Energy Efficient Multipath TCP for Mobile Devices

Qiuyu Peng
Dept. of EE, Caltech

Minghua Chen
Dept. of IE, CUHK

Anwar Walid
Bell Lab, Alcatel Lucent

Steven H. Low
Dept. of CMS & EE, Caltech

ABSTRACT

Most mobile devices today come with multiple access interfaces, *e.g.*, 4G and WiFi. Multipath TCP (MP-TCP) can greatly improve network performance by exploiting the connection diversity of multiple access interfaces, at the expense of higher energy consumption. In this paper, we design MP-TCP algorithms for mobile devices by jointly considering the performance and energy consumption. We consider two main types of mobile applications: realtime applications that have a fixed duration and file transfer applications that have a fixed data size. For each type of applications, we propose a two-timescale algorithm with theoretical guarantee on the performance. We present simulation results that show that our algorithms can reduce energy consumption by up to 22% without sacrificing throughput compared to a baseline MP-TCP algorithm.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks

Keywords

Multipath TCP; Energy Efficiency; Mobile Devices

1. INTRODUCTION

Even though most mobile devices today are equipped with multiple access interfaces such as 4G, 3G and WiFi, they only use one interface at any time. One access interface however may not be able to support the bandwidth requirement of emerging applications such as high definition video. It is therefore advantageous to enable concurrent use of the multiple interfaces.

Multipath TCP (MP-TCP), which is being standardized by the IETF [5], is a TCP extension that allows a TCP connection to stripe traffic over multiple interfaces. Various benefits of deploying MP-TCP on mobile devices have been reported in the literature. In [19], Wischik, *et al* show

that the throughput can be improved by using WiFi and 3G network together. In [17], Raiciu, *et al* show that MP-TCP enables smooth handovers between WiFi and 3G. Thus, MP-TCP is promising in providing more reliable and faster connections for mobile devices by leveraging connection diversities under dynamic environments.

One major concern of using MP-TCP in mobile devices is the higher energy required for maintaining multiple active interfaces [16]. A major portion of the energy for mobile devices is spent on communication [13, 15]. Thus, designing algorithms that can efficiently use multiple interfaces will greatly improve the battery lifespan. In [16, 17], energy saving mechanisms have been proposed to reduce energy consumption for mobile devices when MP-TCP is enabled. They periodically probe all paths and shift all traffic to the most energy efficient path and mute the other paths. It limits the throughput performance to that achieved using only one path. In addition, since energy cost is the only criterion for path selection, the throughput performance would be degraded significantly if the energy efficient path turns out to be the most congested path.

Our goal is to design MP-TCP algorithms that intelligently tradeoff throughput performance and energy consumption for mobile devices. We consider applications that need bulk data transport, *e.g.* video streaming and file transfer.

Real-time applications[1]: The duration T of a data transfer is usually fixed for this type of applications such as video streaming. The energy consumption is $P \times T$, where P is the instantaneous power. Since T is usually determined by external factors and can be regarded as a constant, the energy consumption is proportional to the instantaneous power P.

File transfer applications: The size B of a data transfer is usually fixed for this type of applications such as file download. The energy consumption is $P \times B/x$, where x is the throughput. Since B is usually determined by external factors and can be regarded as a constant, the energy consumption is proportional to P/x, termed "energy per bit".

Due to the difference in requirement of the real-time and file transfer applications, we need to design algorithms separately for each of them. We will see that sometimes the

---

[1]The issue of whether TCP is appropriate for these applications is beyond the scope of the paper; however, it has been reported that more than 50% of commercial streaming traffic is carried over HTTP/TCP [2], and popular applications such as Skype and Windows Media Services use TCP to pass through NATs and firewalls that block UDP traffic. Multipath live streaming over traditional TCP was studied in [18].

most energy efficient path for file transfer applications is 4G while for real-time applications is WiFi.

Contributions: The contributions of this paper are three-fold.

▷ We formulate a global optimization problem for real-time application that considers both throughput and energy consumption, whose tradeoff can be controlled by properly setting certain design parameters. The problem is hard to solve because the power consumption is not a continuous function of rate, *i.e.*, there is sunk cost of maintaining an active access interface. To solve the problem in a distributed manner, we propose a heuristic to decompose it into two subproblems: *path selection* and *congestion control*. For *path selection*, each source individually select a subset of paths to use to optimize certain objective based on its local information. We show that it is NP hard and propose an algorithm to solve it approximately with bounded suboptimality gap. For *congestion control*, we propose an algorithm that adapts the rates on the chosen paths based on both network congestion and energy consumption.

▷ We formulate a global optimization problem for file transfer application that considers both throughput and energy consumption, whose tradeoff can be controlled by properly setting certain design parameters. The problem has the additional difficulty of being non-convex even if the power consumption is linear in rate. We also decompose the problem into two subproblem: *path selection* and *congestion control*. Fortunately, we solve the *path selection* problem, which is mixed integer with nonconvex objective function, optimally under mild conditions. However, the non-convexity could lead to unstable *congestion control* algorithms and we stabilize it through a regulation term.

▷ We evaluate the algorithms for both real-time and file transfer applications through simulations. Our simulations show that the proposed algorithms indeed save energy and guarantee users' performance. In particular, the algorithms can reduce energy consumption by as much as 22% without sacrificing throughput compared to a baseline MP-TCP algorithm.

While path selection and congestion control problems have been studied in the literature before, our algorithms are unique in jointly optimizing the throughput and energy consumption whereas previous solutions focus only on throughput. Moreover, to incorporate energy consumption in the formulation, we need to deal with the additional non-convexity.

The rest of the paper is organized as follows. In Section 2, we describe our network model and present a model of power consumption for mobile devices. The path selection and congestion control algorithms for both types of applications are proposed in Sections 3 and 4, respectively. We illustrate the effectiveness of the algorithms through simulations in Section 5 and conclude the paper in Section 6. The proof of the theorems are relegated to the Appendix. The proof of the lemmas are skipped due to space limitation.

2. SYSTEM MODEL

In this section, we describe our network model, background of MP-TCP and the power consumption model for mobile devices. Unless otherwise specified, a boldface letter is used to denote a vector, whose dimension is clear from the context. $\|\mathbf{x}\|_n := \left(\sum x_i^n\right)^{1/n}$ defines the L^n-norm of a vector $\mathbf{x}$.

2.1 Network model and Multipath TCP

Consider a network that consists of a set $\mathcal{L} = \{1, \ldots, L\}$ of links. The network is shared by a set $\mathcal{S} = \{1, \ldots, S\}$ of sources. Available to source $s \in \mathcal{S}$ is a fixed collection of routes (paths) r. A path r consists of a set of links l. We abuse notation and use s both to denote a source and the set of paths r available to it, depending on the context. Likewise, r is used both to denote a path and the set of links l in the path. Let $\mathcal{R} := \{r \mid r \in s, s \in \mathcal{S}\}$ be the collection of all routes in the network. Let $H \in \{0,1\}^{|\mathcal{L}| \times |\mathcal{R}|}$ be the routing matrix: $H_{lr} = 1$ if link l is in route r (denoted by '$l \in r$'), and 0 otherwise.

For each path r, Let $x_r(t)$ be its sending rate at time t and τ_r denote its round trip time (RTT). For each link l, let $p_l(t)$ denote its congestion price at time t and c_l denote its bandwidth. Let $q_r(t) := \sum_{l \in \mathcal{L}} H_{lr} p_l(t)$ be the aggregate price on route r and $y_l(t) := \sum_{r \in \mathcal{R}} H_{lr} x_r(t)$ be the aggregate traffic on link l. We associate two state variables $(x_r(t), q_r(t))$ for each route $r \in s$. Let $\mathbf{x}_s(t) := (x_r(t), r \in s)$ and $\mathbf{q}_s(t) := (q_r(t), r \in s)$. Then $(\mathbf{x}_s(t), \mathbf{q}_s(t))$ represents the corresponding state variables for each source $s \in \mathcal{S}$.

Congestion control is a distributed algorithm that adapts $\mathbf{x}(t)$ and $\mathbf{p}(t)$ in a closed loop. In [14], all the congestion control algorithms (including both single-path and multi-path TCP) are modeled by

$$\dot{x}_r = k_r(\mathbf{x}_s) \left[\phi_r(\mathbf{x}_s) - q_r\right]_{x_r}^+ \qquad r \in s \quad s \in \mathcal{S} \quad (1)$$

$$\dot{p}_l = \gamma_l \left[y_l - c_l\right]_{p_l}^+ \qquad l \in \mathcal{L}, \quad (2)$$

where $[a]_b^+ = a$ if $b > 0$ and $\max\{0, a\}$ if $b \leq 0$. We omit the time t in the expression for simplicity. (1) models how sending rate is adapted in the congestion avoidance phase by TCP on each path and (2) models how the congestion price is updated on each link by AQM.

TCP (MP-TCP) adapts the size of its congestion window to control the sending rate, the corresponding parameters $(k_r(\mathbf{x}_s), \phi_r(\mathbf{x}_s))$ uniquely determine the window adaptation as described in [14, section 2.2]. Here $k_r(\mathbf{x}_s) \geq 0$ is a positive gain that determines the dynamic property and $\phi_r(\mathbf{x}_s)$ determines the equilibrium property.

The AQM is specified by γ_l, where $\gamma_l > 0$ is a positive gain that determines the dynamic property. This is a simplified model for the random early detection(RED) algorithm that assumes the loss probability is proportional to the backlog.

2.2 Utility Maximization

For single-path TCP, where the rate x_s is a scalar, under very mild conditions, one can show that the TCP/AQM algorithms (1)-(2) solve a network utility maximization problem below [8, 10, 11]:

$$\max \sum_{s \in \mathcal{S}} U_s(x_s) \quad \text{s.t.} \quad y_l \leq c_l \text{ for } l \in \mathcal{L} \quad (3)$$

where $U_s(x_s)$, which is called the utility of source s and satisfies C1 below, represents user's satisfaction at rate x_s, and $y_l \leq c_l$ represents the constraint on each link l that its input traffic y_l is no more than available bandwidth c_l.

C1: For $s \in \mathcal{S}$, $U_s(x_s)$ is twice continuously differentiable and strictly concave increasing in $(0, \infty)$.

Table 1: Comparison of 4G, 3G and WiFi. [7]

	4G	3G	WiFi
Median Upload Speed (Mbps)	5.64	0.72	0.94
Median Download Speed (Mbps)	12.74	1.10	4.12
Download b_r (mW/Mbps)	52.0	122.1	137
Upload b_r (mW/Mbps)	438.4	869	238.2
θ_r (mW)	1288	817.9	132.9

The utility function of most single-path TCP algorithms take the form of α-fairness function [12] as defined in (4).

$$U_s(x;\alpha) = \frac{\gamma}{1-\alpha}x^{1-\alpha}, \tag{4}$$

where $\alpha, \gamma \geq 0$ are constants. For instance, the utility function of the well known TCP NewReno is

$$U_s(x) = -\frac{2}{\tau^2 x}, \tag{5}$$

where τ is the round trip time. It corresponds to $(\alpha, \gamma) = (2, 2\tau^{-2})$ in (4). When multiple paths are available, a natural generalization of (3) is

$$\max \quad \sum_{s \in \mathcal{S}} U_s(\mathbf{1}^T \mathbf{x}_s) \quad \text{s.t.} \quad y_l \leq c_l \text{ for } l \in \mathcal{L} \tag{6}$$

where $\mathbf{1}$ is the all one vector and $\mathbf{1}^T \mathbf{x}_s$ represents the aggregate throughput of source s over all its paths. In (6), each source s only cares about its aggregate throughput $\mathbf{1}^T \mathbf{x}_s$ without considering its energy consumption. In the following, we amend (6) to include energy consumption in the objective. We first investigate the power consumption characteristics of mobiles.

2.3 Power consumption for mobile devices

The wireless interfaces on most mobile devices are 4G, 3G and WiFi. In [7], Huang, *et al* found that the the power consumption $P_r(x_r)$ on path r depends on the throughput x_r according as

$$P_r(x_r) = b_r x_r + \theta_r 1_{\{x_r > 0\}}, \tag{7}$$

where $\theta_r > 0$ models the sunk power cost of being active and $b_r > 0$ models the power consumed per unit throughput. Note that $P_r(x_r)$ is not continuous at $x_r = 0$, making the design of energy efficient algorithm a challenge. The model (7) captures power consumption for most telecommunication standards such as 4G, 3G and WiFi. Some representative parameter values from [7] are summarized in Table 1.

3. REAL-TIME APPLICATIONS

For realtime applications, *e.g.* video streaming, the amount of time T for communication is controlled by users. The energy spent on communication for this type of applications is $T(\sum_{r \in s} P_r(x_r))$. Thus, optimizing energy consumption is equivalent to optimizing power consumption $\sum_{r \in s} P_r(x_r)$. For this type of applications, we add an additional penalty term to represent energy consumption in (6) and the prob-

lem becomes

$$\max \quad \sum_{s \in \mathcal{S}} \left(U_s(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \sum_{r \in s} P_r(x_r) \right) \tag{8a}$$

$$\text{s.t.} \quad y_l \leq c_l \text{ for } l \in \mathcal{L} \tag{8b}$$

We assume for each source $s \in \mathcal{S}$, $U_s(\mathbf{1}^T \mathbf{x}_s)$ satisfies C1 and can be interpreted as users' level of satisfaction for the throughput. $\sum_{r \in s} P_r(x_r)$ represents the power consumption. The parameters $\alpha_s \geq 0$ control the tradeoff between users' satisfaction and power consumption.

Solving (8) is hard because $P_r(x_r)$ is discontinuous at $x_r = 0$ as described in (7). In Lemma 3.2 below, we will show that the problem is NP hard even if there is only one source in the network. To tackle the problem efficiently, we propose to solve the problem in two steps as follows:

1. *Path Selection*: The source first decides a subset of paths $s^* \subseteq s$ to use based on available local information, *e.g.* power characteristics, bandwidth available on each path $r \in s$.

2. *Congestion Control*: Once the set of paths s^* is fixed, the source s turns on the interfaces of these paths and adapts sending rates to network congestion level and energy consumption.

Paths should be re-selected from time to time to adapt to changes in power characteristics or available bandwidths, but these conditions usually change at a much slower timescale than congestion level does. Normally we can periodically probe each path as [17] and rerun path selection after that. However, we should avoid doing it too frequently due to the high energy switching cost for turning interfaces on and off. Next, we will formally define each subproblem and solve it.

3.1 Path Selection

In this step, each source s selects a subset $s^* \subseteq s$ of paths based on local information, *e.g.* the available bandwidth c_r of the access link, allocated by the base stations (4G/3G) or the hotspots (WiFi), and the parameters (b_r, θ_r) in (7) for each path $r \in s$. Then the source s seeks to solve the following problem:
Path Selection Problem-I(PSP-I):

$$\max \quad U_s(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \sum_{r \in s} P_r(x_r) \tag{9a}$$

$$\text{s.t.} \quad x_r \leq c_r \text{ for } r \in s \tag{9b}$$

Let the optimal solution to PSP-I be $\mathbf{x}_s^*$. The paths selected by s are then $s^* = \{r \mid x_r^* > 0, r \in s\}$. Comparing PSP-I with the global optimization problem (8), PSP-I ignores the bandwidth constraints except at the access links. Thus PSP-I provides an upper bound on the throughput. It is a good estimate of the global optimization problem (8) if the access links are bottleneck, which usually holds for most telecommunication standards such as 4G, 3G and WiFi.

Remark: Various path selection problems are formulated in the literature. In [9], Key, *et al* study the performance of both randomized and game theoretic approach to select a fixed number of paths from a pool of available paths. In [3], the markov approximation approach is proposed for choosing peers in P2P system. Unlike PSP-I, neither of them considers the sunk cost but limits the number of peers one can select to implicitly reduce the sunk cost.

Table 2: Candidate for PSP-I with $|s| = 2$

x_1	x_2
$\min\{c_1, (U'_s)^{-1}(\alpha_s b_1)\}$	0
0	$\min\{c_2, (U'_s)^{-1}(\alpha_s b_2)\}$
c_1	$\min\{\max\{(U'_s)^{-1}(\alpha_s b_2) - c_1, 0\}, c_2\}$

We now characterize an optimal solution to PSP-I in Lemma 3.1.

Lemma 3.1. *Suppose C1 holds and $b_i \leq b_j$ for $1 \leq i < j \leq |s|$ without loss of generality. Then the path s^* being chosen is given as*

$$s^* = \arg\max_{s' \subseteq s} \left\{ U_s(\sum_{r \in s'} x_r(s')) - \alpha_s \sum_{r \in s'} P_r(x_r(s')) \right\},$$

where for each subset $s' := \{k_1, \ldots, k_{|s'|}\}$ of s,

$$x_{k_i}(s') := \left[(U'_s)^{-1}(\alpha_s b_{k_i}) - \sum_{j=1}^{i} c_{k_j} \right]_0^{c_{k_i}}$$

and $[a]_b^c := \max\{b, \min\{a, c\}\}$.

Based on Lemma 3.1, we need to compare the objective value of each subset of s to obtain the optimal solution to PSP-I. Solving PSP-I is in general NP hard because the knapsack problem, which is a well known NP complete problem, can be reduced to PSP-I as shown in Lemma 3.2 below.

Lemma 3.2. *Suppose C1 holds. Then PSP-I is NP hard.*

However, there are only two interfaces (4G and WiFi) for most mobiles and we only need to check each combination in Table 2 according to Lemma 3.1.

When there are many WiFi hotspots around, the number of available paths $|s|$ can be larger [4]. Thus PSP-I is hard to be solved efficiently according to Lemma 3.2. We now propose an efficient heuristic for solving PSP-I that has linear complexity and a bounded suboptimality gap.

First consider the case where $\theta_r = 0$ for all $r \in s$ (recall that θ_r is the sunk power cost of using path r), then PSP-I degenerates to a convex program:

$$\max \quad U_s(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \sum_{r \in s} b_r x_r \qquad (10a)$$

$$\text{s.t.} \quad x_r \leq c_r \text{ for } r \in s \qquad (10b)$$

The KKT conditions imply that the source s uses path i if and only if all the links j with $b_j < b_i$ have been fully used. Assume without loss of generality $b_j \leq b_i$ for $j < i$. We initialize $x_r = 0$ for all $r \in s$. Then we select path 1 and calculate the maximum of $U_s(x_1) - \alpha_s b_1 x_1$ for $x_1 \in [0, c_1]$. We stop if its solution $x_1 < c_1$, which means the objective is maximized merely by path 1. Otherwise if $x_1 = c_1$, which means the first path is fully utilized, we add path 2 and calculate maximum of $U_s(c_1 + x_2) - \alpha_s(b_1 c_1 + b_2 x_2)$ for $x_2 \in [0, c_2]$. Likewise, we stop if its solution $x_2 < c_2$ and add path 3 if $x_2 = c_2$. The paths will be added sequentially until $x_k < c_k$ for some $k < |s|$ or $k = |s|$. Then the corresponding $\mathbf{x}_s$ is an optimal solution to (10) and the paths we use are $\{1, \ldots, k\}$.

Motivated by the above procedure to solve (10), we design an algorithm to solve PSP-I when $\theta_r \neq 0$, in a similar manner. For $r \in s$, let $b'_r := b_r + \theta_r/c_r$, which is the power consumption per unit throughput on path r when the throughput $x_r = c_r$. Without loss of generality, assume $b'_i \leq b'_j$ for $1 \leq i < j \leq |s|$. Then we mimic what we do to solve (10): add path in increasing order of b'_i and stop if the newly added path is not fully utilized, as stated in Algorithm 1.

Algorithm 1: Path Selection Algorithm for PSP-I

Input: $U_s(x)$, α_s and (b_r, θ_r, c_r) for $r \in s$.

Output: Subset $s^* \subseteq s$.

1. Calculate $\tilde{x}_k$ and O_k as below:

$$\tilde{x}_k = \left[(U'_s)^{-1}(\alpha_s b_k) - \sum_{1 \leq i < k} c_i \right]_0^{c_k} \qquad (11)$$

$$O_k = U_s(\tilde{x}_k + \sum_{1 \leq i < k} c_i) - \alpha_s(\sum_{1 \leq i < k}(b_i c_i + \theta_i) + \theta_k + b_k \tilde{x}_k)$$

2. Let $k^* = \arg\max\{O_k, k \leq |s|\}$ and $s^* \leftarrow \{1, \cdots, k^*\}$

Algorithm 1 is not guaranteed to find a global optimal solution to PSP-I since b'_r may not be the real power consumption per unit throughput. Next, we bound the gap between the optimal value of PSP-I O^* and the objective value of Algorithm 1 $\hat{O}$. Denote

$$\hat{x}_k := \left[(U'_s)^{-1}(\alpha_s b'_k) - \sum_{1 \leq i < k} c_i \right]_0^{c_k} \qquad (12)$$

$\hat{k} := \max\{k \mid \sum_{i=1}^{k-1} c_i \leq (U'_s)^{-1}(\alpha_s b'_k)\}$ and $\hat{c} := \sum_{i=1}^{\hat{k}-1} c_i$.

Theorem 3.1. *Suppose C1 holds. Then*

$$O^* - \hat{O} \leq \min \left\{ \frac{1}{2} \hat{x}_{\hat{k}}(U'_s(\hat{c}) - \alpha_s b'_{\hat{k}}), \alpha_s \theta_{\hat{k}}(1 - \frac{\hat{x}_{\hat{k}}}{c_{\hat{k}}}) \right\}$$

Theorem 3.1 implies that the gap is upper bounded by $\alpha_s \theta_{\hat{k}}$, *i.e.* the algorithm works better when θ_k are smaller and the gap is closed if there are no sunk costs.

3.2 Congestion Control

Once the source fixes the paths $s^* \subseteq s$ it uses, it turns on the access interfaces of the corresponding paths and adapts the rate on those paths to network congestion and energy consumption. Similar to the design of congestion control for regular MP-TCP in [14], we need to design $(k_r(\mathbf{x}_s), \phi_r(\mathbf{x}_s))$ in (1)-(2) and then we convert it to window based algorithm following a standard procedure derived in [14].

The power consumption is determined by the equilibrium properties (steady state throughput), which is not affected by $k_r(\mathbf{x}_s)$, so we will use the same $k_r(\mathbf{x}_s)$ as proposed in [14]:

$$k_r(\mathbf{x}_s) = \frac{1}{2} x_r(x_r + \|\mathbf{x}_s\|_\infty). \qquad (13)$$

Interested readers can refer to [14, section 4.2] for more details. Next, we need to design $\phi_r(\mathbf{x}_s)$ such that (1)-(2) solves (8). Note that the discontinuity in (8) disappears since the paths for each source are fixed in the path selection step, a standard primal dual decomposition shows that by setting

$$\phi_r(\mathbf{x}_s) = U'_s(\mathbf{1}^T \mathbf{x}_s) - \alpha_s b_r,$$

for $r \in s^*$, (1)-(2) is guaranteed to converge to an optimal solution of (8). This algorithm however is not responsive in a dynamic network because the Hessian of $U_s(\mathbf{1}^T\mathbf{x}_s)$ is only negative semidefinite [1]. To obtain a more responsive algorithm, we multiply a regulation term $f_r(\mathbf{x}_s)$ in front of $U_s'(\mathbf{1}^T\mathbf{x}_s)$ as done in [14] such that

$$\phi_r(\mathbf{x}_s) = f_r(\mathbf{x}_s)U_s'(\mathbf{1}^T\mathbf{x}_s) - \alpha_s b_r.$$

A proper $f_r(\mathbf{x}_s)$ ensures that the Jacobian of $\Phi_s(\mathbf{x}_s) := (\phi_r(\mathbf{x}_s), r \in s)$ is strictly negative definite. We use the same $f_r(\mathbf{x}_s)$ as proposed in [14]:

$$f_r(\mathbf{x}_s) = \beta\frac{\|\mathbf{x}_s\|_\infty}{x_r} + 1 - \beta, \qquad (14)$$

where $\beta \in [0,1]$ and larger β yields a more responsive algorithm. However, it is shown in [14] that a more responsive MP-TCP algorithm is less friendly[2] to single-path TCP. The parameter β controls the tradeoff between responsiveness and friendliness and a larger β yields a more responsive but less friendly algorithm. Here we use $\beta = 0.2$ as suggested in [14] and stability of the system (1)-(2) can be proved in a similar manner as [14].

Remark: To the best of our knowledge, all existing MP-TCP algorithms can be modeled by multiplying a regulation term $f_r(\mathbf{x}_s)$ in front of a marginal utility $U_s'(\mathbf{1}^T\mathbf{x}_s)$, *e.g.* [6,14,19]. This improves the responsiveness compared to an algorithm without a regulation term, but effectively modifies the utility function in (8). Indeed, for some $f_r(\mathbf{x}_s)$, the resulting algorithm does not maximize any (modified) utility functions [14].

In summary, the congestion control algorithm for realtime application is given as sequel.

Congestion Control for Realtime Application:
$$k_r(\mathbf{x}_s) \quad := \quad \frac{1}{2}x_r(x_r + \|\mathbf{x}_s\|_\infty).$$ $$\phi_r(\mathbf{x}_s) \quad := \quad f_r(\mathbf{x}_s)U_s'(\mathbf{1}^T\mathbf{x}_s) - \alpha_s b_r.$$
where $f_r(\mathbf{x}_s)$ is defined in (14).

---

4. FILE TRANSFER APPLICATIONS

In this section, we design an energy efficient MP-TCP algorithm for file transfer applications. Similar to real-time applications, there are two steps in the algorithms: path selection and congestion control. Each source decides a subset of paths to use in the path selection step and adapts rates on the chosen paths to network congestion and energy consumption.

Unlike realtime application, the size of a data transfer B is fixed for file transfer application. Then the total energy consumed for source s is $(\sum_{r \in s} P_r(x_r))\frac{B}{\mathbf{1}^T\mathbf{x}_s}$ per transfer. Since the data size B is controlled by the users and can be regarded as a constant, optimizing energy consumption is equivalent to optimize energy per bit $\frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T\mathbf{x}_s}$. For this type of applications, we amend (6) and seek to solve the

[2]Friendliness performance is a metric describing how much MP-TCP flows overtake single-path TCP flows at bottleneck links.

following optimization problem

$$\max \quad \sum_{s \in \mathcal{S}}\left(U_s(\mathbf{1}^T\mathbf{x}_s) - \alpha_s\frac{1}{\mathbf{1}^T\mathbf{x}_s}\sum_{r \in s}P_r(x_r)\right) \qquad (15a)$$

$$\text{s.t.} \quad y_l \le c_l \text{ for } l \in \mathcal{L} \qquad (15b)$$

Similar to (8), the problem (15) considers users' satisfaction through the utility function $U_s(\mathbf{1}^T\mathbf{x}_s)$ and the energy consumption through the energy per bit $\frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T\mathbf{x}_s}$. Solving (15) is hard not only because of the discontinuity of $P_r(x_r)$ at $x_r = 0$, but also the nonconvexity of the fractional $\frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T\mathbf{x}_s}$. Similar to what we do in section 3, we will also decompose the problem into two steps: path selection and congestion control.

4.1 Path Selection

In this step, each source selects a subset of paths $s^* \subseteq s$ to use based on prior local information, *e.g.* the available bandwidth c_r and the power throughput characteristic b_r, θ_r for each path r of source s. Then the source s seeks to solve the following problem:
Path Selection Problem-II (PSP-II):

$$\max \quad U_s(\mathbf{1}^T\mathbf{x}_s) - \alpha_s\frac{1}{\mathbf{1}^T\mathbf{x}_s}\sum_{r \in s}P_r(x_r) \qquad (16a)$$

$$\text{s.t.} \quad x_r \le c_r \text{ for } r \in s \qquad (16b)$$

Compared with PSP-I, PSP-II is nonconvex even if the sunk cost $\theta_r = 0$ in $P_r(x_r)$. Fortunately, we will show below that under a mild condition C2, which covers most existing TCP algorithms, on the utility functions $U_s(x)$, Algorithm 2 solves PSP-II exactly. Again, let $b_r' := b_r + \theta_r/c_r$ and assume $b_i' \le b_j'$ for $1 \le i < j \le |s|$ without loss of generality.

Algorithm 2: Path Selection Algorithm-II
Input: $U_s(x)$, α_s and (b_r, θ_r, c_r) for $r \in s$.
Output: Subset $s^* \subseteq s$.
1. Calculate $$O_k = U_s\left(\sum_{i=1}^{k}c_i\right) - \alpha_s\frac{\sum_{i=1}^{k}b_i'c_i}{\sum_{i=1}^{k}c_i}$$
2. Let $k^* = \arg\max_{1 \le k \le |s|}\{O_k\}$ and $s^* \leftarrow \{1, \ldots, k^*\}$.

C2 : $x^2U_s'(x)$ is a nondecreasing function of x.

Theorem 4.1. *Suppose C1 and C2 hold. Then Algorithm 2 obtains an optimal solution to PSP-II.*

Suppose the utility function takes the α-fairness form in (4), then

$$x^2U_s'(x;\alpha) = \gamma x^{2-\alpha},$$

which means α-fairness function satisfies C2 when $\alpha \in [0,2]$. Thus, PSP-II can be solved exactly by Algorithm 2 for TCP algorithms with α-fairness utility functions provided $\alpha \in [0,2]$. For most existing TCP protocols, their $\alpha \in [0,2]$, *e.g.* TCP Vegas/FAST ($\alpha = 1$) and TCP NewReno ($\alpha = 2$).

Here we need to point out that for Algorithm 1, the sequence $\{O_k, 1 \le k \le |s|\}$ is unimodal and hence the algorithm can stop once the sequence O_k becomes nonincreasing at some k. However, this property does not hold for

Algorithm 2 due to the non-convexity of the objective function. Thus, we need to check every element in the sequence $\{O_k, 1 \leq k \leq |s|\}$.

4.2 Congestion Control

Similar to the design of realtime applications, we need to specify $(k_r(\mathbf{x}_s), \phi_r(\mathbf{x}_s))$ in (1). For $k_r(\mathbf{x}_s)$, we use the same one as realtime application (13) because it does not affect the equilibrium properties (throughput). For $\phi_r(\mathbf{x}_s)$, even though the discontinuity disappears once paths $s^* \subseteq s$ are fixed, a standard primal dual algorithm (1)-(2) with

$$\phi_r(\mathbf{x}_s) = U_s'(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \frac{\partial E_s(\mathbf{x}_s)}{\partial x_r} \qquad (17)$$

where $E_s(\mathbf{x}_s) := \frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T \mathbf{x}_s}$, may not be stable since the Jacobian of $\Phi_s(\mathbf{x}_s) := (\phi_r(\mathbf{x}_s), r \in s)$ is indefinite. Similar to the scenario of real time application in section 3, we multiply a regulation term $f_r(\mathbf{x}_s)$ in front of $U_s'(\mathbf{1}^T \mathbf{x}_s)$ such that

$$\phi_r(\mathbf{x}_s) = f_r(\mathbf{x}_s)U_s'(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \frac{\partial E_s(\mathbf{x}_s)}{\partial x_r} \qquad (18)$$

The regulation term $f_r(\mathbf{x}_s)$ is merely used to make the algorithm more responsive in section 3. Here, it guarantees a stable algorithm by compensating the indefiniteness of the Jacobian, as the following theorem shows. For convenience, we will only prove the result for the utility function of TCP NewReno defined in (5). Stability can be proved in a similar manner for other utility functions.

Theorem 4.2. *Suppose the round trip time τ of each path is the same. For $\phi_r(\mathbf{x}_s)$ in (18), $f_r(\mathbf{x}_s)$ in (14), and $U_s(x)$ in (5), the system (1)-(2) is stable if*

$$\|\mathbf{x}_s\|_\infty \leq \frac{\beta}{2\alpha_s \tau^2} \frac{9/|s^*| - 1}{\max_{r \in s^*} b_r - \min_{r \in s^*} b_r}, \qquad (19)$$

Only the selected paths in s^* needs to satisfy (19). When the tradeoff parameter α_s increases, the size of s^* becomes small, which means $\max_{r \in s^*} b_r - \min_{r \in s^*} b_r$ also decreases and the bound becomes looser. The bound in (19) is satisfied for most practical scenario. For instance, the bound reduces to $\|\mathbf{x}_s\|_\infty \leq 0.02 \frac{\beta}{\alpha_s \tau^2}$ for smartphones with both 4G and WiFi interface using the median value of b_r in Table 1. In our simulations, $\alpha_s \tau^2$ is typically in $(10^{-5}, 10^{-3})$. If $\beta = 1$, (19) becomes $\|\mathbf{x}_s\|_\infty \leq 2000$Mbps for $\alpha_s \tau^2 = 10^{-5}$ and $\|\mathbf{x}_s\|_\infty \leq 20$Mbps for $\alpha_s \tau^2 = 10^{-3}$, which seems realistic. In addition, the algorithm is alway stable in our preliminary simulations even if the round trip time is different. Thus, the bound develop in Theorem 4.2 is conservative.

In summary, the congestion control algorithm for file transfer application is given as sequel.

Congestion Control for File Transfer Application:

$$k_r(\mathbf{x}_s) \quad := \quad \frac{1}{2} x_r(x_r + \|\mathbf{x}_s\|_\infty).$$

$$\phi_r(\mathbf{x}_s) \quad := \quad f_r(\mathbf{x}_s)U_s'(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \frac{\partial E_s(\mathbf{x}_s)}{\partial x_r}.$$

where $f_r(\mathbf{x}_s)$ is defined in (14) and $E_s(\mathbf{x}_s) := \frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T \mathbf{x}_s}$.

Figure 1: Smartphone with both 4G and WiFi access

5. DISCUSSIONS

In this section, we discuss how to setup the parameters of the algorithms and their impacts on the performance. In particular, we find that the algorithms can save up to 22% of the energy consumption without sacrificing the throughput. We also discuss the cases when there are multiple long term TCP flows from the same device. Unless otherwise specified, we use Table 1 for parameter values: For WiFi network, $(c_r, b_r, \theta_r) = (4.12$ Mbps,238.2 mW/Mbps,132.9 mW); for 4G network, $(c_r, b_r, \theta_r) = (12.74$ Mbps,52 mW/Mbps,1288 mW). As to estimation of these physical parameters, it is beyond the scope of the paper. In general, we can estimate these parameters through probing the network periodically. Interested readers may refer to [13, 15–17] for details.

5.1 Parameters setting and energy saving performance

Most of the existing MP-TCP algorithms [6, 14, 19] use TCP-NewReno as a benchmark single path TCP as specified by the IETF working group [5]. Thus, all of their associated utility functions take the form of (5). To achieve balanced performance between friendliness and responsiveness, a regulation term $f_r(\mathbf{x}_s)$ is multiplied in front of the marginal utility function $U_s'(\mathbf{1}^T \mathbf{x}_s)$, such that $\phi_r(\mathbf{x}_s) = f_r(\mathbf{x}_s)U_s'(\mathbf{1}^T \mathbf{x}_s)$ in (1). Different algorithms correspond to different regulation term $f_r(\mathbf{x}_s)$, which controls the tradeoffs between friendliness and responsiveness.

For the proposed algorithms in section 3 and 4, we also let the utility function $U_s(\mathbf{1}^T \mathbf{x}_s)$ take the form of (5) since TCP-NewReno is the benchmark. The regulation term takes the form of (14). Then the only parameter left unset is α_s, which controls the tradeoff between users' satisfaction and energy consumption. In normal operations, the users can tune α_s based on their own preference: they can save energy by increasing α_s but maybe suffer throughput reduction. Fortunately, sometimes the proposed algorithms can still save energy consumption without affecting the throughput compared to regular MP-TCP. Next, we will study the impact of the value of α_s on throughput and energy consumption through simulations.

Almost all smartphones have both 4G and WiFi access as shown in Fig. 1. In the first set of simulations, we assume the bottleneck is in the network by setting $c_M = 10$Mbps, while the bandwidth of the WiFi and 4G access are 4.12 Mbps and 12.74 Mbps as described in Table 1. Under such scenario, we test the algorithms for both video streaming, which represents realtime application and file download, which represents file transfer application. The simulation results are plotted in Fig. 2. For video streaming, we plot the aggregate throughput and power consumption $\sum_{r \in s} P_r(x_r)$ under different values of α_s. For file download, we plot the aggregate throughput and the energy per bit $\frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T \mathbf{x}_s}$ under different values of α_s. For comparison, we plot the en-
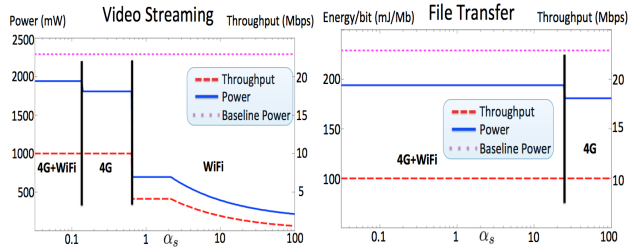
Figure 2: Energy, throughput vs α_s for $c_M = 10$Mbps. The pink dotted line represents the energy consumption of regular MP-TCP.
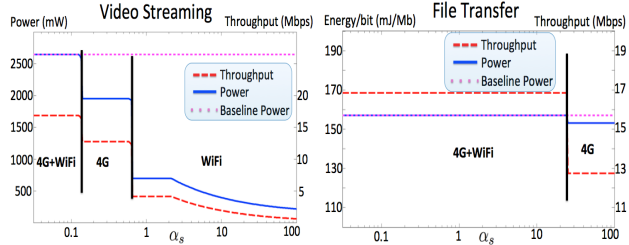

Figure 3: Energy, throughput vs α_s for large c_M. The pink dotted line represents the energy consumption of regular MP-TCP.

ergy consumed by running baseline regular MP-TCP, whose throughput is always c_M.

For video streaming, the smartphone uses both 4G and WiFi when α_s is small, it switches to 4G as α_s increases and eventually switch to WiFi network. The baseline MP-TCP consumes 2291mW power while the proposed algorithm consumes less power for any $\alpha_s > 0$. In particular, the proposed algorithm only consumes 1808mW when $\alpha_s \in [0.11, 0.65]$ while the throughput is the same as running regular MP-TCP. Thus, the proposed algorithm saves 22% energy without affecting the throughput. When $\alpha_s \geq 0.65$, both power and throughput decreases as α_s increases and there is tradeoff between throughput and energy consumption. For file download, the smartphone still uses both 4G and WiFi when α_s is small. Similar to video streaming, the proposed algorithm also saves 22% energy without affecting the throughput when $\alpha_s \geq 24$. However, the smartphone always uses 4G interface and operates at rate 10Mbps however large α_s is. Thus, the most energy efficient path depends on the applications and sometimes may not be the same.

We also simulate the case when the bottleneck is at the access link, assuming the bandwidth c_M is very large. The results are shown in Fig. 3 and are similar to our first set of simulations where $c_M = 10$Mbps. The proposed algorithms still save energy compared to regular MP-TCP for any α_s. However, it cannot save energy without affecting throughput as our first set of simulations, $i.e.$ we need to sacrifice more throughput if we want to save more energy.

Based on the above simulations, the path selection algorithms for realtime and file transfer applications end up at different paths for large α_s, $i.e.$ the most energy efficient path is different for these two applications in the simulation scenario. Actually, we can explain the phenomena using the following two lemmas, which characterize the most energy efficient path for both types of applications. For source

Table 3: Comparisons of Algorithm 1 and Algorithm 2.

	Throughput of video call	Completion time of file transfer	Total energy consumed
Regular MPTCP (4G+WiFi)	8.42 Mbps	119 s	794 J
Algorithm 1(WiFi)	0.40 Mbps	369 s	213 J
Algorithm 2(4G)	0.64 Mbps	82.7 s	161.4 J

s, let $S_1 = \arg\min\{b_r \mid r \in \arg\min\{\theta_k \mid k \in s\}\}$ and $S_2 = \arg\min\{b_r' \mid r \in s\}$.

Lemma 5.1. *Suppose C1 holds. Let $\mathbf{x}_s^*(\alpha_s)$ be the optimal solution to PSP-I given α_s. Then there exists an $\bar{\alpha}_s > 0$ such that when $\alpha_s \geq \bar{\alpha}_s$, there is only one nonzero element in $\mathbf{x}_s^*(\alpha_s)$, which can be any element $r_1 \in S_1$. In addition, $\lim_{\alpha_s \to \infty} x_{r_1}^*(\alpha_s) = 0$.*

Lemma 5.1 describes the optimal path being selected for α_s large enough. It says that the most energy efficient path is the one with minimal θ_r for realtime applications, and the throughput approaches 0 as α_s increases.

Lemma 5.2. *Suppose C1 and C2 hold. Let $\mathbf{x}_s^*(\alpha_s)$ be the optimal solution to PSP-II given α_s. Then there exists a $\bar{\alpha}_s > 0$ such that when $\alpha_s \geq \bar{\alpha}_s$*

$$x_r^*(\alpha_s) = \begin{cases} 0 & r \in s \setminus S_2 \\ c_r & r \in S_2 \end{cases}$$

Lemma 5.2 describes the limiting behavior of file transfer when α_s is large enough. Different from realtime application, its most energy efficient path is the one with minimal b_r' for α_s large enough and it occupies all the available bandwidth on paths with minimal b_r'.

5.2 Multiple TCP connections from a single device

In the above analysis, we assume there is only one long termed TCP connection initiated from a single device. Sometimes there will be multiple concurrent long termed TCP connections initiated from a single device, $e.g.$ there are two concurrent file downloads. In this case, we need to jointly optimize across all the TCP connections from one device because the total power power consumption is not separable for each connection, $i.e.$ they are coupled through the 0-1 sunk cost in (7). However, we can run the same congestion control algorithms as described in section 3 and 4 since the sunk cost becomes a constant once the paths are selected. Thus, we only need to modify the path selection algorithms.

When all the TCP connections carry traffic of the same type of applications (either real time or file transfer application), the proposed path selection algorithms are ready to be generalized. Suppose all the TCP connections carry realtime traffic. Each TCP connection can access the same set of interfaces and will share the bandwidth equally, thus PSP-I (9) can be generalized as sequel.

$$\max \quad NU_s(\mathbf{1}^T\mathbf{x}_s) - \alpha_s \sum_{r \in s} P_r(x_r) \tag{20a}$$

$$\text{s.t.} \quad x_r \leq \frac{c_r}{N} \quad r \in s \tag{20b}$$

where N is the number of TCP connections. The problem (20) can be solved using Algorithm 1 by replacing $U_s(\mathbf{1}^T\mathbf{x}_s)$

with $NU_s(\mathbf{1}^T\mathbf{x}_s)$. Algorithm 2 can be modified in a similar manner when all the TCP connections are file transfer applications.

When there are TCP connections of both types of applications, the path selection algorithm is not ready to generalize. However, we can still run Algorithm 1 (or Algorithm 2) assuming all of them are realtime (or file transfer) applications. The simulation results say that Algorithm 2 selects better paths most of the time. For instance, there are two concurrent TCP connections: one video call of duration 5 mins and one file transfer of size 125Mb, from a smartphone with both 4G and WiFi interfaces as in Fig. 1. For small α_s, both paths will be selected and there is no difference between Algorithm 1 or Algorithm 2. For large α_s, WiFi path is optimal for realtime application while 4G path is optimal for file transfer application. Then we run the congestion control algorithms on those paths individually. The simulation results are shown in Table 3. By using 4G path selected by Algorithm 2, we can save more energy and have larger throughput than the WiFi path selected by Algorithm 1.

6. CONCLUSION

In this paper, we design MP-TCP algorithms by jointly considering both throughput and energy consumption for mobile devices. We consider two types of applications: realtime applications, such as video streaming, and file transfer applications, such as file download/upload. We develop energy efficient MP-TCP algorithms for both of them with theoretical performance guarantee. Our preliminary simulation results show that the algorithms can indeed shift traffic to energy efficient path. More specifically, we can reduce energy consumption up to 22% without affecting throughput as compared to baseline MP-TCP algorithm.

7. REFERENCES

[1] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] E. Brosh, S. A. Baset, V. Misra, D. Rubenstein, and H. Schulzrinne. The delay-friendliness of tcp for real-time traffic. *IEEE/ACM Transactions on Networking*, 18(5):1478–1491, 2010.

[3] M. Chen, S. C. Liew, Z. Shao, and C. Kai. Markov approximation for combinatorial network optimization. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2010.

[4] Y.-C. Chen, E. M. Nahum, R. J. Gibbens, and D. Towsley. Measuring cellular networks: Characterizing 3g, 4g, and path diversity. Technical report, UMass Amherst Technical Report: UM-CS-2012-022.

[5] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural guidelines for multipath tcp development. *RFC6182 (March 2011), www. ietf. ort/rfc/6182*, 2011.

[6] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda. Multipath congestion control for shared bottleneck. In *Proc. PFLDNeT workshop*, 2009.

[7] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of ACM MobiSys*, pages 225–238, 2012.

[8] F. P. Kelly, A. K. Maulloo, and D. K. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[9] P. Key, L. Massoulié, and P. Towsley. Path selection and multipath congestion control. In *Proceedings of IEEE INFOCOM 2007*, pages 143–151, 2007.

[10] S. H. Low. A duality model of tcp and queue management algorithms. *IEEE/ACM Transactions on Networking (TON)*, 11(4):525–536, 2003.

[11] S. H. Low and D. E. Lapsley. Optimization flow control-i: basic algorithm and convergence. *IEEE/ACM Transactions on Networking (TON)*, 7(6):861–874, 1999.

[12] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking (ToN)*, 8(5):556–567, 2000.

[13] A. Pathak, Y. C. Hu, and M. Zhang. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In *Proceedings of ACM european conference on Computer Systems*, pages 29–42, 2012.

[14] Q. Peng, A. Walid, J. Hwang, and S. H. Low. Multipath tcp: Analysis design and implementation. *arXiv preprint arXiv:1308.3119*, 2013.

[15] G. P. Perrucci, F. H. Fitzek, and J. Widmer. Survey on energy consumption entities on the smartphone platform. In *IEEE Vehicular Technology Conference (VTC Spring)*, pages 1–6, 2011.

[16] C. Pluntke, L. Eggert, and N. Kiukkonen. Saving mobile device energy with multipath tcp. In *Proceedings of the international workshop on ACM MobiArch*, pages 1–6, 2011.

[17] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley. Opportunistic mobility with multipath tcp. In *Proceedings of the international workshop on ACM MobiArch*, pages 7–12, 2011.

[18] B. Wang, W. Wei, Z. Guo, and D. Towsley. Multipath live streaming via tcp: scheme, performance and benefits. *ACM Transactions on TOMCCAP*, 5(3):25, 2009.

[19] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *Proceedings of USENIX conference on Networked systems design and implementation*, pages 8–8, 2011.

8. ACKNOWLEDGEMENTS

APPENDIX

A. PROOF OF THEOREMS

A.1 Proof of Theorem 3.1

To characterize the sub-optimality gap of Algorithm 1, we introduce the following optimization problem, which can be regarded as a convex relaxation of PSP-I.
Relaxation of Path Selection Problem-I(RPSP-I):

$$\max \quad U_s(\mathbf{1}^T\mathbf{x}_s) - \alpha_s \sum_{r \in s} b'_r x_r \qquad (21a)$$

$$\text{s.t.} \quad x_r \le c_r \text{ for } r \in s \qquad (21b)$$

RPSP-I is a convex problem obtained by relaxing the non-continuous $P_r(x_r)$ to $b'_r x_r$. It is hard to evaluate the optimal solution to PSP-I due to its nonconvexity, but we can obtain an upper bound of its optimal value through RPSP-I. Then we study the gap between this upper bound and the objective value given by Algorithm 1.

Lemma A.1. *RPSP-I provides an upper bound for PSP-I. Moreover, the solution of RPSP-I $\mathbf{x}_s^*$ is optimal to PSP-I if and only if x_r^* is either 0 or c_r for all $r \in s$.*

PROOF. For every feasible $\mathbf{x}_s$, the objective value of RPSP-I is no less than PSP-I, which follows from

$$b'_r x_r \le b_r x_r + \theta_r 1_{\{x_r > 0\}} = P_r(x_r) \qquad (22)$$

for any $r \in s$. Since equality can be obtained if and only if x_r is either 0 or c_r in (22), the relaxation provides an exact

solution to PSP-I if and only if the solution to RPSP-I x_r^* is either 0 or c_r for each $r \in s$. $\square$

Next, we provide a close form solution to RPSP-I in Lemma A.2.

Lemma A.2. *Suppose C1 holds. Then $\hat{\mathbf{x}}_s$ (defined in (12)) is an optimal solution to RPSP-I. Moreover,*

$$
\hat{x}_i = \begin{cases} c_i & i < \hat{k} \\ (U_s')^{-1}(\alpha_s b_{\hat{k}}') - \sum_{i=1}^{\hat{k}-1} c_i & i = \hat{k} \\ 0 & i > \hat{k} \end{cases} \tag{23}
$$

and $\hat{k} := \max\{k \mid \sum_{i=1}^{k-1} c_i \leq (U_s')^{-1}(\alpha_s b_{\hat{k}}')\}$

PROOF. The Lagrangian of RPSP-I can be written as

$$
\begin{aligned}
L(\mathbf{x}, \mu) &= U_s\left(\sum_{i=1}^{|s|} x_i\right) - \alpha_s \sum_{i=1}^{|s|} b_i' x_i - \sum_{i=1}^{|s|} \mu_i(x_i - c_i) \\
&= U_s\left(\sum_{i=1}^{|s|} x_i\right) - \sum_{i=1}^{|s|}(\alpha_s b_i' + \mu_i) x_r + \sum_{i=1}^{|s|} \mu_i c_i,
\end{aligned}
$$

where $\mu_i \geq 0$ is the dual variable corresponding to each constraint $x_i \leq c_i$. By C1, $U_s(x)$ is strictly concave and differentiable , the KKT conditions imply that, at optimality

$$
\left[U_s'\left(\sum_{i=1}^{|s|} x_i\right) - (\alpha_s b_k' + \mu_k) \right]_{x_k}^+ = 0 \tag{24}
$$

$$
[c_k - x_k]_{\mu_k}^+ = 0 \tag{25}
$$

since RPSP-I is a convex problem. Now we will show $\hat{x}_k$ together with some $\hat{\mu}_k$ satisfy (24) and (25), meaning that $\hat{x}_k$ is indeed an optimal solution. Denote

$$
L_k := (U_s')^{-1}(\alpha_s b_k') - \sum_{1 \leq i < k} c_i,
$$

which is a monotone decreasing sequence and $L_{i+1} - L_i \leq -c_i$. By definition of $\hat{k}$, $0 \leq L_{\hat{k}} \leq c_{\hat{k}}$ and

$$
L_k \geq \sum_{i=k}^{\hat{k}-1} c_i \geq c_k \text{ for } k < \hat{k}
$$

$$
L_k \leq L_{\hat{k}} - \sum_{i=\hat{k}}^{k} c_i < 0 \text{ for } k > \hat{k},
$$

from which we can obtain (23). In addition, define

$$
\hat{\mu}_k := \left[U_s'\left(\sum_{i=1}^{k} \hat{x}_i\right) - \alpha_s \sum_{i=1}^{k} b_i' \right]_0^+
$$

and clearly that $\hat{x}_k$ together with $\hat{\mu}_k$ satisfy (24) and (25), which means $\hat{x}_k$ is the optimal solution to RPSP-I. $\square$

By Lemma A.1, the optimal solution of PSP-I O^* is upper bounded by the optimal value of RPSP-I. By Lemma A.2, $\hat{\mathbf{x}}_s$ is the optimal solution to RPSP-I, thus

$$
\begin{aligned}
O^* &\leq U_s(\mathbf{1}^T \hat{\mathbf{x}}_s) - \alpha_s \sum_{r \in s} b_r' \hat{x}_r \tag{26} \\
&= U_s\left(\sum_{i=1}^{\hat{k}-1} c_i + \hat{x}_{\hat{k}}\right) - \alpha_s \left(\sum_{i=1}^{\hat{k}-1} b_i' c_i + b_{\hat{k}}' \hat{x}_{\hat{k}}\right),
\end{aligned}
$$

Using the upper bound of O^* in (26), we have

$$
\begin{aligned}
O^* &- O_{\hat{k}} \tag{27} \\
&= \alpha_s \theta_{\hat{k}} - \alpha_s(b_{\hat{k}}' - b_{\hat{k}})\hat{x}_{\hat{k}} - \int_{\hat{c}+\hat{x}_{\hat{k}}}^{\hat{c}+\hat{x}_{\hat{k}}} (U_s'(y) - \alpha_s b_{\hat{k}}) dy \\
&\leq \alpha_s \theta_{\hat{k}} - \alpha_s(b_{\hat{k}}' - b_{\hat{k}})\hat{x}_{\hat{k}} \\
&= \alpha_s \theta_{\hat{k}}\left(1 - \frac{\hat{x}_{\hat{k}}}{c_{\hat{k}}}\right)
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
O^* - O_{\hat{k}-1} &= \int_{\hat{c}}^{\hat{c}+\hat{x}_{\hat{k}}} (U_s'(y) - \alpha_s b_{\hat{k}}') dy \tag{28} \\
&\leq \frac{1}{2} \hat{x}_{\hat{k}}(U_s'(\hat{c}) - \alpha_s b_{\hat{k}}')
\end{aligned}
$$

Clearly that $O_i \leq O_{\hat{k}-1}$ for $i \leq \hat{k} - 1$ and $O_i \leq O_{\hat{k}}$ for $i \geq k$. Thus by definition of k^*, we have $k^* \in \{\hat{k}, \hat{k}-1\}$, which means the objective value given by Algorithm 1: $\hat{O} = \max\{O_{\hat{k}}, O_{\hat{k}-1}\}$. Then

$$
\begin{aligned}
O^* - \hat{O} &= O^* - \max\{O_{\hat{k}}, O_{\hat{k}-1}\} \\
&\leq \min\{O^* - O_{\hat{k}}, O^* - O_{\hat{k}-1}\}
\end{aligned}
$$

and we obtain the bound in the theorem by substituting (27) and (28) into the above inequality.

A.2 Proof of Theorem 4.1

To show that Algorithm 2 obtains the exact optimal solution to PSP-II, we introduce the following optimization problem, which can be seen as a relaxation of PSP-II.
Relaxation of Path Selection Problem-II (RPSP-II):

$$
\max \quad U_s(\mathbf{1}^T \mathbf{x}_s) - \alpha_s \frac{1}{\mathbf{1}^T \mathbf{x}_s} \sum_{r \in s} b_r' x_r \tag{29a}
$$

$$
\text{s.t.} \quad x_r \leq c_r \text{ for } r \in s \tag{29b}
$$

Lemma A.3. *RPSP-II provides an upper bound for PSP-II. Its solution $\mathbf{x}_s^*$ is also optimal to PSP-II if and only if x_r^* is either 0 or c_r for all $r \in s$.*

The proof can be carried out in a similar manner as Lemma A.1. Different from RPSP-I, which is a convex problem, RPSP-II itself is a nonconvex problem. However, KKT condition still serves as a necessary condition on the structure for optimal solution.

Lemma A.4. *Suppose C1 and C2 hold. Then there exists an optimal solution $\hat{\mathbf{x}}_s$ to RPSP-II satisfying*

$$
\hat{x}_i = \begin{cases} c_i & i \leq \tilde{k} \\ 0 & i > \tilde{k} \end{cases} \tag{30}
$$

for some $\tilde{k} \leq |s|$.

PROOF. Using similar manner as Lemma A.2, we can show that there exists an optimal solution $\hat{\mathbf{x}}_s$ to RPSP-II such that

$$
\hat{x}_i = \begin{cases} c_i & i < \tilde{k} \\ 0 & i > \tilde{k} \end{cases} \tag{31}
$$

for some $\tilde{k} \leq |s|$. Now we only need to show the last unknown entry in $\hat{\mathbf{x}}_s$: $\hat{x}_{\tilde{k}}$ is also either 0 or $c_{\tilde{k}}$. Note that $\hat{x}_i$ is

a constant for any $i \neq \tilde{k}$ by (31). Then RPSP-I becomes a single variable optimization problem as follows:

$$\max \quad h(x_{\tilde{k}}) \quad s.t. \quad x_{\tilde{k}} \in [0, c_{\tilde{k}}],$$

where

$$h(x_{\tilde{k}}) := U_s(\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}}) - \alpha_s \frac{\sum_{i=1}^{\tilde{k}-1} b_i' c_i + b_{\tilde{k}}' x_{\tilde{k}}}{\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}}}.$$

The derivate of $h(x_{\tilde{k}})$ is given as

$$
\begin{aligned}
h'(x_{\tilde{k}}) &= U_s'(\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}}) + \alpha_s \frac{\sum_{i=1}^{\tilde{k}-1}(b_i' - b_{\tilde{k}}') c_i}{(\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}})^2} \\
&= \frac{g(\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}})}{(\sum_{i=1}^{\tilde{k}-1} c_i + x_{\tilde{k}})^2},
\end{aligned}
$$

where $g(x) := x^2 U_s'(x) + \alpha_s \sum_{i=1}^{\tilde{k}-1}(b_i' - b_{\tilde{k}}') c_i$. By C2, $g(x)$ is a nondecreasing function, based on which we know $h(x_{\tilde{k}})$ achieves its maxima at the boundary, i.e. $h(x_{\tilde{k}})$ achieves its maxima at 0 or $c_{\tilde{k}}$. $\square$

In Algorithm 2, it compares all the solution with the structure in (30), thus obtains the optimal solution to RPSP-II according to Lemma A.4. Then it is also the optimal solution to the PSP-II based on Lemma A.3 because x_r^* is either 0 or c_r for all $r \in s$.

A.3   Proof of Theorem 4.2

In [14], it is proved that the system (1)-(2) is stable if C1-C3 are satisfied. For each source s, let $\Phi_s(\mathbf{x}_s) := (\phi_r(\mathbf{x}_s), r \in s)$, then C1-C3 are stated below.

C1: For each $s \in \mathcal{S}$ there exists a nonnegative solution $\mathbf{x}_s := \mathbf{x}_s(\mathbf{p})$ to $\dot{x}_r = 0$ in (1) for any $\mathbf{p} \geq 0$. Moreover,

$$\frac{\partial y_l^s(\mathbf{p})}{\partial p_l} \leq 0, \qquad \lim_{p_l \to \infty} y_l^s(\mathbf{p}) = 0$$

where $y_l^s(\mathbf{p}) := \sum_{r \in s} H_{lr} x_r(\mathbf{p})$ is the aggregate traffic at link l from source s.

C2: For each $s \in S$ and each $\mathbf{x}_s$, $\Phi_s(\mathbf{x}_s)$ is continuously differentiable; moreover the symmetric part of the Jacobian $[\frac{\partial \Phi_s(\mathbf{x}_s)}{\partial \mathbf{x}_s}]^+ := \frac{1}{2}(\frac{\partial \Phi_s(\mathbf{x}_s)}{\partial \mathbf{x}_s} + (\frac{\partial \Phi_s(\mathbf{x}_s)}{\partial \mathbf{x}_s})^T)$ is negative definite.

C3: For each $r \in R$, $\phi_r(\mathbf{x}_s) = \infty$ if and only if $x_r = 0$.

Clearly that the $\phi_r(\mathbf{x}_s)$ defined in (18) satisfies C1 and C3 and we are left to show C2.

Lemma A.5. Let $E_s(\mathbf{x}_s) = \frac{\sum_{r \in s} P_r(x_r)}{\mathbf{1}^T \mathbf{x}_s}$, then

$$\mathbf{z}_s^T \frac{\partial^2 E_s(\mathbf{x}_s)}{\partial \mathbf{x}_s^2} \mathbf{z}_s \geq \frac{M^2}{(\mathbf{1}^T \mathbf{x}_s)^2} (\min_{r \in s} b_r - \max_{r \in s} b_r) \qquad (32)$$

for any $\mathbf{z}_s \in \mathbb{R}^{|s|}$ with $\sum_{r \in s} z_r = M$, where M is any constant.

PROOF. Let $\Lambda_b := diag\{b_r, r \in s\}$, then

$$
\begin{aligned}
& \frac{\partial^2 E_s(\mathbf{x}_s)}{\partial \mathbf{x}_s^2} \\
&= \frac{2 \sum_{r \in s}(\theta_r + b_r x_r)}{\|\mathbf{x}_s\|_1^3} \mathbf{1} \mathbf{1}^T - \frac{1}{\|\mathbf{x}_s\|_1^2}(\Lambda_b \mathbf{1} \mathbf{1}^T + \mathbf{1} \mathbf{1}^T \Lambda_b)
\end{aligned}
$$

which means

$$
\begin{aligned}
\mathbf{z}_s^T \frac{\partial^2 E_s(\mathbf{x}_s)}{\partial \mathbf{x}_s^2} \mathbf{z}_s &= \mathbf{z}_s^T A_s(\mathbf{x}_s) \mathbf{z}_s + \frac{2 \sum_{r \in s} \theta_r}{\|\mathbf{x}_s\|_1^3}(\mathbf{1}^T \mathbf{z}_s)^2 \\
&\geq \mathbf{z}_s^T A_s(\mathbf{x}_s) \mathbf{z}_s
\end{aligned}
$$

where

$$A_s(\mathbf{x}_s) := \frac{2 \sum_{r \in s} b_r x_r}{\|\mathbf{x}_s\|_1^3} \mathbf{1} \mathbf{1}^T - \frac{1}{\|\mathbf{x}_s\|_1^2}(\Lambda_b \mathbf{1} \mathbf{1}^T + \mathbf{1} \mathbf{1}^T \Lambda_b).$$

And

$$
\begin{aligned}
& \mathbf{z}_s^T A(\mathbf{x}_s) \mathbf{z}_s \qquad\qquad\qquad\qquad\qquad\qquad (33) \\
&= \frac{1}{(\mathbf{1}^T \mathbf{x}_s)^2} \left((\sum_{r \in s} b_r \frac{x_r}{\mathbf{1}^T \mathbf{x}_s})(\sum_{r \in s} z_r)^2 - (\sum_{r \in s} b_r z_r)(\sum_{r \in s} z_r) \right) \\
&= \frac{M}{(\mathbf{1}^T \mathbf{x}_s)^2} \left(\sum_{r \in s} b_r(M \frac{x_r}{\mathbf{1}^T \mathbf{x}_s} - z_r) \right)
\end{aligned}
$$

Without loss of generality, assume $b_i \leq b_j$ for $i < j$. Let $a_r := M \frac{x_r}{\mathbf{1}^T \mathbf{x}_s}$, then

$$
\begin{aligned}
& \sum_{r \in s} b_r(a_r - z_r) - M(b_1 - \beta_{|s|}) \\
&= \sum_{r \in s} b_r(a_r - z_r) - (b_1 \sum_{r \in s} a_r - b_{|s|} \sum_{r \in s} z_r) \\
&= \sum_{r \in s} a_r(b_r - b_1) + \sum_{r \in s} z_r(b_{|s|} - b_r) \geq 0
\end{aligned}
$$

which means

$$\sum_{r \in s} b_r(M \frac{x_r}{\mathbf{1}^T \mathbf{x}_s} - z_r) \geq M(b_1 - \beta_{|s|})$$

and substitute the above inequality into (33), we can obtain (32). $\square$

Lemma A.6. The $\phi_r(\mathbf{x}_s)$ defined in (18) satisfies C2 if (19) holds.

PROOF. The Jacobian $\partial \Phi_s(\mathbf{x}_s)/\partial \mathbf{x}_s$ can be written as:

$$-\frac{4(1-\beta)}{\tau^2 \|\mathbf{x}_s\|_1^3} \mathbf{1} \mathbf{1}^T - \beta \frac{2\|\mathbf{x}_s\|_n}{\tau^2 \|\mathbf{x}_s\|_1^2} \Lambda_s^{-1} \left(I_{|s|} + \mathbf{1} \mathbf{a}_s^T \right) \Lambda_s^{-1} - \frac{\partial E_s(\mathbf{x}_s)}{\partial \mathbf{x}_s}$$

where $\Lambda_s = diag\{\mathbf{x}_s\}$ and $\mathbf{a}_s = \left(\frac{2x_r}{\|\mathbf{x}_s\|_1} - \frac{x_r^n}{\|\mathbf{x}_s\|_n^n}, r \in s \right)$. In [14], it is shown in Theorem 4.1 that for $\sum_{r \in s} z_r = M$,

$$\mathbf{z}_s^T (I_{|s|} + \mathbf{1} \mathbf{a}_s^T) \mathbf{z}_s \geq \frac{M^2}{4} \left(\frac{9}{|s|} - 1 \right),$$

which means if $\sum_{r \in s} z_r = M$, we have

$$\mathbf{z}_s^T \Lambda_s^{-1} (I_{|s|} + \mathbf{1} \mathbf{a}_s^T) \Lambda_s^{-1} \mathbf{z}_s \geq \frac{M^2}{4\|\mathbf{x}_s\|_\infty^2} \left(\frac{9}{|s|} - 1 \right),$$

Based on the above inequality and Lemma A.5, for any $\mathbf{z}_s$ with $\sum_{r \in s} z_r = M$, we have

$$
\begin{aligned}
& \mathbf{z}_s^T \frac{\partial \Phi_s(\mathbf{x}_s)}{\partial \mathbf{x}_s} \mathbf{z}_s \\
& \leq \frac{-\beta M^2 \|\mathbf{x}_s\|_n}{2\tau^2 (\mathbf{1}^T \mathbf{x}_s)^2 \|\mathbf{x}_s\|_\infty^2} \left(\frac{9}{|s|} - 1 \right) + \frac{\alpha_s M^2}{(\mathbf{1}^T \mathbf{x}_s)^2}(\min_{r \in s} b_r - \max_{r \in s} b_r) \\
& \leq 0
\end{aligned}
$$

if (19) is satisfied. $\square$

Therefore, the $\phi_r(\mathbf{x}_s)$ satisfies C1-C3 provided (19) is satisfied and the corresponding system (1)-(2) is asymptotically stable.