A Rent-Seeking Framework for Multipath TCP

Shiva Raj Pokhrel
shiva.pokhrel@deakin.edu.au
School of Information Technology, Deakin University
Geelong, Victoria, Australia

Carey Williamson
carey@cpsc.ucalgary.ca
Department of Computer Science, University of Calgary
Calgary, Alberta, Canada

ABSTRACT

Network utility maximization (NUM) for Multipath TCP (MPTCP) is a challenging task, since there is no well-defined utility function for MPTCP [6]. In this paper, we identify the conditions under which we can use Kelly's NUM mechanism, and explicitly compute the equilibrium. We obtain this equilibrium by using Tullock's *rent-seeking* framework from game theory to define a utility function for MPTCP. This approach allows us to design MPTCP algorithms with common delay and/or loss constraints at the subflow level. Furthermore, this utility function has diagonal strict concavity, which guarantees a globally unique (normalized) equilibrium.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; • **Networks** → Network reliability.

KEYWORDS

Multipath TCP Tullock Rent Seeking Congestion Control Normalized Equilibrium

ACM Reference Format:
Shiva Raj Pokhrel and Carey Williamson. 2020. A Rent-Seeking Framework for Multipath TCP. In *IFIP Peformance '20, November 02–06, 2020, Politecnico di Milano, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/draft

1 INTRODUCTION

The ongoing development of 5G and Gigabit WiFi solutions will considerably improve throughput and delay performance on future wireless networks. These developments offer new possibilities to enhance existing applications by using Multipath TCP (MPTCP) to exploit multiple available network interfaces.

A challenging design issue that arises is how best to use WiFi and non-WiFi 5G seamlessly [10] across a variety of applications and platforms. This is a challenge since different applications impose different requirements and constraints for successful MPTCP usage.

One example application is video streaming. This seems like an ideal use case for MPTCP, since it has been difficult to scale existing services to meet the growing traffic demands of mobile users. For some existing applications, such as online video conferencing, high resolution video still suffers from jitter, unwanted glitches, and frozen screens when used over mobile networks.

Augmented Reality (AR) is another emerging application that faces similar challenges due to bandwidth demands. AR allows users with nomadic wireless devices to seamlessly access information, by overlaying graphics and digital information upon their perception of the physical world. This enables immersive AR/VR applications, but requires dependable, low latency, and high-bandwidth data communication.

Vehicular networking is yet another application that could benefit from MPTCP. For example, autonomous vehicles may need to send 3D views of their environment to remote monitoring servers under extremely tight deadlines to ensure safe operation. Similarly, unmanned aerial vehicles such as delivery drones have stringent communication requirements for navigating their dynamically changing environment in a cooperative fashion. One example of this would be using live video from multiple drones to anticipate obstacles ahead, and avoid risky blind spots while on the move.

These aforementioned applications have three key similarities. First, they are throughput-intensive. Second, they require low end-to-end delay. Third, they need reliable data packet transport.

Although the mobile/wireless networking standards are evolving to support higher capacity requirements, existing transport-layer protocols (e.g., TCP, UDP, QUIC, MPTCP) rarely provide adequate support for low latency and high throughput applications, especially in wireless networks. One oft-cited reason for this is the tendency of TCP congestion control to induce excessive buffering delay at the network bottleneck. Further compounding this problem are

IFIP Performance '20, November 02–06, 2020, Politecnico di Milano, Italy

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/draft

the vagaries of wireless channels, which often exhibit non-deterministic behaviour in signal quality and throughput at short time scales.

Motivation and Contributions

Kelly *et al.* [3] designed a distributed congestion control framework for obtaining a network equilibrium with an objective of achieving *proportional fairness*. This model holds as long as the TCP sources react to the sum of the congestion prices in their paths. Moreover, when the TCP sources react to the maximum of the prices in the network paths, the system instead achieves *max-min fairness*.

Our main insight is that Kelly's approach of "*transforming the fairness problem into a game between TCP users*" shares several features with the Tullock *rent-seeking* game [13]. The main differences are discussed later.

In coupled MPTCP algorithms, it is known that, if one path fails to meet the loss and delay constraints, then the subflows on other paths suffer [5, 8, 9]. Therefore, MPTCP subflows contests can be considered as a game with common coupled constraints. In games with common coupled constraints, when a player fails to achieve the constraints common to other players, all other players will also be penalized. Such games can therefore be viewed as cooperative in the objectives of meeting common constraints and non-cooperative in the context of utilities. We exploit the *rent-seeking* game framework for studying the contests between subflows, thus providing insights in designing *cooperative MPTCP* algorithms [11].

Our idea is to find a *pseudo-price* that is globally constant per unit resource across all network paths. Recall that such an equilibrium with a global fixed *pseudo-price* is the desired *normalized* equilibrium for our MPTCP dynamics over the multipath network. Therefore, the Tullock game framework explained recently in [1] is used to redesign MPTCP.

To summarize, the main contributions of our work are:

- We design a utility function for MPTCP based on Tullock's rent-seeking game.
- We provide extensive experimental results showing the performance of our MPTCP algorithm.

The rest of this paper is organized as follows. Section 2 provides an overview of our MPTCP modeling approach, and situates it in the context of prior work. Section 3 discusses the details of our approach, including the game theory framework, and the equilibrium properties of our system. Section 4 presents experimental results evaluating the performance of our MPTCP solution using network emulation experiments. Section 5 concludes the paper.

2 MODEL OVERVIEW

This section provides the background context for our work, by summarizing prior work on network optimization and MPTCP design. The section concludes with a conceptual overview of our MPTCP modeling approach using Tullock's rent-seeking framework (RSF).

Background and Related Work

Kelly's theory on *network utility maximization* (NUM) [3] provides a rigorous foundation for the design of efficient TCP algorithms. It has proven useful for understanding the structural properties of networks under end-to-end congestion control, and for designing optimal TCP algorithms. For path selection in multipath congestion control, Key *et al.* [4] developed another novel approach to study the potential benefits of coordinated congestion control. It enables users to select (or reselect) routes by adopting the notion of a Nash equilibrium.

There are two distinct steps in most existing optimization frameworks for TCP congestion control design. First, one develops a mathematical model for the resource allocation problem (e.g., fluid and/or queueing model [6]). Second, one designs an algorithm to solve the model (e.g., convex optimization based on its properties) [7].

Existing MPTCP congestion control algorithms [5, 6, 8, 10] have used common coupled constraints (e.g., loss/delay) for representing the channel characteristics and congestion conditions of the network paths (see Table 1 later for a concise summary of the TCP window update operations for LIA, OLIA, and BALIA). If one path fails to meet the constraints, then the subflows on other paths may also suffer [5, 8, 9].

Moreover, the throughput achieved by a subflow depends in subtle ways upon the throughputs of all other subflows belonging to the same MPTCP connection [2, 8, 10]. When packets are sent via diverse wireless paths, they can arrive at the receiver out-of-order, or be lost due to channel impairments. Lost packets must be retransmitted, and out-of-order packets must be buffered for resequencing, so that they can be delivered to the application in proper order. Both of these artifacts affect subflow throughput, because of buffering delays, and the need for retransmissions at either the data-link layer or the MPTCP level [2].

At a conceptual level, almost all coupled MPTCP congestion control designs focus on minimizing reordering delay and Head of Line (HOL) blocking. They provide both load balancing and congestion control by coupling the delays and losses across all subflows. Based on our understanding of the design space, we propose a new MPTCP algorithm that generalizes existing algorithms, and strikes a good balance

Table 1: Summary of MPTCP Congestion Control for LIA, OLIA, and BALIA

Algorithm	α Parameter	Window Increase on path i	Window Decrease on path i
LIA [10]	$\alpha = \frac{\max\left(W_i/\tau_i^2\right)}{\left(\sum_i \frac{W_i}{\tau_i}\right)^2} \sum_i W_i$	For each acknowledgment, $\Delta(W_i) = \min\left(\frac{1}{W_i}, \frac{\alpha}{\sum_i W_i}\right)$	For each packet loss, $W_i \leftarrow \frac{W_i}{2}$ same as regular TCP
OLIA [5]	$\alpha_i = \begin{cases} \frac{1/\lvert\mathcal{I}\rvert}{\lvert C\rvert}, & i \in C \\ \frac{1/\lvert\mathcal{I}\rvert}{\lvert\mathcal{X}\rvert} & i \in \mathcal{X}, \lvert C\rvert > 0 \\ 0 & \text{otherwise} \end{cases}$	For each acknowledgment, $\Delta(W_i) = \left(\frac{\alpha_i}{W_i} + \frac{W_i/\tau_i^2}{\left(\sum_i \frac{W_i}{\tau_i}\right)^2}\right)$	For each packet loss, $W_i \leftarrow \frac{W_i}{2}$ same as regular TCP
BALIA [6]	$\alpha_i = \frac{\max\{\lambda_k\}}{\lambda_i},\ \lambda_i = \frac{W_i}{\tau_i},\ \lambda_k = \left(\sum_{k\in\mathcal{I}} \frac{W_k}{\tau_k}\right)^2$	For each acknowledgment, $\Delta(W_i) = \frac{\lambda_i}{\tau_i(\sum\lambda_k^2)} \frac{1+\alpha_i}{2} \frac{4+\alpha_i}{5}$	For each packet loss, $W_i \leftarrow W_i - \frac{W_i}{2}\min\left(\alpha_i, \frac{3}{2}\right)$

among these properties. We have implemented this algorithm in the Linux kernel, and we evaluate its performance experimentally using our prototype.

Our Proposed Scheme

We now summarize our proposed MPTCP algorithm. The key idea is to schedule the packets via multiple paths at different rates in such a way that they balance the delay across all paths, and are therefore more likely to arrive at the receiver in-order. More specifically, an MPTCP congestion control mechanism with heterogeneous network paths will perceive different utility on each path.

In essence, we strive to harmonize the performance achieved across multiple paths, by opportunistically exploiting path dynamics to provide seamless MPTCP performance. The primary advantage of our method is that it implicitly addresses all known issues that limit MPTCP performance (i.e., HOL blocking, reordering latency, route heterogeneity). Furthermore, it enables the pooling of multipath resources in a way that emulates the performance of a single harmonized path.

One way to achieve utility maximization is to formulate a Tullock rent-seeking game [13, 14], and use this to equalize the delays across all paths in order to ameliorate reordering delay. A plausible approach is to set equal delay for all subflows (across all paths), and determine the corresponding sending rates that jointly satisfy the delay constraints and load balancing objectives.

Delay-Balanced MPTCP (RSF-DB): In RSF-DB, each source has a set of paths and a predefined target delay $d > 0$, which is fixed and equal for all paths. The source maintains a congestion window W_i for each path i, and measures its forward delay q_i and round trip time τ_i for each path. Here, q_i is the queueing plus service delay from the source to the sink on path i, and N is the number of paths. Then the window adaptation in the congestion avoidance phase is:

- For each packet loss on path i, update $W_i \longleftarrow W_i/2$;
- For each new acknowledgment received on path i, increase W_i using $W_i \longleftarrow W_i + \Delta(W_i)$, where:

$$\Delta(W_i) = \frac{1}{W_i} \min\left\{1, \frac{d - q_i}{q_i}\right\}. \tag{1}$$

This algorithm is designed to adjust the window size dynamically so that q_i converges toward the target delay d in steady-state. Note that q_i is not necessarily strictly bounded by d, so that the "window increase" could in some cases be negative, if the situation warrants.

A concise summary of the window update operations for Well-known MPTCP algorithms, LIA, OLIA, and BALIA is shown in Table 1.

Similarly, we design a complementary solution called Loss-Balanced MPTCP (RSF-LB), in which each MPTCP source has a set of paths and predefined target loss rate. The key idea here is to balance the loss *rate* (i.e., lost packets per second) on each path, to ameliorate the potential effects of heterogeneity in the packet loss *probabilities*, l_i, on different paths. For RSF-LB, we replace (1) by

$$\Delta(W_i) = \frac{1}{W_i} \min\left\{1, \frac{\mu - \lambda_i l_i}{\lambda_i l_i}\right\} \tag{2}$$

so that the instantaneous $\lambda_i l_i$ converges toward the target loss rate μ packets/sec in steady-state.

We extend Kelly's framework [3] by exploiting the well-known Tullock *rent-seeking* game [13] to design a cooperative MPTCP with common coupled constraints. We consider a non-linear dynamical system model for an MPTCP connection over K network paths, as seen in Equation (3):

$$\dot{y} = g(y), \quad y(0) = y_0, \tag{3}$$

where $g(y) = 0$ has a unique solution. In this equation,

$$y = (\lambda, p);\ \lambda = (\lambda_1, \lambda_2, \dots \lambda_K)$$

the λ_i values denote the packet sending rates, and

$$p = (p_1, \dots p_K)$$

denote the loss l_i or delay q_i, the congestion prices, for the respective paths. Then, motivated by the generalized MPTCP analysis in [6], we specify the following fluid model:

$$\dot{\lambda}_i = k_i(\lambda)(\phi_i(\lambda) - p)^+_{\lambda_i};\ \ \dot{p}_i = \gamma_i(\lambda_i - \theta_i)^+_{p_i} \tag{4}$$

for analyzing MPTCP [6, (3)-(4)]. Here, $\gamma_i > 0$ is a positive gain, $k_i(\lambda)$ is a vector of positive gains determining the dynamic properties of the system, and $\phi_i(\lambda)$ determines the equilibrium properties.

3 MODEL DETAILS

This section provides further details on our MPTCP model, including the Tullock rent-seeking game and its equilibrium properties.

Rent-Seeking Game Framework

Rent-seeking is a concept from game theory in economics. Unlike profit-seeking, which tries to increase total wealth, rent-seeking tries to increase one's own share of existing wealth, without increasing the total wealth. Specifically, it is an attempt to obtain a reward R (rent) by manipulating the environment in which the activities occur, rather than by creating new wealth. One example is lobbying and/or bribery of government officials, in which the efforts (e.g., time and/or money) spent may result in favourable outcomes (e.g., tax laws, corporate rebates), albeit with potentially reduced economic efficiency. Other examples include taxi-licensing fees, bridge tolls, and bidding in loss-load curves [15], all of which result in reallocation of resources.

Strategies for *rent-seeking* games were studied by Gordon Tullock in 1967, and form a good basis for studying MPTCP flows. We use this rent-seeking approach to design our utility function below.

We consider an MPTCP connection in a network setting with K paths, and one subflow on each path. Let $\lambda_i = W_i/\tau_i$ (strictly positive $\lambda_i > 0$) represent the sending rate for the i^{th} subflow ($1 \leq i \leq K$), and let p_i represent the corresponding price for that subflow. Then, assuming that the sending rates on each path are the actions to be chosen, and that connection throughput is concave, the individual throughput (payoff) obtained from path i can be represented as:

$$\theta_i = V\left(\lambda_i \Big/ \sum_{j=1}^{K} \lambda_j\right), \tag{5}$$

where $V(.)$ is a concave function. Observe that the throughput θ_i comes at the cost of $\hat{p}\lambda_i$ (where $\hat{p}$ is a constant). With this formulation, the utility for the MPTCP subflow over path i is structurally similar to that of a Tullock rent-seeking game [1], for which:

$$U_i(\lambda) = \lambda_i \Big/ \sum_{j=1}^{K} \lambda_j - \hat{p}\lambda_i. \tag{6}$$

In contrast to Kelly's approach [3], the subflows of an MPTCP connection can have an action on one path (such as an increase or decrease in λ_i) that is not dependent on the actions over other paths. Further, similar to [3], the capacity of the system is limited in real communication networks. Therefore, the aggregate network bandwidth can be bounded by a finite constant, say B. As a result,

$$\sum_{j=1}^{K} \lambda_j \leq B. \tag{7}$$

Equilibrium Properties

We now discuss the application of our new utility function from Equation (6) to the solution of our dynamic system defined in Equation (4).

Let $(\lambda^\star, p^\star)$ be the equilibrium of the aforementioned dynamical system and $\lambda^\star_{(-\mathrm{i})}$ be the action vectors of all the MPTCP subflows, except for the one along path i. Now, by using KKT, since $U_i(\lambda)$ is concave for each λ_i, there exists a Lagrange multiplier $L_i(\lambda^\star_{(-\mathrm{i})})$ such that $\lambda^\star_i$ maximizes the Lagrangian:

$$Q_i(\lambda_i) \quad = \quad U_i(\lambda, x_{(-\mathrm{i})}) - L_i(x_{(-\mathrm{i})})\Big(B - \sum_{j=1}^{K} \lambda_j\Big); \tag{8}$$

$$\text{and} \qquad L_i(x_{(-\mathrm{i})})\Big(B - \sum_{j=1}^{K} \lambda_j\Big) = 0. \tag{9}$$

Equation (9) is obtained by using the complementary property. Also, observe in Equation (8) that the Lagrangian $Q_i(.)$ is replacing the utility $U_i(.)$.

The Lagrange multipliers are interpreted as *pseudo-prices* of the paths. If a price is set on path i while other players are at equilibrium, then the subflow on path i pays $\lambda_i L_i(\lambda^\star_{(-\mathrm{i})})$ for its use of the bandwidth. In this case, $\lambda^\star$ is an equilibrium of our non-linear dynamical system with the capacity constraints. However, the pricing is not tractable since it may vary from one subflow to another (for similar use of the network resources). Also, it depends on the equilibrium selected.

Diagonal Strict Concavity

Consider a game with K subflows of an MPTCP connection. The strategy space is $S \subset \mathbb{R}$, where S is a bounded set, and subflow i's utility function is $U_i : S^K \rightarrow \mathbb{R}$. Rosen's condition [12] for uniqueness of the Nash equilibrium in a

K-player game states that the equilibrium is unique when: i) $U_i(\lambda)$ is concave in subflow i's own action; ii) there exists a vector z of non-negative numbers such that the function $\rho(\lambda, z) := \sum_{i=1}^{K} z_i U_i(\lambda)$ is diagonally strictly concave.

To define the concept of *diagonal strict concavity*, we first compute the pseudo-gradient of $\rho(.)$:

$$g(\lambda, z) = \begin{bmatrix} z_1 \frac{\partial U_1(\lambda_1, \lambda_{-1})}{\partial \lambda_1} \\ z_2 \frac{\partial U_2(\lambda_2, \lambda_{-2})}{\partial \lambda_1} \\ \vdots \\ z_K \frac{\partial U_K(\lambda_K, \lambda_{-K})}{\partial \lambda_1} \end{bmatrix} \tag{10}$$

where $\lambda_{-i} = \sum_{j=1(\neq i)}^{K} \lambda_j$. Then, the function $\rho(.)$ is said to be *diagonally strictly dominant* in $z \in S$ for fixed $z \leq 0$ if for every $\lambda_0, \lambda_1 \in S$

$$(\lambda_0 - \lambda_1)' g(\lambda_1, z) + (\lambda_1 - \lambda_0)' g(\lambda_0, z) > 0. \tag{11}$$

Condition C_0: A sufficient condition for the function $\rho(.)$ to be diagonally strictly concave is that the matrix $[G'(\lambda, z) + G(\lambda, z))]$ is negative definite for $z \in S$: here $G(\lambda, z)$ is the Jacobian of the pseudo-gradient $g(\lambda, z)$ with respect to λ.

Condition C_1: A sufficient condition for the existence of a unique global equilibrium is that $\rho(.)$ is diagonally strictly concave for some z (i.e., C_0 holds).

Multiple MPTCP Connections

Now, we extend our solution to the case of N MPTCP sources sharing K paths. Each subflow i of an MPTCP connection n over the K paths has sending rate λ_i^n and the following constraint should hold:

$$\sum_{n=1}^{N} \sum_{i=1}^{K} \lambda_i^n \leq B. \tag{12}$$

The throughput obtained by n^{th} MPTCP connection is the sum of the throughput by all subflows, and is given by:

$$\theta^n = \sum_{i=1}^{K} \theta_i^n(\lambda^n). \tag{13}$$

where λ^n is the vector $[\lambda_1^n, \lambda_2^n, \ldots, \lambda_K^n]$ and

$$\theta_i^n(\lambda^n) = V\left(\frac{\lambda_i^n}{\sum_{m=1}^{N} \sum_{j=1}^{K} \lambda_j^m}\right) \text{(recall Equation (5)).} \tag{14}$$

Therefore, the utility of the n^{th} MPTCP connection is:

$$U^n(\lambda) = \sum_{i=1}^{K} \left(\frac{\lambda_i^n}{\sum_{m=1}^{N} \sum_{j=1}^{K} \lambda_j^m} - \hat{p}\lambda_i^n\right). \tag{15}$$

In realistic network settings, each network path i has its own capacity (bandwidth), therefore let C be the vector with the i^{th} element being a finite constant C_i. Then, $B = \sum C_i$, and for each i, we have the following constraint:

$$\sum_{n=1}^{N} \lambda_i^n \leq C_i. \tag{16}$$

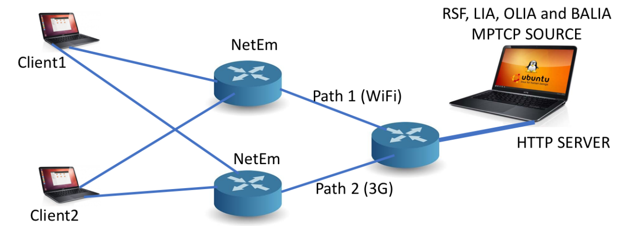

Figure 1: Experimental setup: we use Ethernet links and emulate the characteristics of WiFi and 3G paths.

4 EVALUATION

In this section, we present the performance evaluation of our new MPTCP algorithm in different network scenarios. We describe the network settings used in the experiments, and then present our experimental results for each scenario.

Experimental Setup

We have evaluated our *cooperative MPTCP* algorithm based on the Rent-Seeking Framework (RSF) by comparing it with three other well-known MPTCP algorithms: LIA [10], OLIA [5], and BALIA [6]. The latter three algorithms are all available in an experimental Linux implementation[1] of MPTCP v0.93 by Paasch *et al.* We used this Linux implementation as the basis for our experiments, and added implementations for our RSF MPTCP, including both Delay-Balanced (RSF-DB) and Loss-Balanced (RSF-LB) versions.

The test environment in our lab consists of three laptops, all running Ubuntu Linux 16.04. As shown in Fig. 1, one of these is set up as an HTTP server, while the other two are clients. The laptops are connected to an Ethernet switch, each with two 1 Gbps interfaces, thus creating two separate paths for our experiments. In particular, each of our MPTCP connections has two subflows in our experiments (for simplicity [6]). We used the NetEm[2] network emulator to control the bandwidth, delay, and packet loss on the Gigabit Ethernet links in order to emulate the characteristics of wireless paths. We have used $d = 100$ ms and $\lambda_i p = 4$ packets/sec in all of our experiments.

The performance evaluation of the *MPTCP* algorithms is done with the following settings. We have used five MPTCP connections (each with two subflows) between the server and two clients. The server runs a single MPTCP implementation

[1] C. Paasch *et al.*, "Multipath TCP in the Linux Kernel", available from https://www.multipath-tcp.org (released: 02-Nov-2017).
[2] NetEm: https://wiki.linuxfoundation.org/networking/netem

Table 2: Summary of Throughput Results from Network Emulation Experiments

ID	Emulated Network Scenario	Regular TCP	MPTCP Algorithm (Mbps) LIA	OLIA	BALIA	RSF	Comments and Observations
0	Single Path	1.58	1.53	1.57	1.58	1.58	RSF performs same as TCP & BALIA
1	Homogeneous Paths: No Loss	1.58	2.52	2.58	2.54	2.53	RSF performs same as LIA
2	Homogeneous Paths: 2% Loss	-	0.88	0.98	0.98	1.98	RSF has 2x higher throughput
3	Homogeneous Paths: 10% Loss	-	0.31	0.35	0.39	0.62	RSF has 60-100% higher throughput
4	Heterogeneous Paths: Bandwidth	-	1.88	1.88	1.98	2.89	RSF has 40-50 %higher throughput
5	Heterogeneous Paths: Delay	-	2.59	2.69	2.69	4.59	RSF has 70% higher throughput
6	Heterogeneous Paths: Loss	-	3.13	3.10	3.12	4.48	RSF has 40% higher throughput

at a time for each experiment. The packet traffic was generated by a large file transfer from the HTTP server, where the file size ranged from 2 MB to 10 MB. Packet traces are captured using *tcpdump*[3] and analyzed with *wireshark*[4].

Table 2 provides a concise summary of our results, which are presented and explained in the following subsections.

Baseline Results

For verification and validation of our experimental environment, we started with two baseline scenarios.

Scenario 0 used a single path network, for which all TCP versions should perform the same. The first row of results in Table 2 confirms this observation, which is summarized in the rightmost column of the table.

Scenario 1 used two homogeneous paths with no packet loss at all. In this scenario, our RSF MPTCP should perform the same as LIA. The next row of results in Table 2 confirms this observation, which is also summarized in the table. In fact, we observe the exact same results for both RSF-DB and RSF-LB MPTCP.

Homogeneous Paths

The next set of experiments considers homogeneous paths, but with non-zero packet loss. In these scenarios, RSF MPTCP starts to differentiate itself from the other MPTCP algorithms.

For Scenario 2, we set the capacity (10 Mbps), delay (150 ms), and Bernoulli packet loss probability (2%) for the links to be the same for both paths. We then studied the impacts on RSF-DB throughput when downloading files of different sizes ($2\ MB \leq size \leq 8\ MB$) with different MPTCP algorithms, each operating under the same settings.

Figure 2 and Figure 3 illustrate the throughput and fairness results, respectively. Figure 2 shows that when the size of the file is 4 *MB*, our proposed *RSF MPTCP* achieves almost double the throughput of LIA, OLIA, and BALIA. Figure 3 illustrates that when the size of the file is 4 *MB* the corresponding Jain's fairness index stays quite close to unity. Hence, the throughput is highly improved by using our design without any adverse impact on fairness.

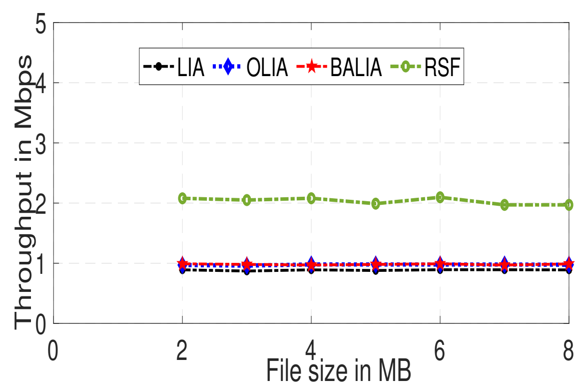

Figure 2: Throughput results for different file sizes in Scenario 2: Homogeneous Paths with 10 Mbps bandwidth, 150 ms delay, and 2% random packet loss.

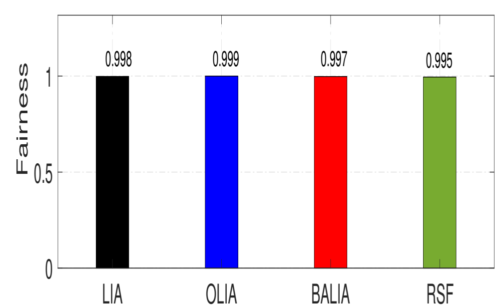

Figure 3: Fairness results for 4 MB files in Scenario 2: Homogeneous Paths with 10 Mbps bandwidth, 150 ms delay, and 2% random packet loss.

We also considered Scenario 3 with a much higher Bernoulli packet loss probability (10%) on both paths. While the throughput declines for all MPTCP algorithms, our RSF MPTCP still provides much higher throughput than the others.

[3]https://www.tcpdump.org
[4]https://www.wireshark.org

The results from Scenario 2 and Scenario 3 lead to the following observation:

- Observation 2/3. The proposed *RSF MPTCP* outperforms all other MPTCP algorithms in terms of throughput, and achieves similar fairness.

Heterogeneous Paths

The next set of experiments considers heterogeneous network paths.

Scenario 4 has heterogeneous paths that differ in their bandwidth. We set the delay (150 ms) and random packet loss probability (4%) to be the same for both paths. We set the bandwidth to a fixed value of 20 Mbps for one path, while varying the capacity of the other one from 10 Mbps to 80 Mbps. We then measured the RSF-DB performance when downloading a 10 MB file using different MPTCP algorithms.

Figure 4 shows the impact on performance. In general, the throughputs achieved by the existing MPTCP algorithms are fairly low, and remain almost constant even with increasing bandwidth. That is, they do not exploit the additional bandwidth well. In contrast, the RSF-DB algorithm shows consistent improvement in throughput under the same network settings. When packet losses are frequent, the existing MPTCP algorithms suffer a lot, but our algorithm is able to discover heterogeneous paths quickly and improve the load balancing.

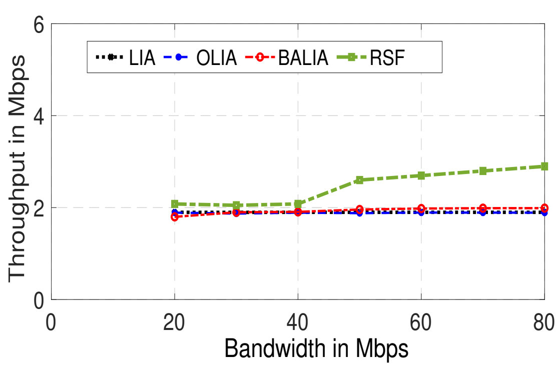

Figure 4: Throughput results for 10 MB file transfer in Scenario 4: Heterogeneous Bandwidth Paths. One path is 20 Mbps, while the bandwidth of the other one is varied. Both paths have 150 ms delay, and 4% random packet loss.

The results from Scenario 4 are summarized as follows:

- Observation 4. The proposed *RSF MPTCP* outperforms the other MPTCP algorithms when paths have heterogeneous bandwidth.

Next, Scenario 5 considers paths with heterogeneous delay characteristics. We set equal bandwidths for the two paths (20 Mbps), and the same random packet loss probability (3%) for both paths. We fix the delay for one of the paths at 20 ms, and then vary the delay of the other path from 20 ms to 100 ms. As previously, we observe the RSF-DB throughput when downloading a 10 MB file.

Figure 5 illustrates the overall impact on throughput, which can be summarized as follows:

- Observation 5. The throughput of all MPTCP algorithms (including RSF) decreases with increasing delay, as expected. However, the decline is quite gradual for RSF MPTCP, and quite pronounced for other MPTCP algorithms.

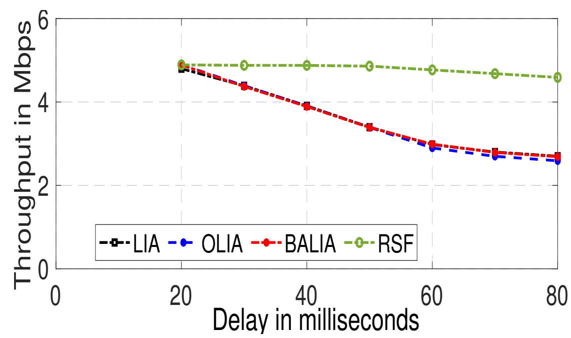

Figure 5: Throughput results for 10 MB file transfer in Scenario 5: Heterogeneous Delay Paths. One path has 20 ms delay, while the delay of the other one is varied. Both paths have 20 Mbps bandwidth, and 3% random packet loss.

This result is mainly due to the delay required for retransmission of lost packets, as well as the buffering delay when re-sequencing out-of-order packets received at the client. In contrast, the balanced pricing strategy adopted in our design makes RSF MPTCP superior to others in an environment with heterogeneous delay, and non-negligible packet losses.

Scenario 6 examines heterogeneous paths with different loss characteristics. We set the packet loss probability to a fixed value of 3% for one path, while varying the packet loss probability for the other path. Both paths have equal bandwidth (20 Mbps) and delay (150 ms).

Figure 6 depicts the throughput performance when downloading a file of size 10 MB. The results show that with increasing packet loss, the throughput achieved by existing MPTCP algorithms decreases. In contrast, the RSF-LB algorithm outperforms them and provides robust performance even with increasing channel errors. This is because most existing algorithms significantly reduce congestion windows after detecting packet losses. However, the proposed RSF MPTCP can dynamically determine the best way to adjust the congestion windows, using the balanced pricing approach.

This experiment can be summarized as follows:

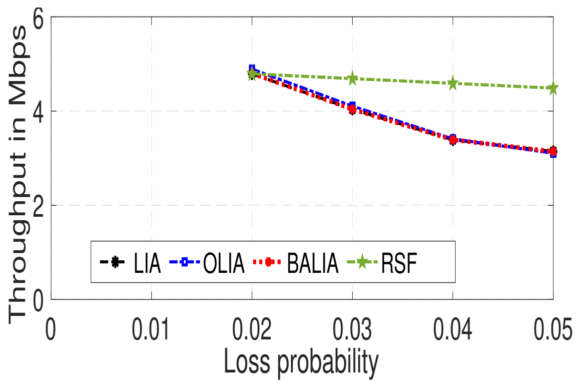

Figure 6: Throughput results for 10 MB file transfer in Scenario 6: Heterogeneous Loss Paths. One path has 3% packet loss, while the loss probability of the other one is varied. Both paths have 20 Mbps bandwidth, and 150 ms delay.

- Observation 6. The proposed MPTCP algorithm is robust and outperforms others under adverse channel conditions.

5 CONCLUSION

We have developed a novel framework for designing MPTCP algorithms by using a *rent-seeking* game-theoretic framework. This approach guarantees the existence of a utility function with strict diagonal concavity.

Our network emulation experiments have demonstrated higher throughput for our RSF MPTCP implementations, without compromising fairness. Furthermore, our RSF MPTCP approach is highly robust to the characteristics of heterogeneous network paths, and provides better responsiveness than other existing MPTCP algorithms.

REFERENCES

[1] E. Altman, M. Datar, G. Burnside, and C. Touati, "Normalized Equilibrium in Tullock Rent Seeking Game", *Proceedings of Game Theory for Networks*, Paris, France, pp. 109-116, April 2019.

[2] Y. Chen, Y. Lim, R. Gibbens, E. Nahum, R. Khalili, and D. Towsley, "A Measurement-based Study of Multipath TCP Performance over Wireless Networks", *Proceedings of ACM Internet Measurement Conference* (IMC), Barcelona, Spain, pp. 455–468, October 2013.

[3] F. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of the Operational Research society*, Vol. 49, No. 1, pp. 237-252, February 1998.

[4] P. Key, L. Massoulié, and D. Towsley, "Path Selection and Multipath Congestion Control", *Communications of the ACM*, Vol. 54, No. 1, pp. 109-116, January 2011.

[5] R. Khalili, N. Gast, M. Popovic, and J-Y. Le Boudec, "MPTCP is Not Pareto-optimal: Performance Issues and a Possible Solution", *IEEE/ACM Transactions on Networking*, Vol. 21, No. 5, pp. 1651-1665, October 2013.

[6] Q. Peng, A. Walid, J. Hwang, and S. Low, "Multipath TCP: Analysis, Design, and Implementation", *IEEE/ACM Transactions on Networking*, Vol. 24, No. 1, pp. 596-609, February 2016.

[7] Q. Peng, A. Walid, and S. Low, "Multipath TCP: Theory and Design", *Proceedings of ACM SIGMETRICS*, Pittsburgh, PA, pp. 305-316, June 2013.

[8] S. Pokhrel and M. Mandjes, "Improving Multipath TCP Performance over WiFi and Cellular Networks: An Analytical Approach", *IEEE Transactions on Mobile Computing*, Vol. 18, No. 11, pp. 2562-2576, November 2019.

[9] S. Pokhrel, M. Panda, and H. Vu, "Fair Coexistence of Regular and Multipath TCP over Wireless Last-Miles", *IEEE Transactions on Mobile Computing*, Vol. 18, No. 3, pp. 574-587, March 2019.

[10] C. Raiciu, M. Handly, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", IETF RFC 6356, October 2011.

[11] A. Reiffers-Masson, Y. Hayel, and E. Altman, "Game Theory Approach for Modeling Competition over Visibility on Social Networks", *Proceedings of 6th IEEE International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, pp. 1-6, January 2014.

[12] J. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games", *Econometrica*, Vol. 33, No. 3, pp. 520-534, July 1965.

[13] G. Tullock, *The Rent-Seeking Society*, Vol. 5, Liberty Fund, 2005.

[14] G. Tullock, G. Brady, and A. Seldon, *Government Failure: A Primer in Public Choice*, Cato Institute, May 2002.

[15] C. Williamson, "Dynamic Bandwidth Allocation Using Loss-Load Curves", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 6, pp. 829-839, December 1996.