	3.64 ± 0.01 4	4.60 ± 0.09 4.89	2.84 ± 0.09 3.11	0 0
Phase 3	1.40 ± 0.14 1.40	1.37 ± 0.12 1.40	2.77 ± 0.19 2.80	2.34 ± 0.14 2.60	2.43 ± 0.12 2.60
Phase 4	3.56 ± 0.04 4	3.56 ± 0.06 4	7.12 ± 0.04 8	0 0	0 0
Different RTT					
Phase 1	3.55 ± 0.01 4	3.63 ± 0.00 4	7.18 ± 0.01 8	0 0	0 0
Phase 2	0.89 ± 0.17 0.89	3.62 ± 0.01 4	4.51 ± 0.18 4.89	2.91 ± 0.17 3.11	0 0
Phase 3	1.09 ± 0.07 0.99	2.45 ± 0.07 2.98	3.54 ± 0.11 3.98	2.71 ± 0.07 3.01	1.24 ± 0.07 1.02
Phase 4	3.55 ± 0.02 4	3.63 ± 0.00 4	7.19 ± 0.02 8	0 0	0 0

different RTTs, the packet-level experiment results are consistently close to the theoretical values obtained by Algorithm 1.

4.4. Comparison with MPTCP

In this experiment, we compare our mReno to the multi-path transport protocol in [20,21,16] (MPTCP) in terms of three goals:

1. (Improve-throughput) the multi-path flow rate must be at least as if it was a single-path flow on its best path;
2. (Do-not-harm) the multi-path flows should not take more bandwidth on a path than if it was a single-path flow on that path; and
3. (Balance-congestion) a multi-path flow tends to shift the traffic off the congested path.

First, we notice that with utility function $\frac{-3/2}{D^2 x}$, $\frac{3/2}{D^2}$ serves as the weight in the maximizing the sum of negative terms. For example, if there are two different-RTT flows on the same bottleneck link, i.e., $D_1 < D_2$, then allocating more rate to flow 1 will increase the total $-\frac{3/2}{D_1^2 x_1} - \frac{3/2}{D_2^2 x_2}$ faster than allocating more rate to flow 2. As such, TCP Reno is known to discriminate against the path with a

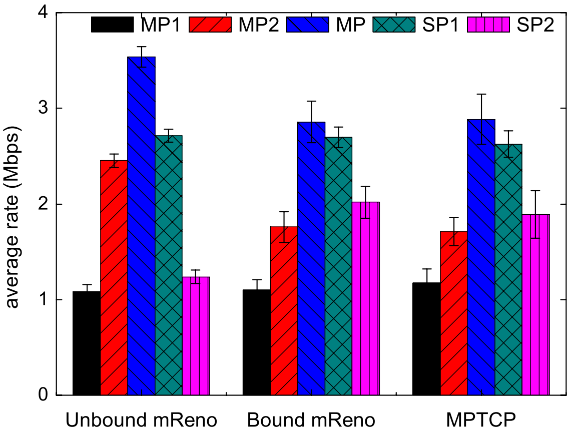

Fig. 6. Compare to MPTCP [20,21,16] (phase 3). MP1, MP2, MP, SP1, SP2 are the rates of subflow 1, subflow 2, multi-path flow (MP1 + MP2), single-path flow 1 and single-path flow 2, respectively. MP2 > SP2 in unbound mReno whereas SP1 > MP1, SP2 > MP2, and MP is larger than both SP1 and SP2 in bound mReno and MPTCP.

longer RTT. Choosing the utility for multi-path user with $D^{\min}$ in the non-separable term results in the utility of the multi-path user with the same weight as if it was the single-path user on the path with $D^{\min}$. Goal 1 is satisfied.

Second, mReno addresses Goal 3. As stated earlier, the equation $q_{s,i} = ((1-\epsilon)\hat{\theta}^2_{s,i} + \epsilon)\frac{3}{2D^2_{s,i}x^2_{s,i}}$ holds at the equilibrium. It is approximated to $\hat{\theta}^2_{s,i}\frac{3}{2D^2_{s,i}x^2_{s,i}} = \frac{3/2}{D_s^{\min 2}(\sum_{i\in\mathbf{s}} x_{s,i})^2}$ when ϵ is small. Hence, the mReno's controller always tries to balance congestion prices among its paths. If the congestion on one path is higher than that of the other paths, the rate allocated on this path will be reduced to decrease congestion. Balancing is perfect if $\epsilon = 0$. However, flappiness occurs as $\epsilon = 0$. When ϵ increases, the multi-path controller is less balanced as the approximation is no longer correct, but the flappiness is reduced as we have mentioned. This fact also agrees with a characteristic of MPTCP, where a trade-off between flappiness and load-balancing in the multi-path controller is observed [20, Section 2.1].

We now consider Goal 2. In Table 1, in the case with different RTTs, mReno results in a higher subflow 2 rate of the multi-path user than the rate of the single-path on the same link, i.e., MP2 > SP2. This result does not satisfy Goal 2 in the three design objectives of MPTCP. The multi-path user "harms" the single-path user on link 2. We apply the same technique as MPTCP to address Goal 3, i.e., bounding the increase part of every subflow without exceeding the increment as if it was a single-path flow on the same path [20,21]. As a result, *the window-size of the bound mReno (Bound mReno) increases by 1 if $\hat{\theta} \geq 1$ and by $(1-\epsilon)\hat{\theta}^2_{s,i}+\epsilon$ if $\hat{\theta} < 1$ in every RTTs; decreases by half whenever it detects a packet loss*. (Note that $\min(1, (1-\epsilon)\hat{\theta}^2_{s,i}+\epsilon)$ equals 1 if $\hat{\theta} \geq 1$ and equals $(1-\epsilon)\hat{\theta}^2_{s,i}+\epsilon$, otherwise.)

Fig. 6 shows the results of Bound mReno and MPTCP satisfying do-not-harm objective, whereas the (unbound) mReno does not. The rates of flows and subflows from Bound mReno are also close to those obtained from MPTCP.

5. Conclusions

We proposed a modified multi-path NUM problem, and solved it using successive approximation approach. Utilizing Jensen's inequality, we approximated the multi-path NUM to a separable and strictly convex problem. The solution sequence generated by solving the approximation problems successively is proven to converge to the globally optimal solution of the multi-path NUM. Based on the reverse engineering framework of TCPs, we also develop a series of multi-path TCPs compatible with the current single-path TCPs. Experiments on mReno show that the practical results are close to the theoretical results. A comparison with MPTCP is also presented. By bounding the increasing part of mReno, our multi-path protocol satisfies the three design goals of MPTCP.

Acknowledgments

The authors would like to thank Dr. Thai Duong Tran for useful discussions in the convergence proof. This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013- H0301-13-4006) supervised by the NIPA (National IT Industry Promotion Agency). Dr. CS Hong is the corresponding author.

References

[1] D.P. Bertsekas, A. Nedić, A.E. Ozdaglar, Convex Analysis and Optimization, Athena Scientific, 2003.

[2] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, NY, USA, 2004.

[3] M. Chiang, C.W. Tan, D. Palomar, D. O'Neill, D. Julian, Power control by geometric programming, IEEE Tran. Wirel. Commun. 6 (7) (2007) 2640–2651.

[4] G.B. Dantzig, J. Folkman, N. Shapiro, On the continuity of the minimum set of a continuous function, J. Math. Anal. Appl. 17 (1967) 519–548.

[5] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley, Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet, IEEE/ACM Trans. Netw. 14 (6) (2006) 1260–1271.

[6] J. He, M. Suchara, M. Bresler, J. Rexford, M. Chiang, Rethinking internet traffic management: from multiple decompositions to a practical protocol, in: Proceedings of the 2007 ACM CoNEXT Conference, 2007.

[7] B. Jiang, Y. Cai, D. Towsley, On the resource utilization and traffic distribution of multipath transmission control, Perform. Eval. 68 (11) (2011) 1175–1192.

[8] F. Kelly, T. Voice, Stability of end-to-end algorithms for joint routing and rate control, SIGCOMM Comput. Commun. Rev. 35 (2005) 5–12.

[9] X. Lin, N. Shroff, Utility maximization for communication networks with multipath routing, IEEE Trans. Automat. Control 51 (5) (2006) 766–781.

[10] S. Low, A duality model of tcp and queue management algorithms, IEEE/ACM Trans. Netw. 11 (4) (2003) 525–536.

[11] S. Low, D. Lapsley, Optimization flow control, i: basic algorithm and convergence, IEEE/ACM Trans. Netw. 7 (6) (1999) 861–874.

[12] S. Malik, Principles of Real Analysis, New Age International (P) Ltd, 1982.

[13] B.R. Marks, G.P. Wright, A general inner approximation algorithm for nonconvex mathematical programs, Oper. Res. 26 (4) (1978) 681–683.

[14] Multipath tcp on ns-2. [Online].
Available: http://code.google.com/p/multipath-tcp/.

[15] J. Papandriopoulos, S. Dey, J. Evans, Optimal and distributed protocols for cross-layer design of physical and transport layers in manets, IEEE/ACM Trans. Netw. 16 (6) (2008) 1392–1405.

[16] C. Raiciu, M. Handley, D. Wischik, Coupled congestion control for multipath transport protocols, in: RFC 6356, vol. 6356, 2011, pp. 1–12. [Online].
Available: http://www.ietf.org/rfc/rfc6356.txt.

[17] N. Tran, C.S. Hong, Joint rate and power control in wireless network: a novel successive approximations method, IEEE Commun. Lett. 14 (9) (2010) 872–874.

[18] P.L. Vo, N.H. Tran, C.S. Hong, Joint rate and power control for elastic and inelastic traffic in multihop wireless networks, in: Global Telecommunications Conference, 2011, Dec. 2011, pp. 1–5.

[19] W.-H. Wang, M. Palaniswami, S.H. Low, Optimal flow control and routing in multi-path networks, Perform. Eval. 52 (2–3) (2003) 119–132.

[20] D. Wischik, M. Handley, C. Raiciu, Control of Multipath TCP and Optimization of Multipath Routing in the Internet, in: R. Nez-Queija, J. Resing (Eds.), Ser. Lecture Notes in Computer Science, vol. 5894, Springer, Berlin, Heidelberg, 2009.

[21] D. Wischik, C. Raiciu, A. Greenhalgh, M. Handley, Design, implementation and evaluation of congestion control for multipath tcp, in: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, in: Ser. NSDI'11, USENIX Association, Berkeley, CA, USA, 2011, p. 8–8.

[22] W.I. Zangwill, Nonlinear Programming: A Unified Approach, Prentice-Hall, 1969.

Phuong Luu Vo received her B.E. and M.E. degrees in electrical–electronics engineering from University of Technology, Ho Chi Minh city, Vietnam in 1998, 2002, respectively. Currently, she is pursuing the Ph.D. degree in the department of Computer Engineering, Kyung Hee University, South Korea. Her research interests include network optimization and future internet technologies.

Tuan Anh Le received his MS and Ph.D. degrees in Computer Engineering from Kyung Hee University, Korea, in 2005, 2012, respectively. He received his BSc degree in Information Technology from Natural Sciences University, Ho Chi Minh city, Viet Nam in 2002. His research interests are traffic engineering, flow control, congestion control, and resource management.

Sungwon Lee received the Ph.D. degree from Kyung Hee University, Korea. He is a professor of the Computer Engineering Department at Kyung Hee University, Korea. He was a senior engineer of Telecommunications and Networks Division at Samsung Electronics Inc. from 1999 to 2008. He is an editor of the Journal of Korean Institute of Information Scientists and Engineers: Computing Practices and Letters. His research interests include mobile broadband networks and communication protocols. He is a member of IEEE, KIISE, KICS and KIPS.

Choong S. Hong received his B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. From Sept. 1993, he joined Keio University, Japan. He received his Ph.D. degree at Keio University in March 1997. He had worked for the Telecommunications Network Lab. KT served as a senior member of technical staff and as the director of the networking research team until August 1999. Since September 1999, he has worked as a professor of the department of computer engineering, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT and ICOIN. His research interests include Ad hoc Networks, Power Line Communications, Network Management and Future Internet. He is a member of IEEE, ACM, IEICE. IPSJ, KIISE, KICS and KIPS.

Byeongsik Kim received his B.E., M.S. and Ph.D. degrees in Computer Engineering from Chungnam National University, Daejeon, Korea, in 1993, 1995, and 2001, respectively. Before joining Electronics and Telecommunications Research Institute (Korea) he worked at National Institute of Standards and Technology (United States) from 1997 to 1998. He is a researcher at the Next Generation Communication Research Laboratory of ETRI. His research interests include Network Management, Content-Oriented Networking, and Routing.

Hoyoung Song is currently working as the director of Smart Node Team in ETRI, Rep. of Korea. He received M.A. and Ph.D. degrees in information & communication engineering from Chungbuk National University, and a B.S. degree in computer science from Hongik University, Seoul Korea. He has been with ETRI since 1983. His recent research interests include Information Centric Networking, Cloud networking, Software Defined Networking and future internet technologies.