MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution

Ramin Khalili, Nicolas Gast, Miroslav Popovic, and Jean-Yves Le Boudec, *Fellow, IEEE*

***Abstract*—Multipath TCP (MPTCP) has been proposed recently as a mechanism for transparently supporting multiple connections to the application layer. It is under discussion at the IETF. We nevertheless demonstrate that the current MPTCP suffers from two problems: P1) Upgrading some TCP users to MPTCP can reduce the throughput of others without any benefit to the upgraded users, which is a symptom of not being Pareto-optimal; and P2) MPTCP users could be excessively aggressive toward TCP users. We attribute these problems to the linked-increases algorithm (LIA) of MPTCP and, more specifically, to an excessive amount of traffic transmitted over congested paths. The design of LIA forces a tradeoff between optimal resource pooling and responsiveness. We revisit the problem and show that it is possible to provide these two properties simultaneously. We implement the resulting algorithm, called the opportunistic linked-increases algorithm (OLIA), in the Linux kernel, and we study its performance over our testbed by simulations and by theoretical analysis. We prove that OLIA is Pareto-optimal and satisfies the design goals of MPTCP. Hence, it can avoid the problems P1 and P2. Our measurements and simulations indicate that MPTCP with OLIA is as responsive and nonflappy as MPTCP with LIA and that it solves problems P1 and P2.**

***Index Terms*—Congestion control algorithm, multipath TCP, performance evaluation, protocol design.**

I. Introduction

THE REGULAR TCP uses a window-based congestion-control mechanism to adjust the transmission rate of users [1]. It always provides a Pareto-optimal allocation of resources: It is impossible to increase the throughput of one user without decreasing the throughput of another or without increasing the congestion cost [2]. It also guarantees a fair allocation of bandwidth among the users, but favors the connections with lower round-trip time (RTT) [3].

Various mechanisms were used to build a multipath transport protocol compatible with the regular TCP. Authors of [4]–[6] propose a family of algorithms inspired by utility maximization frameworks. These algorithms tend to use only the best paths available to users and are optimal in static settings where paths have similar RTTs. In practice, however, they suffer from several problems [7]–[9]. First, they sometimes fail to quickly detect free capacity because they do not probe paths with high loss probabilities sufficiently. Second, they exhibit flappiness: When there are multiple good paths available to a user, the user will randomly flip its traffic between these paths. This is not desirable, specifically, when the achieved rate depends on RTTs, as with TCP.

Multipath TCP (MPTCP) is a concrete proposal for multipath transport; it is under discussion at the IETF [10]. Because of the issues aforementioned, its congestion control part does not follow the algorithms in [4]–[6]. Instead, it follows an *ad hoc* design based on three goals [10]: 1) Improve throughput: A multipath TCP user should perform at least as well as a TCP user that uses the best path available to it. 2) Do no harm: A multipath TCP user should never take up more capacity from any of its paths than a TCP user. 3) Balance congestion: A multipath TCP algorithm should balance congestion in the network, subject to meeting the first two goals.

MPTCP compensates for different RTTs and solves many problems of multipath transport [7], [9]: It can effectively use the available bandwidth; compared to independent TCP flows, it improves throughput and fairness in many scenarios; and it solves the flappiness problem. Through analysis and by using measurements over a testbed, we nevertheless demonstrate that MPTCP still suffers from the following problems.

P1) Upgrading some regular TCP users to MPTCP can reduce the throughput of other users without any benefit to the upgraded users. Hence, MPTCP is not Pareto-optimal.

P2) MPTCP users could be excessively aggressive toward TCP users.

We attribute these problems to the "linked increases" algorithm (LIA) of MPTCP [10] and specifically to an excessive amount of traffic transmitted over congested paths. These problems indicate that MPTCP fails to fully satisfy its design goals, especially goal 3).

The design of LIA forces a tradeoff between optimal resource pooling and responsiveness; it cannot provide both at the same time. Hence, to provide good responsiveness, LIA's current implementation must depart from Pareto optimality, which leads to problems P1 and P2. We revisit the design and show that it is possible to simultaneously provide both properties. We introduce OLIA, the "opportunistic linked-increases algorithm," as an alternative to LIA. Based on utility maximization frameworks, we prove that OLIA is Pareto-optimal. Hence, it can avoid the problems P1 and P2. Furthermore, its construction makes it as responsive and nonflappy as LIA.

OLIA is a window-based congestion-control mechanism. Similarly to LIA, it couples the additive increases and uses

Manuscript received February 01, 2013; accepted June 21, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Allman. Date of publication August 21, 2013; date of current version October 11, 2013. This work was supported in part by the EU 7th Framework Programme (FP7/2007-2013) under Grant Agreement No. 257740 (Network of Excellence "TREND") and the EU project CHANGE (FP7-ICT-257422).

R. Khalili is with T-Labs/TU-Berlin, 10587 Berlin, Germany (e-mail: ramin@net.t-labs.tu-berlin.de).

N. Gast, M. Popovic, and J.-Y. Le Boudec are with the IC-LCA2, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: nicolas.gast@epfl.ch; miroslav.popovic@epfl.ch; jean-yves.leboudec@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2274462

unmodified TCP behavior in the case of a loss. OLIA's increase part, (5), has two terms.

- The first term is an adaptation of the increase term of Kelly and Voice's algorithm [4]. This term is essential to provide Pareto optimality.
- The second term guarantees responsiveness and nonflappiness of OLIA. By measuring the number of transmitted bits since the last loss, it reacts to events within the current window and adapts to changes faster than the first term.

By adapting the window increases as a function of RTTs, OLIA also compensates for different RTTs.

We implement OLIA in the Linux kernel and study its performance over our testbed by simulations and by theoretical analysis. Using a fluid model of OLIA based on differential inclusion, we prove that OLIA is Pareto-optimal (Theorem 3) and that it satisfies the design goals of MPTCP (Corollary 2). Our measurements and simulations indicate that MPTCP with OLIA is as responsive and nonflappy as MPTCP with LIA and it solves problems P1 and P2. Note that OLIA is now part of the Louvain MPTCP implementation [11].

A recent study by Chen *et al.* [12] shows that MPTCP with OLIA always outperforms MPTCP with LIA in wireless networks and is very responsive to the changes in the environment. These results confirm our findings in this paper. Hence, we believe that MPTCP working group in IETF [13] should revisit the congestion control part of MPTCP, and that an alternative algorithm, such as OLIA, should be considered.

In Section II, we briefly introduce MPTCP and LIA and discuss related work. In Section III, we provide a number of examples and scenarios where MPTCP with LIA exhibits problems P1 and P2. In Section IV, we introduce OLIA and detail its Linux implementation. In Section V, we prove that OLIA is Pareto-optimal and satisfies MPTCP's design goals. In Section VI, we study the performance of OLIA through measurements and by simulations.

II. MPTCP and Related Work

MPTCP is a set of extensions to the regular TCP, which allows users to spread their traffic across potentially disjoint paths [10]. MPTCP discovers the number of paths available to a user, establishes the paths, and distributes traffic across these paths through creation of separate subflows [14], [15]. The congestion control algorithm of MPTCP is inspired by the utility frameworks of [4] and [5], which provide optimal resources pooling. However, it departs from the optimal resource pooling principle [16] to avoid flappiness and to improve response time [8], [9], [17].

Congestion control algorithm of MPTCP forces a tradeoff between optimal resource pooling and responsiveness [8]. The idea behind the algorithm is to transmit over a path r at a rate proportional to $p_r^{-1/\varepsilon}$, where p_r is the loss probability over this link and $\varepsilon \in [0,2]$ is a design parameter. The choice $\varepsilon = 0$ corresponds to the fully coupled algorithm of [4]–[6]: The traffic is sent only over the best paths, and it is Pareto-optimal but is flappy. The choice $\varepsilon = 2$ corresponds to using uncoupled TCP flows on each path: It is very responsive and nonflappy, but does not balance congestion. MPTCP's implementation uses $\varepsilon = 1$ to provide a compromise between optimal resource pooling and responsiveness. This algorithm is called LIA [10].

Let w_r and rtt_r be the window size and the estimated round-trip time on path $r \in \mathcal{R}_u$. $\mathcal{R}_u$ is the set of all paths available to user u. LIA works as follows.

- For each ACK on subflow r, increase w_r by

$$\min\left(\frac{\max_{i\in\mathcal{R}_u} w_i/\mathrm{rtt}_i^2}{\left(\sum_{i\in\mathcal{R}_u} w_i/\mathrm{rtt}_i\right)^2}, \frac{1}{w_r}\right). \tag{1}$$

- For each loss on subflow r, decrease w_r by $w_r/2$.

LIA increases by at most $1/w_r$ to be at most as aggressive as regular TCP on any of its paths. When the RTTs are similar, this minimum can be neglected as the first term $(\max_i w_i/\mathrm{rtt}_i^2)/(\sum_i w_i/\mathrm{rtt}_i)^2$ will always be less than $1/w_r$. In this case, a fixed-point analysis provides a simple loss-throughput formula for LIA [9]: LIA allocates to a path r a window w_r proportional to the inverse of the loss probability $1/p_r$ and such that the total rate $\sum_{p\in\mathcal{R}_u} w_p/\mathrm{rtt}_p$ equals the rate that a regular TCP user would get on the best path, i.e., $\max_{p\in\mathcal{R}_u}\sqrt{2/p_p}/\mathrm{rtt}_p$. Thus, the window size for the flow on a path r is given by

$$w_r = \frac{1}{p_r}\cdot\frac{\max_{p\in\mathcal{R}_u}\sqrt{2/p_p}/\mathrm{rtt}_p}{\sum_{p\in\mathcal{R}_u} 1/(\mathrm{rtt}_p p_p)}. \tag{2}$$

Hence, two paths with similar qualities get equal windows, removing flappiness. When the path qualities differ, a larger window is allocated to the path with higher rate, providing some load balancing.

Besides MPTCP and algorithms in [4]–[6], a few other algorithms have been proposed to implement multipath protocols. In [18], an opportunistic multipath scheduler measures the path conditions on time scales up to several seconds. Reference [19] uses a mechanism to detect shared bottlenecks and to avoid the use of multiple subflows on the same bottleneck. Reference [20] proposes to use uncoupled TCP flows with a weight depending on the congestion level. These mechanisms are complex, their robustness is not clear, and they need explicit information about congestion in the network. Our proposed algorithm, OLIA, differs from these works as it is implemented, proven to be Pareto-optimal, and relies only on information that is available to regular TCP. It also differs from [4]–[6] as it is not flappy and has a better responsiveness.

III. Performance Problems of MPTCP

In this section, we investigate the behavior of MPTCP with LIA in three different scenarios: A, B, and C. Using scenarios A and B, we show that upgrading some regular TCP users to MPTCP could reduce the throughput of other users in the network without any benefit to the upgraded users (problem P1). In Scenario C, we discuss the aggressiveness of MPTCP users that compete with regular TCP users (problem P2). Our conclusions are based on analytical results and measurements over a testbed.

A. Testbed Setup

To investigate the behavior of the algorithms, we create three testbed topologies that represent our scenarios. Server–client PCs run MPTCP (with LIA or OLIA) enabled Linux kernels. In all scenarios, laptop PCs are used as routers. We install "Click

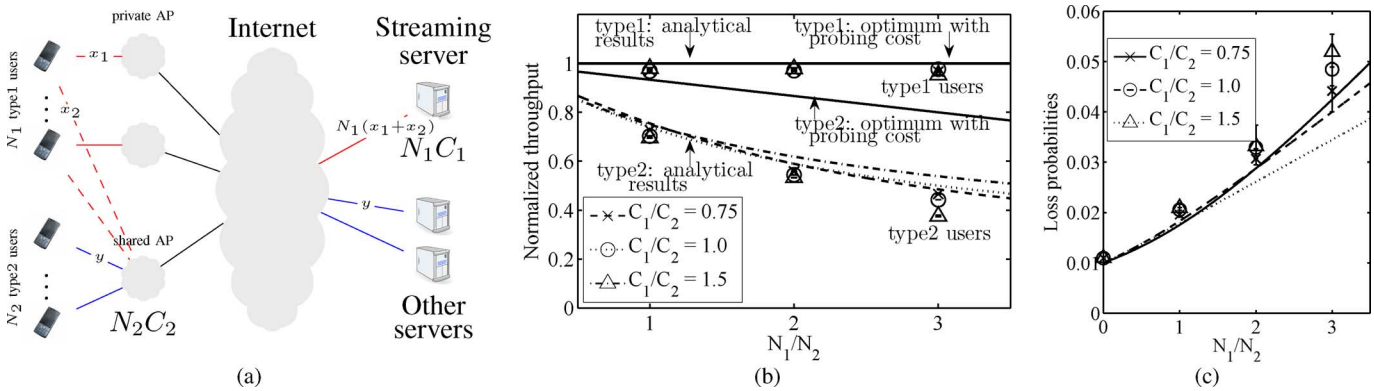


Fig. 1. Scenario A: Type1 users are all downloading through the same streaming server and have access to both a private high-speed access point and a shared access point. Type2 users have access only to the shared access point. The performance of MPTCP with LIA obtained by measurement (points) or numerical analysis (lines) is shown in (b) and (c). We observe that it is not Pareto-Optimal, penalizes type2 users, and its performance is far from the theoretical optimum with probing cost. It also fails to balance the congestion. (a) Scenario A. (b) Normalized throughput of users $(x_1+x_2)/C_1$ and y/C_2. (c) Loss probability p_2 at the shared AP.

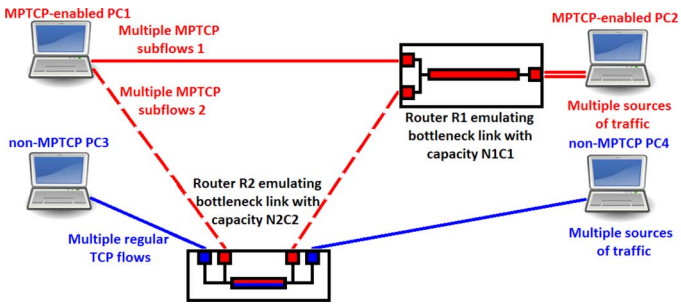


Fig. 2. Testbed implementation of scenario A: Router R_1 emulates the bottleneck at the server side, and router R_2 the shared AP bottleneck. Iperf is used to emulate multiple connections. The top PCs use MPTCP, and the bottom PCs use regular TCP.

Modular Router" software [21] to emulate topologies with different characteristics. This is possible as Click allows custom manipulation of packets from the moment they arrive at one of the interfaces until the moment they leave the router. The testbed configuration of the scenario described in Fig. 1(a) is represented in Fig. 2.

We emulate links with configurable bandwidth and delay with RED queuing (drop-tail queuing is also studied in the simulations that use htsim; see Section VI-B). We set the propagation delay, the round-trip time between a sender and a receiver over an uncongested path, to 80 ms. For a 10-Mb/s link, we set the dropping probability equal to 0 up to a queue size of $\min_{\text{th}} = 25$. Then, it grows linearly to the value 0.1 at $\max_{\text{th}} = 50$. It again increases linearly up to 1 at $2\max_{\text{th}}$. The queue size is set to 300 packets. The parameters are proportionally adapted when the link capacity changes. This results to an average queuing delay of 70 ms in the queues, as observed by measurements. We use Iperf to generate the traffic that emulates bulk transfers of large sizes. Each Iperf session runs for 120 s to allow the flows to reach equilibrium. The flows are initiated in the random order.

B. Scenario A: MPTCP Is Not Pareto-Optimal and Penalizes Regular TCP Users

Consider a network with two types of users as shown in Fig. 1(a). There are N_1 users of type1, each with a high-speed private connection, accessing different files on a media streaming server. The server has a network connection with capacity limit of N_1C_1 Mb/s. These users can activate a second connection through a shared access point (AP) by using MPTCP. There are also N_2 type2 users that have connections only through the shared AP, downloading their contents from the Internet. The shared AP has a capacity of N_2C_2 Mb/s.

Let x_1 be the rate that a type1 user receives over its private connection. By symmetry, every user of type1 will receive the same rate x_1. Similarly, let x_2 (resp. y) be the rate that a type1 (resp. type2) user receives over the shared connection. We denote by p_1 and p_2 the loss probability at the link connected to the streaming server and the shared AP, respectively. The loss probabilities at the Internet backbone and the private APs are assumed negligible.

When type1 users use only their own private AP, we have $x_1 = C_1$, $x_2 = 0$, and $y = C_2$. In this case, the normalized throughput for both type1 and type2 users is 1. In the other case, assuming that all paths have RTT rtt, when all type1 users activate their public connections and use MPTCP with LIA to balance load between their connections, we have

$$\text{(a)}\quad N_1(x_1+x_2) = N_1C_1 \quad N_1x_2 + N_2y = N_2C_2$$

$$\text{(b)}\quad x_1 + x_2 = \frac{1}{\text{rtt}}\sqrt{\frac{2}{p_1}} \quad x_2 = \frac{1}{2+p_2/p_1}\frac{1}{\text{rtt}}\sqrt{\frac{2}{p_1}}$$

$$\text{(c)}\quad y = \frac{1}{\text{rtt}}\sqrt{2/p_2}$$

where (a) are the capacity constraints at the two bottlenecks, (b) comes from the loss-throughput formula for LIA [(2)], and (c) follows the TCP loss-throughput formula [22]. This system has a unique solution (see Appendix-A). Fig. 1(b) depicts the normalized throughput of type1 and type2 users, i.e., $(x_1+x_2)/C_1$ and y/C_2. As shown in Appendix-A, these values depend only on the ratios C_1/C_2 and N_1/N_2.

A theoretically optimal algorithm (as discussed in [4] and [5]) will allocate a normalized throughput of 1 to both type1 and type2 users. In practice, however, the value of the congestion windows are bounded below by 1 maximum segment size (MSS). Hence, with a window-based congestion-control algorithm, a minimum probing traffic of 1 MSS per RTT will be sent over an established path. In this paper, we introduce a

theoretical baseline for window-based congestion-control algorithms, called *theoretical optimum with probing cost*; it provides optimal resource pooling in the network, given that a minimum probing traffic of 1 MSS per RTT is sent over each path. It serves as a reference to see how far from the optimum LIA is.

We measure the performance of LIA in Scenario A by using the testbed, as shown in Fig. 2. PC1 and PC2 run MPTCP-enabled Linux kernel implementation and have two Ethernet interfaces. PC3 and PC4 use regular TCP. Within router PCs R1 and R2, we emulate links with capacities N_1C_1 and N_2C_2 modeling respectively the bottleneck at the server side and the shared AP. With Iperf, we generate independent MPTCP connections between PC1 and PC2 and regular TCP connections between PC3 and PC4.

The measurements are taken for $N_2 = 10$ and three values of $N_1 = 10, 20, 30$. The capacities of R1 and R2 are N_1C_1 and N_2C_2 Mb/s, where we set $C_2 = 1$ Mb/s and $C_1 = 0.75, 1, 1.5$ Mb/s. All paths have similar RTTs (link delay plus queuing delay is around 150 ms over all paths). For each case, we took five measurements. The results are reported in Fig. 1(b). Note that in all cases we present 95% confidence intervals, but in many cases they are too small to be visible. We also show our analytical analysis of LIA, as well as the theoretical optimum with probing cost as defined above. Note that the network setting is very static, and the randomness of our results mainly comes from the congestion losses at the queues and the fact that the flows are initiated in the random order. Moreover, the queuing delay in a queue depends on its queue size and is therefore random.

These figures have multiple implications. First, they show that MPTCP with LIA exhibits problem P1 from the Introduction: Upgrading type1 users to MPTCP penalizes type2 users without any gain for type1 users. As the number of type1 users increases, the throughput of type2 users decreases, but the throughput of type1 users does not change as it is limited by the capacity C_1 of the streaming server. For $N_1 = N_2$, type2 users see a decrease of about 30% in their throughput. When $N_1 = 3N_2$, this decrease is between 50%–60%. This is explained by the fact that LIA does not fully balance congestion, as shown in Fig. 1(c). It excessively increases congestion on the shared AP (not in compliance with goal 3). Note that p_1 depends only on C_1. Our measurements show that in average $p_1 = 0.02, 0.009, 0.004$ for $C_1 = 0.75, 1, 1.5$ Mb/s, respectively. Hence, we observe that LIA performs far from how an optimal algorithm with probing cost would perform. Furthermore, these figures show that the fixed-point analysis predicts accurately the behavior of the algorithm: The theoretical and experimental curves exhibit the same trend.

C. Scenario B: MPTCP Is Not Pareto-Optimal and Can Penalize Other MPTCP Users

Consider the multihoming scenario depicted in Fig. 3. We have four Internet service providers (ISPs), X, Y, Z, and T. Y is a local ISP in a small city, which connects to the Internet through Z. X, Z, and T are nationwide service providers and are connected to each other through high-speed links. X provides Internet services to users in the city and is a competitor of Y. They have access capacity limits of C_X, C_Y, C_Z, and C_T.

Z and T host different video streaming servers. There are two types of users: N_B Blue users download contents from a server in Z, and N_R Red users download from a server in ISP T. Blue users use multihoming and are connected to both ISPs X and Y to increase their reliability. Red users can connect either only to Y or to both X and Y. We assume that only ISPs X and T are bottlenecks and denote by p_X and p_T the loss probabilities. All paths have similar RTTs.

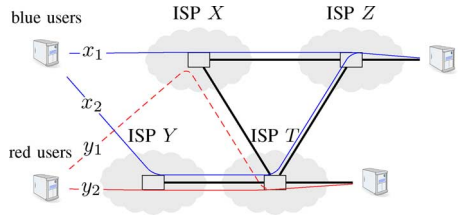


Fig. 3. Scenario B. Thick lines represent peering agreements. Blue users are downloading from servers in ISP Z, and Red users from servers in ISP T. Blue users use multihoming and have access to ISPs X and Y. Initially, Red users have access only to ISP Y but upgrade to MPTCP and connect to both X and Y (by activating the dashed connection).

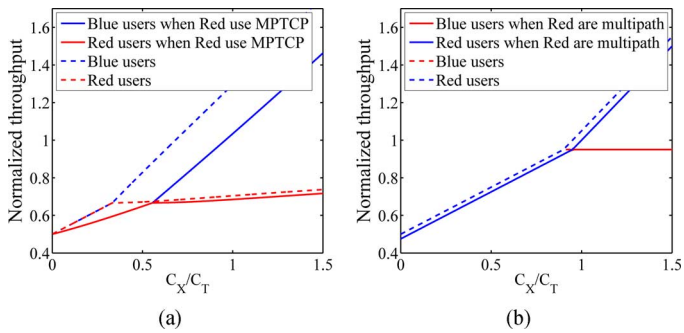


Fig. 4. Analytical results for Scenario B with 15 Blue and 15 Red users. We show the normalized throughput $(15(x_1 + x_2)/C_T$ and $15(y_1 + y_2)/C_T)$ as a function of C_X/C_T. Dashed curves: normalized throughput when Red users connect only to ISP Y. Solid curves: the case when Red users upgrade to multipath. For all values of C_X/C_T, the throughput of all users decreases when Red users upgrade to MPTCP. (a) Performance of LIA. (b) Optimum with probing cost.

We first present a theoretical analysis of the rate that each user would achieve using MPTCP. To simplify the analysis, we assume similar numbers of Blue and Red users. There are two possible cases. When Red users connect only to Y, the analysis is the same as the one of scenario C, given in Section III-D. Here, we analyze the case when Red users upgrade to MPTCP. The loss throughput formula [(2)] shows that the throughput of the different connections are

$$\begin{cases} y_1 = \frac{1/\mathrm{rtt}}{2+\frac{p_X}{p_T}}\sqrt{\frac{2}{p_T}} \\ y_2 = \frac{p_X+p_T}{p_T}y_1 \end{cases}, \quad \begin{cases} x_1 = \frac{1/\mathrm{rtt}}{1+p_X/p_T}\sqrt{\max\frac{2}{p_X},\frac{2}{p_T}} \\ x_2 = \frac{1/\mathrm{rtt}}{1+p_T/p_X}\sqrt{\max\frac{2}{p_X},\frac{2}{p_T}} \end{cases}.$$

As shown in Appendix-B, this set of equations has a unique positive solution. A numerical evaluation of these formulas is depicted in Fig. 4(a). Fig. 4(b) depicts the performance of a theoretical optimum with probing cost (see Appendix-B). The results are presented for $\mathrm{RTT} = 150$ ms, $C_Y = C_Z = 100$ Mb/s, and $C_T = 36$ Mb/s. We consider 15 Blue users and 15 Red users in the network. We depict the normalized throughput $(15(x_1 + x_2)/C_T$ and $15(y_1 + y_2)/C_T)$ as a function of C_X/C_T. The results show that upgrading Red users to MPTCP with LIA decreases the performance for everyone. As an example, when $C_X/C_T \approx 0.75$, by upgrading the Red users, we reduce the

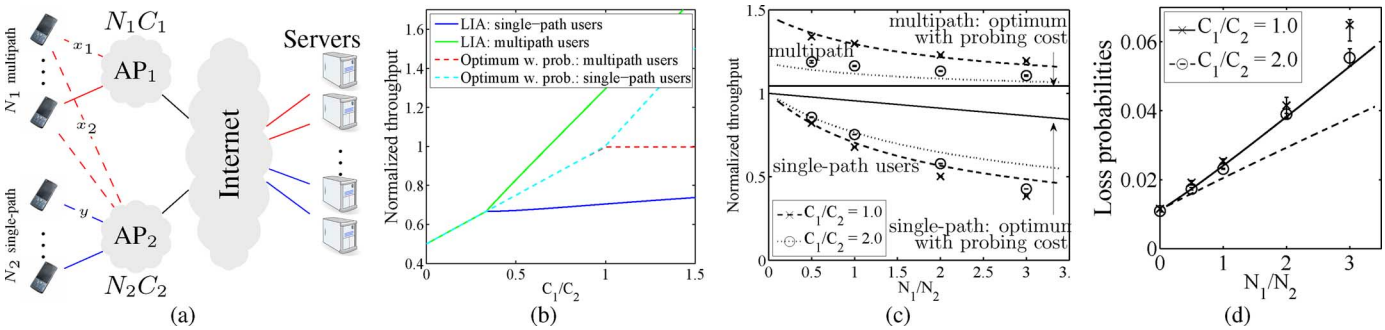


Fig. 5. Scenario C: MPTCP with LIA excessively penalizes TCP users (when $C_1/C_2 \geq 1$, for any fairness criterion, MPTCP users should not impact TCP users). We show the normalized throughputs ($(x_1+x_2)/C_1$ and y/C_2) received by the users, as well as p_2. The performance of LIA is far from the theoretical optimum with probing cost. (a) Scenario C: N_1 multipath users and N_2 single-path users are connected to two APs with capacities N_1C_1 and N_2C_2 Mb/s. (b) Analytical results: normalized throughput of all users using LIA (solid) or optimum with probing cost (dashed) for $N_1 = N_2$. (c) Normalized throughputs using LIA, obtained by measurement (points) or analysis (lines). (d) Loss probability p_2 at AP2: LIA fails to balance the congestion.

TABLE I
MEASUREMENT RESULTS FOR SCENARIO B

Red users	Rate/user		Aggregate
	Blue users	Red users	
Single-path	2.5	1.5	59.8
Multipath	2.0	1.4	52.0

The number of Red and Blue users is 15 and all values are recorded in Mbps. By upgrading Red users to MPTCP, the throughput drops for all users and the aggregate throughput falls by 13%.

throughput of the Blue users by up to 21%. This decrease is about 3% when we use an optimal algorithm with probing cost [Fig. 4(b)].

We emulate this scenario in our testbed in a similar manner as for Scenario A. The measurement results are reported in Table I for a similar setting with $C_X = 27$ Mb/s. We observe that when Red users only connect to ISP Y, the aggregate throughput of users is close to the cut-set bound, 63 Mb/s. However, Blue users get a higher share of the network bandwidth. Now consider that Red users upgrade to MPTCP by establishing a second connection through X (shown by dashed line in Fig. 3). Our results in Table I show that Red users do not receive any higher throughput. However, the average rate of Blue users drops by 20%, which results in a drop of 13% in aggregate throughput.

D. Scenario C: MPTCP Users Could Be Excessively Aggressive Toward TCP Users

We consider a scenario with N_1 multipath users, N_2 single-path users, and two APs with capacities N_1C_1 and N_2C_2 Mb/s (see Fig. 5). Multipath users connect to both APs, and they share AP2 with single-path users.

If the allocation of rates is proportionally fair, multipath users will use AP2 only if $C_1 < C_2$ and all users will receive $(N_1C_1+N_2C_2)/(N_1+N_2)$. When $C_1 > C_2$, a fair multipath user will not transmit over AP2. This fair allocation is represented by dashed lines in Fig. 5(b) when we take into account the minimum probing cost (the analysis is similar to what we proposed in Appendix-B, Case 1). However, using MPTCP with LIA, multipath users get a larger share of bandwidth as soon as $C_1 \geq C_2/(2+N_1/N_2)$.

Let p_1 and p_2 be the loss probabilities at APs, x_1 and x_2 be rates that a multipath user receives over its paths, and y be the rate of a single-path user. Assume all RTTs are the same. When $C_1/C_2 < 1/(2+N_1/N_2)$, we have $p_1 > p_2$, and all users receive the same rate: $x_1+x_2 = y = (C_1+C_2)/2$. When $C_1/C_2 > 1/(2+N_1/N_2)$, we have $p_1 < p_2$, and the fixed-point formula of LIA gives

$$x_1 = \frac{p_2}{p_1+p_2}\frac{1}{\text{rtt}}\sqrt{\frac{2}{p_1}} \quad \text{and} \quad x_2 = \frac{p_1}{p_1+p_2}\frac{1}{\text{rtt}}\sqrt{\frac{2}{p_1}}.$$

Moreover, both the APs are bottlenecks, and we have $x_1 = C_1$ and $x_2 + y = C_2$. Let $z := \sqrt{p_1/p_2}$. Using that the TCP loss throughput formula $y = \sqrt{2/p_2}$, the quantity z is the unique positive root of

$$z^3 + \frac{N_1}{N_2}z^2 + z - \frac{C_2}{C_1}.$$

The normalized throughputs of multipath users are $(x_1 + x_2)/C_1 = (1 + p_1/p_2)/C_1 = 1 + z^2$. The single-path users receive a rate of $y/C_2 = 1 - \frac{N_1C_1}{N_2C_2}z^2$. Again, this quantity only depends on the ratio N_1/N_2 and C_1/C_2.

Fig. 5(b) reports a numerical evaluation of these fixed-point equations for the case $N_1 = N_2$. We show the normalized throughputs ($(x_1+x_2)/C_1$ and y/C_2) received by the users, as well as p_2. We observe that LIA is fair with regular TCP users, as long as $C_1 < C_2/3$. However, as C_1 exceeds $C_2/3$, it takes most of the capacity of AP2 for itself.

We emulate the scenario in our testbed and measure the performance of MPTCP with LIA. The results are reported in Fig. 5(c) and (d) for $C_2 = 1$ Mb/s and $C_1 = 1$, 2 Mb/s, with $N_2 = 10$ and $N_1 = 5$, 10, 20, 30. As in scenario A, we also present the theoretical optimum with probing cost in Fig. 5(c). When $C_1/C_2 \geq 1$, multipath users should not use AP2 at all. However, our results show that MPTCP users are disproportionately aggressive and exhibit problem P2. Fig. 5(d) shows the loss probability at AP2. We observe that LIA excessively increases congestion on AP2 and is unable to fully balance congestion in the network. Also, we have $p_1 = 0.01$ and 0.003 for $C_1 = 1$ and 2 Mb/s, respectively.

IV. OLIA: THE OPPORTUNISTIC LINKED INCREASES ALGORITHM

In this section, we introduce OLIA as an alternative for MPTCP's LIA. OLIA is a window-based congestion-control algorithm that couples the increase of congestion windows and uses unmodified TCP behavior in the case of a loss. The increase part of OLIA has two terms. The first term is an

adaptation of Kelly and Voice's increase term and provides the Pareto optimality. Kelly and Voice's algorithm is based on scalable TCP; the first term is a TCP compatible version of their algorithm that compensates also for different RTTs. The second term, with α, guarantees responsiveness and nonflappiness. We first present the algorithm and its Linux implementation. Then, we illustrate with an example its operation and its difference with LIA.

A. Detailed Description of OLIA

Let $\mathcal{R}_u$ be the set of paths available to user u, and let $r \in \mathcal{R}_u$ be a path. We denote by $\ell_{1r}(t)$ the number of bits that were successfully transmitted by u over path r between the last two losses seen on r, and by $\ell_{2r}(t)$ the number of bits that are successfully transmitted over r after the last loss. If no losses have been observed on r up to time t, then $\ell_{1r}(t) = 0$ and $\ell_{2r}(t)$ is the total number of bits transmitted on r. Also, let $\ell_r(t) = \max\{\ell_{1r}(t), \ell_{2r}(t)\}$, and let $\mathrm{rtt}_r(t)$ and $w_r(t)$ be respectively RTT and the window on r at time t. We define

$$\mathcal{M}(t) = \left\{ i(t) \,|\, i(t) = \arg\max_{p \in \mathcal{R}_u} w_p(t) \right\} \tag{3}$$

$$\mathcal{B}(t) = \left\{ j(t) \,|\, j(t) = \arg\max_{p \in \mathcal{R}_u} \frac{\ell_p(t)}{\mathrm{rtt}_p(t)^2} \right\}. \tag{4}$$

$\mathcal{M}(t)$ is the set of the paths of u with the largest window sizes at time t. $\mathcal{B}(t)$ is the set of the paths at time t that are presumably the best paths for u: $1/\ell_r(t)$ can be considered as an estimate of packet loss probability on path r at time t, and hence the rate that path r can provide to a TCP user can be estimated by $\sqrt{2\ell_r(t)}/\mathrm{rtt}_r$ [22].

Our algorithm is as follows (to simplify notation, we drop the time argument t; however, note that w_r, rtt_r, ℓ_r, $\mathcal{M}$, and $\mathcal{B}$ are all functions of time).

- For each ACK on path r, **increase** w_r by

$$\frac{w_r/\mathrm{rtt}_r^2}{\left(\sum_{p \in \mathcal{R}_u} w_p/\mathrm{rtt}_p\right)^2} + \frac{\alpha_r}{w_r} \tag{5}$$

  where α_r is calculated as follows:

$$\alpha_r = \begin{cases} \frac{1/|\mathcal{R}_u|}{|\mathcal{B}\setminus\mathcal{M}|}, & \text{if } r \in \mathcal{B} \setminus \mathcal{M} \neq \emptyset \\ -\frac{1/|\mathcal{R}_u|}{|\mathcal{M}|}, & \text{if } r \in \mathcal{M} \text{ and } \mathcal{B} \setminus \mathcal{M} \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

  $\mathcal{B}\setminus\mathcal{M}$ is the set of elements in $\mathcal{B}$ but not in $\mathcal{M}$, $\emptyset$ is the empty set, and $|\mathcal{R}_u|$ is the number of paths available to u at the time. Note that $\sum_{r \in \mathcal{R}_u} \alpha_r = 0$.
- For each loss on path r, **decrease** w_r by $\frac{w_r}{2}$.

By definition of α_r, if all the best paths have the largest window size, i.e., if $\mathcal{B} \setminus \mathcal{M} = \emptyset$, then $\alpha_r = 0$ for any $r \in \mathcal{R}_u$. This is because we already use the capacity available to the user by using all the best paths.

If there is any best path with a small window size, i.e., if $\mathcal{B} \setminus \mathcal{M} \neq \emptyset$, then α_r is positive for all r in $\mathcal{B}\setminus\mathcal{M}$ and negative for all r in $\mathcal{M}$. Hence, our algorithm increases windows faster on the paths that are presumably best but that have small windows. The increase will be slower on the paths with maximum windows. In this case, OLIA reforwards traffic from fully used paths (i.e., paths in $\mathcal{M}$) to paths that have free capacity available to the users (i.e., paths in $\mathcal{B}\setminus\mathcal{M}$).

B. Linux Implementation of OLIA

We implemented OLIA in the MPTCP release supported on the Linux kernel 3.0.0 [11]. Similarly to LIA, our algorithm only applies to the increase part of the congestion avoidance phase. The fast retransmit and fast recovery algorithms, as well as the multiplicative decrease of the congestion avoidance phase, are the same as in TCP [1]. We also use a similar slow-start algorithm as in TCP, with the modification that we set the slow-start threshold (ssthresh) to be 1 MSS if multiple paths are established. In the case of a single-path flow, we use similar minimum ssthresh as in TCP (2 MSS). The purpose of this modification is to avoid transmitting unnecessary traffic over congested paths when multiple paths are available to a user. The minimum congestion windows size is 1 MSS as in TCP. Our implementation is now part of the Louvain MPTCP implementation [11].

One important part of our implementation is the measurement of ℓ_r on a path r. This can be done easily by using information that is already available to a regular TCP user. Our algorithm for computing ℓ_r is as follows.

- For each ACK on r: $\ell_{2,r} \leftarrow \ell_{2,r}+$ (number of bits that are acknowledged by ACK);
- For each loss on r: $\ell_{1,r} \leftarrow \ell_{2,r}$ and $\ell_{2r} \leftarrow 0$;

where $\ell_r = \max\{\ell_{1,r}, \ell_{2,r}\}$. $\ell_{1,r}$ and $\ell_{2,r}$ are initially set to zero when the connection is established. To compute a smoothed estimate of rtt_r, we use the algorithm proposed in [23] and implemented in the Linux kernel.

C. Illustrative Example of OLIA's Behavior

To give more insight into how OLIA performs, we show the evolution of window sizes and α values for a two-path flow (see Figs. 6–8). The testbed configuration is shown in Fig. 7. The measurement results on our testbed are reported in Figs. 6 and 8.

We first consider a symmetric case, depicted in Fig. 7(a). As both of the paths are equally good, a multipath user will benefit from using both of them. Fig. 6(a) shows the evolution of w_r and α_r as a function of time. We observe that OLIA simultaneously uses both of the paths, similarly to LIA [Fig. 6(b)], which is the desired behavior. There is no sign of flappiness as α_1 and α_2 react quickly to changes and adjust w_1 and w_2 accordingly.

We now study the asymmetric scenario of Fig. 7(b). In this case, the second path is shared with 10 TCP flows, and multipath users should use only the first path. This is what we observe in Fig. 8(a). The window on the congested path is 1, most of the time (because of the first increase term). However, due to α, the window increases from time to time over the congested path whenever the path has the largest interloss distance ℓ_r. This increase is brief as losses occur more frequently on this path. LIA, however, transmits significant traffic over the congested paths and lower traffic, compared to OLIA, over the good path as depicted in Fig. 8(b).

V. Pareto Optimality of OLIA

In this section, we build a fluid model of OLIA by using differential inclusions. We show that this model provides a Pareto-optimal allocation (Theorem 3) that satisfies the three design goals of MPTCP [10] (Corollary 2). Also, we prove that MPTCP with OLIA is fair with TCP: If all routes of a user have the same

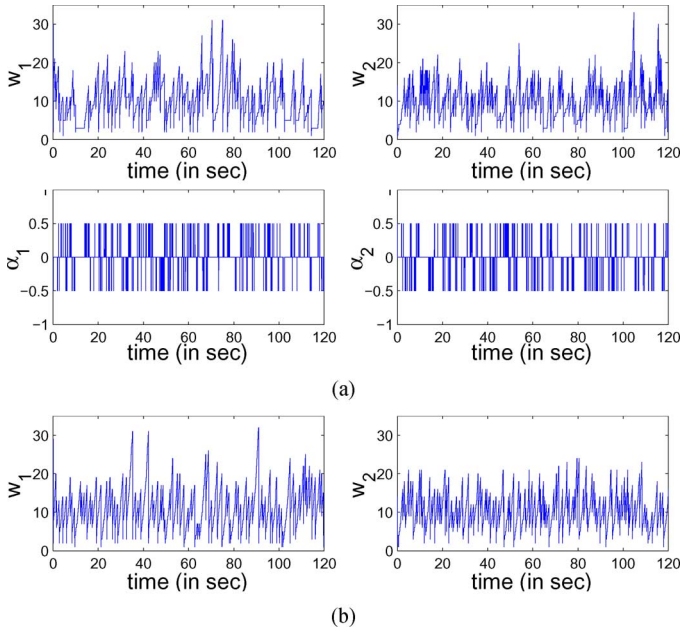


Fig. 6. Evolution of w and α values for a two-path flow. Each path is shared with five regular TCP users. OLIA uses both of the paths, similarly to LIA, and there is no sign of flappiness. (a) MPTCP–OLIA: window size and α_r as a function of time. (b) MPTCP–LIA: window size.

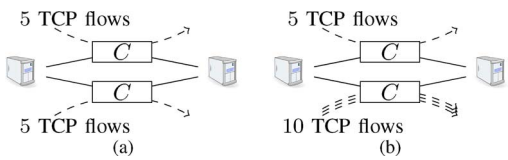


Fig. 7. Multipath user sharing two bottlenecks of the same capacity C with single-path users. (a) Symmetric scenario. (b) Asymmetric scenario.

RTT, then OLIA maximizes the same fairness criteria as the regular TCP (Theorem 4).

A. Fluid Model of OLIA

We consider a network model similar to [3]. The network is static and composed of a set $\mathcal{L}$ of links (or resources). We denote by $\mathcal{R}_u$ the set of paths available to a user u, each path being a set of links. If the route r is available to user u, we write $r \in R_u$. If a route r uses a resource ℓ, we write $\ell \in r$. Similarly, we refer to all routes that cross ℓ as $r \ni \ell$.

Let $x_r(t) \geq 0$ be the rate of traffic transmitted by the user u on a path $r \in \mathcal{R}_u$. We assume that the RTT of a route r is fixed in time, and we denote it by rtt_r. In the fluid model, the rate x_r is an approximation of the window size divided by the RTT, i.e., $x_r = w_r/\mathrm{rtt}_r$.

Let $p_\ell(\sum_{\ell\in r} x_r)$ be the loss rate at link ℓ. p_ℓ depends on the capacity of the link, C_ℓ, and the total amount of traffic sent through the link, $\sum_{\ell\in r} x_r$. We assume that p_ℓ is an increasing function of $\sum_{\ell\in r} x_r$. To simplify the notation, we omit the dependence on x and write only p_ℓ. However, note that if x varies with time, p_ℓ will also vary. We assume that the loss probabilities of links are independent and small; hence, the loss probability on a route r is $p_r = 1-\prod_{\ell\in r}(1-p_\ell) \approx \sum_{\ell\in r} p_\ell$.

When p_r is small, a user u receives acknowledgments on a route $r \in \mathcal{R}_u$ at rate x_r and increases the window w_r as (5).

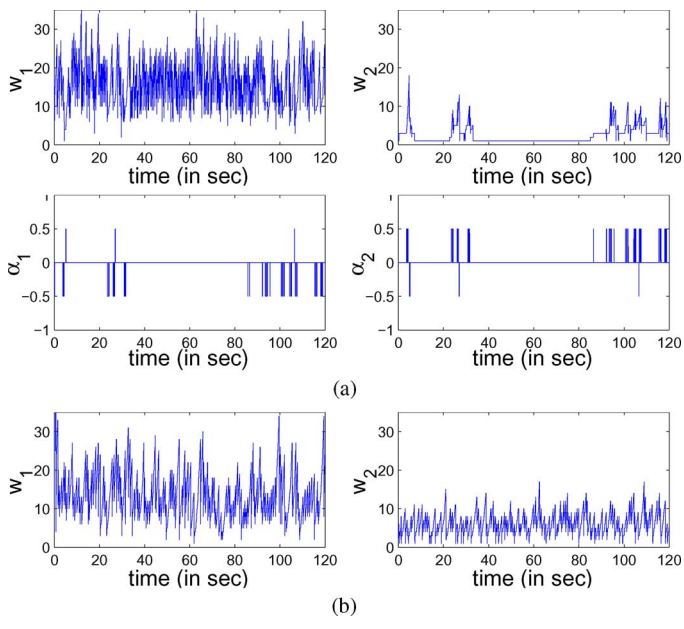


Fig. 8. Evolution of w and α for a two-paths flow. The first path is shared with five TCP flows, and the second with 10. OLIA uses only the good path. LIA transmits significant traffic over the congested path and less than OLIA over the good path. (a) MPTCP–OLIA: window size and α_r as a function of time. (b) MPTCP–LIA: window size.

Losses occur at rate $p_r x_r$ on r, and the user decreases w_r by half whenever it detects a loss. We consider a fluid approximation of OLIA in which we replace the stochastic variations of rates by their expectation. This leads to the differential equation

$$\frac{dx_r}{dt} = x_r^2 \left(\frac{1/\mathrm{rtt}_r^2}{\left(\sum_{p\in\mathcal{R}_u} x_p\right)^2} - \frac{p_r}{2} \right) + \frac{\alpha_r}{\mathrm{rtt}_r^2}. \tag{7}$$

α_r depends on the values p_p and w_p for all paths $p \in \mathcal{R}_u$ of users u. It is defined by (6). To compute α_r, we approximate ℓ_r by its average: $l_r = 1/p_r$.

For a user u, the set of best paths $\mathcal{B}_u$ and the set of paths with maximum window size $\mathcal{M}_u$ depend noncontinuously on the probability of loss on each route, as well as on the various window sizes of the routes of this user. This implies that the right-hand side of (7) is not a continuous function of x_r. Therefore, this differential equation is not well defined and can have no solutions. A natural way to deal with a differential equation with a discontinuous right-hand size is to replace the differential (7) by a differential inclusion $dx/dt \in F(x)$ where the discontinuous α_r of (7) is replaced by the convex closure of the possible values of α_r in a small neighborhood of x [24], [25].

We show in Appendix-C that the differential inclusion corresponding to (7) is

$$\frac{dx_r}{dt} = x_r^2 \left(\frac{1/\mathrm{rtt}_r^2}{\left(\sum_{p\in\mathcal{R}_u} x_p\right)^2} - \frac{p_r}{2} \right) + \frac{\bar{\alpha}_r}{\mathrm{rtt}_r^2} \tag{8}$$

where $\bar{\alpha} = (\bar{\alpha}_1 \dots \bar{\alpha}_{|\mathcal{R}_u|})$ is such that

$$(\bar{\alpha}_r \cdot |\mathcal{R}_u|) \in \begin{cases} [\mathbf{1}_{|\mathcal{B}_u|=1}, 1], & \text{if } r \in \mathcal{B}_u \setminus \mathcal{M}_u \\ [-1, -\mathbf{1}_{|\mathcal{M}_u|=1}], & \text{if } r \in \mathcal{M}_u \setminus \mathcal{B}_u \\ [-\mathbf{1}_{|\mathcal{B}_u|\geq 2}, \mathbf{1}_{|\mathcal{M}_u|\geq 2}], & \text{if } r \in \mathcal{M}_u \cap \mathcal{B}_u \\ \{0\}, & \text{if } r \notin \mathcal{M}_u \cup \mathcal{B}_u \end{cases} \quad (9)$$

with $\sum_{r\in\mathcal{R}_u} \bar{\alpha}_r = 0$ and $\sum_{r\in\mathcal{B}_u} \bar{\alpha}_r = 1/|\mathcal{R}_u|$ if $\mathcal{B}_u \cap \mathcal{M}_u = \emptyset$. The notation $\mathbf{1}_{|\mathcal{B}_u|=1}$ means that this term is equal to 1 if $|\mathcal{B}_u| = 1$, and 0 otherwise. For example, when there is only one best path (i.e., $|\mathcal{B}| = 1$), $\bar{\alpha}_r = 1/|\mathcal{R}_u|$ for $r \in \mathcal{B}_u \setminus \mathcal{M}_u$. If there are two or more best paths (i.e., $|\mathcal{B}| \neq 1$), then $\bar{\alpha}_r \in [0, 1/|\mathcal{R}_u|]$ for $r \in \mathcal{B}_u \setminus \mathcal{M}_u$.

Note that there are multiple $\bar{\alpha}$ that correspond to definition (9). The differential inclusion might have multiple solutions, but this does not affect our analysis.

B. Pareto Optimality of OLIA

A fixed point of the congestion control algorithm (8) is a vector of rates $x = (x_1 \dots x_{|\mathcal{R}|})$ such that there exists $\bar{\alpha}$ satisfying (9) and such that (8) is equal to zero for any route r. We say that x is a nondegenerate allocation of rates if each user transmits with a nonzero rate on at least one of its paths. In practice, due to reestablishment routines in traditional TCP, the allocation of rates will not be degenerate. Hence, in our analysis, we consider only the nondegenerate fixed points and analyze their properties.

Theorem 1: Any nondegenerate fixed point x of OLIA congestion control algorithm, given by (8), has the following properties.

(i) Only the best paths will be used, i.e., paths r with maximum $\sqrt{2/p_r}/\mathrm{rtt}_r$.

(ii) The total rate obtained by a user u is equal to the rate that a regular TCP user would receive on the best path available to u

$$\sum_{r\in\mathcal{R}_u} x_r = \max_{r\in\mathcal{R}_u} \frac{1}{\mathrm{rtt}_r}\sqrt{\frac{2}{p_r}}.$$

Proof: The proof is given in Appendix-D. □

This theorem implies the following corollary:

Corollary 2: OLIA satisfies the three design goals suggested by the RFC [10].

Proof: The proof is given in Appendix-E. □

The following theorem gives a global optimality property of OLIA. For a rate allocation x, we define the total congestion cost by $C(x) = \sum_\ell \int_0^{\sum_{r\ni\ell} x_r} p_\ell(y)dy$.

Theorem 3: Any nondegenerate fixed point x of our congestion control algorithm (8) is Pareto-optimal, i.e.:

- It is impossible to increase the quantity $\sum_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2$ for some users without decreasing it for others or increasing the congestion cost $C(x)$.

Proof: The proof is given in Appendix F. □

Remark 1: If the probability p_ℓ is sharp around C_ℓ, i.e., if $p_\ell(y) \approx 0$ when $y < C_\ell$ and p_ℓ grows rapidly when y exceeds C_ℓ, then the cost C is a binary function: It is very small if the capacity constraints $\sum_{r\in\ell} x_r \leq C_\ell$ are respected, and grows rapidly otherwise. In this case, Theorem 3 shows that if x is a fixed point of our algorithm, it is impossible to increase the quantity $\sum_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2$ for some users without decreasing it for others while respecting the capacity constraints.

Remark 2: As pointed out by Kelly [2], as $C(x)$ is an increasing function of rates, single-path congestion control mechanisms are always Pareto-optimal and the choice of an allocation of rates is only a matter of fairness. However, if we have multiple paths, it is likely that an algorithm will lead to a non-Pareto-optimal allocation [2]. Theorem 3 guarantees that this cannot happen with OLIA. As a consequence, our algorithm will not exhibit either problem P1 nor P2.

Remark 3: Although the utility function of each user $\sum_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2$ could appear to be an *ad hoc* utility function, it reflects the fact that like TCP, OLIA favors paths with low rtt. When all paths belonging to a user have the same RTT, this theorem implies that the rate allocation of OLIA is such that one user cannot increase its rate without decreasing the rate of some other users. Hence, OLIA can successfully avoid problems P1 and P2. When RTTs over paths available to a user are different, satisfying goals 1 and 2 of the RFC [10] can lead to sending traffic on paths that are not the least congested but have a small round-trip times. Therefore, using a TCP-compatible algorithm, it is not possible to avoid problems P1 and P2 in all possible settings. However, we can see from Theorem 1 that by using OLIA, only the best paths available to a user would be used. This indicates that OLIA provides an allocation as close as or closer to the optimal than any TCP-compatible algorithm. To completely avoid problems P1 and P2, it is necessary to depart from the compatibility with regular TCP by using congestion mechanisms that are less sensitive to round-trip times, such as CUBIC [26] or STCP [27].

C. TCP Compatibility

As we show in Appendix-F, OLIA maximizes the utility function $V^*(x)$ given by (17). We now show that our algorithm is fair with the regular TCP under the assumption (A): All the paths belonging to a user u have the same RTT rtt_u. Under this assumption, $V^*(x)$ simplifies as follows:

$$V(x) = \sum_{u\in\mathcal{U}} -\frac{1}{\mathrm{rtt}_u^2 \sum\limits_{r\in\mathcal{R}_u} x_r} - \frac{1}{2}\sum_{l\in\mathcal{L}} \int_0^{\sum_{r\ni l} x_r} p_\ell(x)dx$$

where x is the set of all the rates of the users.

Theorem 4: Under the assumption (A), the congestion control algorithm defined by (8) converges to a maximum of the utility function V

$$\lim_{t\to\infty} V(x(t)) = \max_{x\geq 0} V(x).$$

Proof: The proof is given in Appendix-G. □

This implies that OLIA maximizes the same utility function as the regular TCP of [28], where we replace the rate of a connection by the total rate that a user achieves on all its paths. If the probabilities of loss p_ℓ are sharp around C_ℓ, then our algorithm converges to an optimum of the following global maximization problem:

$$\max \sum_{u\in\mathcal{U}} -\frac{1}{\mathrm{rtt}_u^2 \sum\limits_{r\in\mathcal{R}_u} x_r} \quad \text{subject to} \begin{cases} \sum\limits_{r\ni\ell} x_r \leq C_\ell \\ x_r \geq 0. \end{cases}$$

This is analog to the TCP maximization problem.

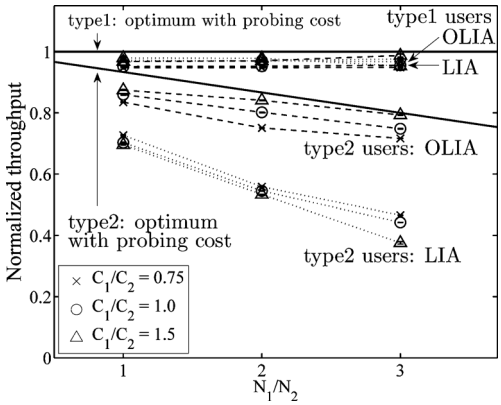


Fig. 9. Scenario A—Normalized throughput of type1 and type2 users: We compare performance of LIA and OLIA. By using OLIA, type2 users achieve up to 2 times higher rates. OLIA performs close to the theoretical optimum with probing cost.

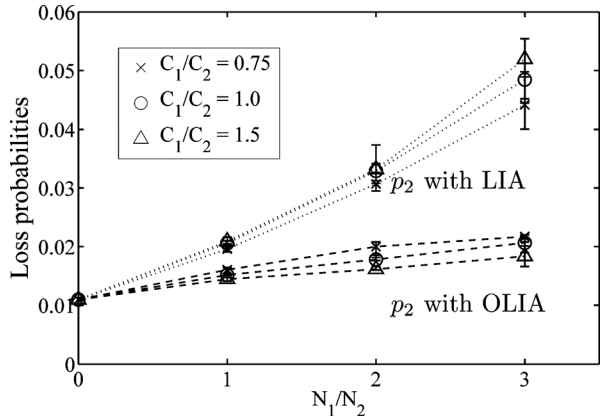


Fig. 10. Scenario A—Loss probability p_2 at shared AP: We observe that OLIA significantly reduces the congestion level at this bottleneck and improves the congestion balancing.

VI. OLIA Evaluation: Measurements and Simulations

In this section, we study the performance of MPTCP with OLIA through measurements and by simulations. We first perform measurements on our testbed to show that OLIA outperforms LIA in all the scenarios from Section III, as evidence that OLIA solves problems P1 and P2. Results from this section are in line with our theoretical analysis from Section V. We then study the performance of OLIA in a data center by using htsim simulator [7].

A. Performance of OLIA in Scenarios A–C

In this section, we study the performance of MPTCP with OLIA, in the scenarios A–C described in Sections III-B–III-D. We show that, in practice, OLIA is very close to the theoretical optimum with probing cost. These results are obtained through measurements over our testbed by using our Linux implementation of OLIA.

1) Scenario A: We have shown in Section III-B that when the addition of an extra link does not help (like in Scenario A), using MPTCP with LIA can reduce the throughput of competing TCP users. Here, we show by measurements that MPTCP with OLIA significantly outperforms MPTCP with LIA and comes close to the theoretical optimum with probing cost. Figs. 9 and 10 report measurements obtained on the testbed shown in Fig. 2. Fig. 9 depicts the normalized throughput of type1 and type2 users that use LIA or OLIA. The results show that OLIA performs close to an optimal multipath algorithm that transmits the minimum traffic over congested paths (theoretical optimum with probing cost). OLIA significantly outperforms LIA: By using OLIA, type2 users achieve rates up to two times higher than with LIA, with no reduction for type1 users.

Fig. 10 depicts the measured loss probability p_2 on the shared access point. We observe that OLIA balances congestion much better than LIA. When we use OLIA, p_2 increases only by a factor of 1.3 in the worst case, whereas with LIA, p_2 increases by a factor of 5. p_1 is almost the same when using LIA or OLIA.

2) Scenario B: We now show the performance of OLIA in the scenario B described in Section III-C. As we have shown, OLIA is Pareto-optimal. Hence, taking into account the minimum probing cost, we expect only 3% reduction in the Blue users' rates and in the aggregate throughput when we upgrade Red users to OLIA [see Fig. 4(b)].

TABLE II
Measurement Results for Scenario B

Red users	Rate/user		Aggregate
	Blue users	Red users	
Single-path	2.2	1.8	59.3
Multipath	2.2	1.7	57.8

Using OLIA, we observe a small drop of 3.5% in the aggregate throughput, which is due to the overhead of minimum traffic $(1/\mathrm{rtt})$ over the congested path. Compared to LIA (see Table I), we see significant improvement.

Table II presents the measurements for the scenario described in Section III-C using OLIA. We set $C_X = 27$, $C_T = 36$, and $C_Z = 100$, all in megabits per second. We have 15 Red and 15 Blue users. We set RTTs to 150 ms over all paths. Our results show that there is a 3.5% decrement in aggregate throughput when we update Red users to OLIA, which is much smaller than the 13% reduction we observed when we used LIA (see Table I). This 3.5% reduction in the aggregate throughput is due to the minimum traffic transmitted by users over congested paths and cannot be reduced as it is bounded below by $1/\mathrm{rtt}$ packets/s.

3) Scenario C: Finally, we study the performance of MPTCP with OLIA in scenario C described in Section III-D. Theorems 1 and 4 imply that by using our algorithm, multipath users do not send any traffic on their path crossing AP2. Next, we show by measurements that OLIA provides a fair allocation among users and performs close to an optimal algorithm with probing cost [Fig. 5(b), dashed lines].

Fig. 11 depicts the normalized throughput of single-path and multipath users, as a function of N_1/N_2 and for $C_1/C_2 = 1,\ 2$. We show the results for LIA and OLIA, as well as for an optimal algorithm with minimum probing cost. This figure shows that with OLIA multipath users transmit only one packet per RTT over AP2. Compared to LIA, type2 users receive up to 2 times higher throughput. Hence, OLIA is less aggressive than LIA toward regular TCP users.

Fig. 12 shows the measured loss probability p_2. The results show again that OLIA balances congestion in the network and reduces the loss probability in bottlenecks much better than LIA. In particular, we observe that by increasing N_1 from 0 to $3N_2$, p_2 increases by a factor of 2 using OLIA, whereas the increase

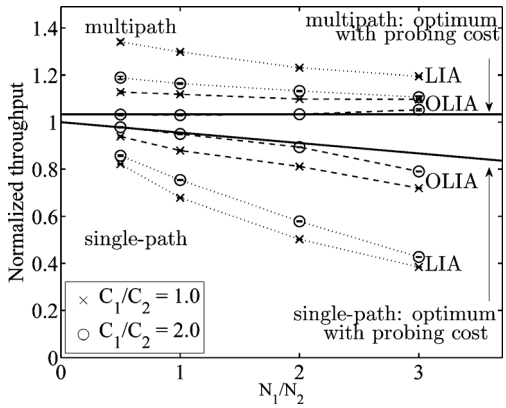


Fig. 11. Scenario C—Normalized throughput of single-path and multipath users: We compare the performance of LIA and OLIA. We observe that by using OLIA, type2 users achieve up to 2 times higher rates. OLIA performs close to the theoretical optimum with probing cost.

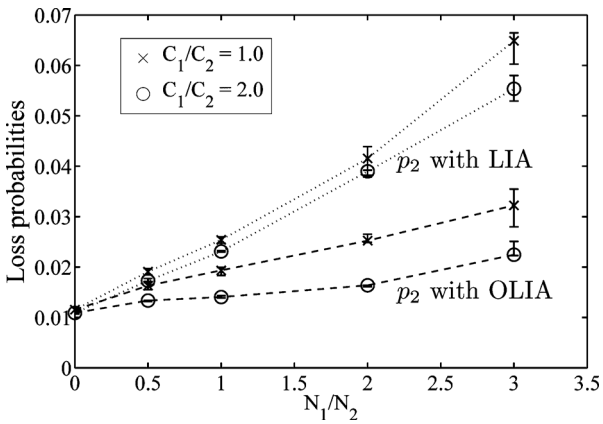


Fig. 12. Scenario C—Loss probability p_2 at shared AP: We observe that OLIA significantly reduces the congestion level at this bottleneck (4–6 times lower compared to LIA).

is in the order of 4–6 times when using LIA. p_1 is almost the same when using OLIA or LIA.

B. Performance of OLIA in Data Center and Dynamic Scenarios

The three preceding examples show that by providing a better congestion balance, MPTCP with OLIA outperforms MPTCP with LIA in Scenarios A–C. In this section, we show that by being nonflappy and as responsive as LIA, OLIA can fully use the multiple paths available in a data center. Our study is based on a series of scenarios in which MPTCP with LIA is studied in [7]. Because of space constraints, we present the results for only two of the cases where LIA was shown to be very efficient. We observe that OLIA performs as well or better than LIA in these two scenarios. This indicates that it is not flappy and has a very good responsiveness. These results are obtained using htsim simulator used in [7], provided by Raiciu *et al.* We implemented OLIA in the simulator and use the same scenarios as [7].

1) Static FatTree Topology: We first study exactly the same scenario as in [7, Section 4.2]: The network is a FatTree with 128 hosts, 80 eight-port switches, 100-Mb/s links. Each host sends a long-lived flow to another host chosen at random. Fig. 13(a) shows the aggregate throughput achieved by long-lived TCP

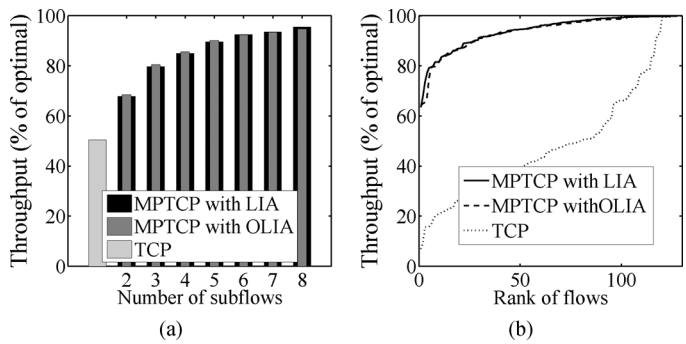


Fig. 13. Performance of OLIA in a FatTree with many possible parallel paths between users. OLIA successfully explores the path diversity and uses the available capacity (a sign of nonflappiness). LIA performs similarly as, in this scenario, it can successfully balance the congestion. (a) Aggregated throughput. (b) Throughput of users.

TABLE III
PERFORMANCE OF OLIA IN A HIGHLY DYNAMIC SETTING

algorithm	Short flow finish time (mean/stdev)	Network core utilization
MPTCP - LIA	98 ± 57 ms	63.2%
MPTCP - OLIA	90 ± 42 ms	63%
Regular TCP	73 ± 57 ms	39.3%

OLIA uses the available capacity as efficient as LIA, but decreases the average completion time of short flows by 10%.

and MPTCP (LIA and OLIA) flows. We show the results for different numbers of subflows used. Our results show that OLIA can successfully exploit the multiple paths that exist in the network and can use the available capacity. This is a sign that it is not flappy. Regular TCP shows a poor performance. Fig. 13(b) shows the throughput of individual users ranked in order of achieved throughputs, for LIA and OLIA with eight subflows per user and with TCP; LIA and OLIA provide similar fairness among users and are more fair than TCP. We observe that, in this scenario, LIA performs close to an optimal algorithm and exhibits a similar performance to OLIA. The reason is that the users have multiple equally good paths. Hence, LIA also successfully balances the congestions in the network, similarly to OLIA, and performs optimally. During the experiments, we measured the loss probabilities of links available to users. The results confirm our reasoning: For this scenario, the obeserved loss probabilities are similar on all paths.

2) Dynamic Setting With Short Flows: We study the same scenario as the one described in [7, Section 4.3.4]. The scenario is a 4:1 oversubscribed FatTree where each host sends to one other host. One third of the hosts send a continuous flow by using either TCP, MPTCP with LIA (eight subflows), or MPTCP with OLIA (eight subflows). The remaining hosts send short flows of size 70 kB every 200 ms on average (they generate these flows according to a Poisson process). They use regular TCP. This is a highly dynamic setting in which changes occur in the order of milliseconds. Table III shows the average completion time for short flows and the network core usage. Fig. 14 shows the distribution of completion times of short flows. Our results show that although OLIA uses the available capacity as efficiently as LIA, the average completion time of short flows decreases by 10% using OLIA. Moreover, we observe in Fig. 14 that OLIA decreases the completion time of both fast and slow short flows. For slow flows, the decrease is more than 25%. This

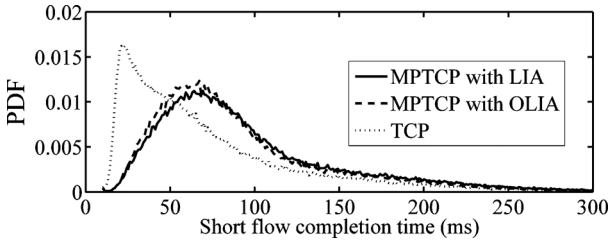


Fig. 14. Completion time of short flows competing with long-lived TCP, MPTCP with LIA, or MPTCP with OLIA flows in a highly dynamic setting. OLIA reacts faster to the changes in the network and is fairer toward short flows.

shows that OLIA has a better responsiveness than LIA, is more fair to TCP users, and uses capacity quickly when it is available. With TCP, we have a lower average completion time for short flows, but very low network utilization.

VII. Conclusion

We have shown that MPTCP with LIA suffers from important performance issues. Moreover, it is possible to build an alternative to LIA, which performs close to an optimal algorithm with probing cost while being as responsive and nonflappy as LIA. Our theoretical results show that our proposed algorithm, OLIA, is Pareto-optimal and satisfies the three design goals of MPTCP [10]. Moreover, we have shown through measurements and by simulation that OLIA is as responsive and nonflappy as LIA, and that it solves identified problems with LIA.

Multiple directions could be explored to go further. The first one comes from the fixed-point analysis of Theorem 3. The stability and convergence of OLIA is another important question that will be studied in future work. Another one would be to vary the minimum probing traffic rate by an adjustment of the retransmit timer or by discarding bad paths from the set of available paths. Also, we plan to perform more detailed experiments to include other factors such as background traffic, flow durations, and receive window limitations.

Appendix

These Appendixes are divided in two parts. The first part (Appendixes-A and B) focuses on the proofs of the analytical results for LIA. It contains the fixed-point analysis and the computation of the optimal allocation with probing cost for scenarios A–C. The second part (Appendixes C–G) contains the proofs related to the Pareto optimality of OLIA.

A. Fixed-Point Analysis for Scenario A

In this Appendix, we present a fixed-point analysis of the scenario A of Section III. For more clarity, we represent the scenario A in Fig. 15.

Recall that p_1 and p_2 are the loss probabilities at the link connected to the streaming server and at the shared AP. Also, we assume that the private APs are not the bottlenecks, hence the loss probabilities at the private APs are negligible. We provide the analysis for the case where RTTs are the same over all connections and equal to rtt. x_1 is the rate of a type1 user over the path that crosses its private AP, and x_2 is its rate over the shared AP. y is the rate of a type2 user.

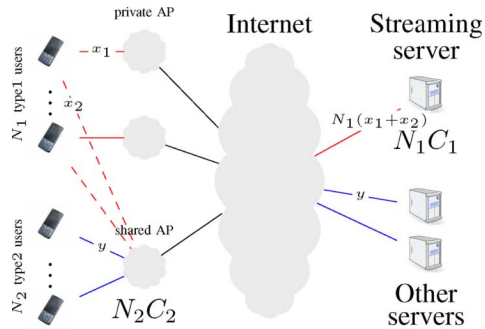


Fig. 15. Scenario A of Section III.

1) MPTCP With LIA: The type1 users use MPTCP with LIA on two paths with loss probabilities p_1 and $p_1 + p_2$. Thus, the fixed-point formula [(2)] of LIA gives

$$x_1 + x_2 = C_1 = \frac{1}{\mathrm{rtt}}\sqrt{\frac{2}{p_1}} \quad \text{and} \quad x_2 = \frac{1}{2 + p_2/p_1}\frac{1}{\mathrm{rtt}}\sqrt{\frac{2}{p_1}}.$$

Users of type2 are using the regular TCP over a link with probability of loss p_2; they get a throughput

$$y = \frac{1}{\mathrm{rtt}}\sqrt{2/p_2}.$$

This comes from the loss-throughput formula for TCP.

As the link connected to the streaming server and shared AP are the bottlenecks, the capacity constraints give

$$N_1(x_1 + x_2) = N_1C_1 \quad \text{and} \quad N_1x_2 + N_2y = N_2C_2.$$

Let $z := \sqrt{p_1/p_2}$. A direct computation shows that z is a root of

$$z + \frac{z^2}{1 + 2z^2}\frac{N_1}{N_2} = \frac{C_2}{C_1}. \tag{10}$$

As $z^2/(1+2Z^2)$ is an increasing function of z, this equation has a unique positive solution. Although this solution has no simple closed-form solution (it is the root of a third-order polynomial), it can be easily computed numerically. Hence, It provides a numerical scheme for computing x_1, x_2, and y.

Type1 users always receive a rate of C_1; hence, their normalized throughput, $(x_1 + x_2)/C_1$, is always 1. The normalized throughput of type2 users, y/C_2, is equal to $\sqrt{p_1/p_2}\sqrt{2/p_1} = zC_1$, where z is the unique positive solution of (10). In particular, this shows that y/C_2 only depends on the ratios C_1/C_2 and N_1/N_2.

2) Optimal With Probing Cost: In scenario A, the throughput of type1 users is bounded by the streaming server. Using the shared AP can reduce the throughput of type2 users, but cannot bring any gain to type1 users. Thus, an optimal algorithm should put as low traffic as possible on the second path. Assuming that the minimum traffic sent over a link is one packet of size MSS per round-trip time, this leads to the following allocation of rate:

$$x_1 + x_2 = C_1 \quad \text{and} \quad x_2 = \frac{\mathrm{MSS}}{\mathrm{rtt}}$$
$$y = C_2 - \frac{N_1}{N_2}\frac{\mathrm{MSS}}{\mathrm{rtt}}.$$

This allocation is represented by the solid lines in Fig. 1(b).

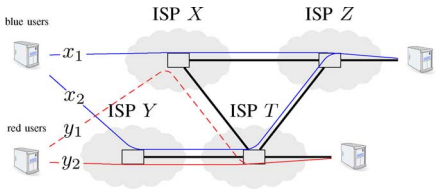


Fig. 16. Scenario B of Section III.

B. Fixed-Point Analysis for Scenario B

We present a theoretical analysis of the throughput achieved by Blue and Red users when multipath users use MPTCP with LIA and when users use optimal algorithm with probing cost. We represent scenario B in Fig. 16. We assume that the capacities of link Y and link Z are greater than C_X+C_T. This ensures that only links X and T are bottlenecks, and we denote by p_X and p_T the probabilities of loss over them.

1) MPTCP With LIA: If Red users are only connected to Y, the theoretical analysis is the same as the one of scenario C, and we refer to Section III-D for more details. In the case where all paths are activated, i.e., when Red users upgrade to MPTCP users, the loss throughput formula [(2)] for LIA shows that the throughput of the different connections are

$$\begin{cases} y_1 = \frac{1/\mathrm{rtt}}{2+\frac{p_X}{p_T}}\sqrt{\frac{2}{p_T}} \\ y_2 = \frac{p_X+p_T}{p_T}y_1 \end{cases}, \quad \begin{cases} x_1 = \frac{1/\mathrm{rtt}}{1+p_X/p_T}\sqrt{\max\frac{2}{p_X},\frac{2}{p_T}} \\ x_2 = \frac{1/\mathrm{rtt}}{1+p_T/p_X}\sqrt{\max\frac{2}{p_X},\frac{2}{p_T}} \end{cases}.$$

Moreover, as ISP X and Y are bottlenecks, we have

$$C_X = N(x_1+y_1) \quad \text{and} \quad C_T = N(x_2+y_1+y_2)$$

and $y_1+y_2 = \sqrt{2/p_T}/\mathrm{rtt}$.

Let us first assume that $p_X > p_T$. In that case, we have $x_1 + x_2 = \sqrt{2/p_T}$. Let $z = p_X/p_T$. The capacity constraints imply

$$C_X = \frac{1/\mathrm{rtt}}{1+z}\sqrt{\frac{2}{p_T}} + \frac{1/\mathrm{rtt}}{2+z}\sqrt{\frac{2}{p_T}}$$
$$C_T = \frac{z/\mathrm{rtt}}{1+z}\sqrt{\frac{2}{p_T}} + \frac{1}{\mathrm{rtt}}\sqrt{\frac{2}{p_T}}.$$

This implies that

$$2z^2 + z\left(5-2\frac{C_T}{C_X}\right) + 2 - 3\frac{C_T}{C_X}.$$

This equation has only one positive root. This root is greater than one only when $C_X/C_T < 5/9$. Thus, p_X/p_T is the root of this second-order polynomial in this case.

When $C_X/C_T > 5/9$, we must have $p_T > p_X$. A similar computation as above shows that in this case z is the unique positive root of the fifth-order polynomial

$$z^5+z^4+z^3\left(3-\frac{C_T}{C_X}\right)+z^2\left(2-\frac{C_T}{C_X}\right)+z\left(2-\frac{C_T}{C_X}\right)-2\frac{C_T}{C_X}.$$

These equation provide an efficient numerical method to evaluate the rate sent over the various links and therefore evaluate the performance of LIA. Note that he solutions of these equation only depend on C_T/C_X.

2) Optimal With Probing Cost: To simplify the notations, we present the analysis for $N_{\mathrm{B}} = N_{\mathrm{R}} = N$, which is the case in the scenarios studied in Section III-C. The analysis is similar when $N_{\mathrm{B}} \neq N_{\mathrm{R}}$. We distinguish two cases: first when Red users use the regular TCP, then when Red users uses an optimal multipath algorithm and activate the dashed connection.

Case 1: Red users are only connected to ISP Y: As ISP Y and Z are not bottlenecks, we have $x_1 = C_X/N$. Moreover, the capacity constraint for ISP T implies that $N(x_2+y_2) = C_T$. Assuming that $x_2 \geq \mathrm{MSS}/\mathrm{rtt}$, there are two scenarios.

- When $C_X \leq C_T - N\mathrm{MSS}/\mathrm{rtt}$, a fair allocation will allocate the same rate, i.e., $(C_X+C_T)/(2N)$, to all users.
- When $C_X > C_T - N\mathrm{MSS}/\mathrm{rtt}$, Blue users will get more than Red users. Thus, Blue users should only transmit the minimal traffic $x_2 = \mathrm{MSS}/\mathrm{rtt}$ over the second link.

This shows that using an optimal algorithm with probing, each Blue user will get a rate x_1+x_2 and each Red user will get a rate y_2, where

$$x_1+x_2 = \max\left(\frac{C_X}{N}+\frac{\mathrm{MSS}}{\mathrm{rtt}}, \frac{C_T+C_X}{2N}\right) \tag{11}$$
$$y_2 = \min\left(\frac{C_T}{N}-\frac{\mathrm{MSS}}{\mathrm{rtt}}, \frac{C_X+C_T}{2N}\right). \tag{12}$$

Case 2: Red users activate the dashed connection: As y_1 and y_2 share the same bottleneck, ISP T, the Red users should only transmit the minimum traffic over the dashed path, i.e., $y_1 = \mathrm{MSS}/\mathrm{rtt}$. If the Red users transmit over the dashed path, they will penalize the other users without any benefit for themselves. This implies that $x_1 = C_X/N - \mathrm{MSS}/\mathrm{rtt}$. Also, the capacity constraints for ISP T gives $N(x_2+y_1+y_2) = C_T$. Therefore, we have: $N(x_1+x_2+y_1+y_2) = C_T+C_X-N\mathrm{MSS}/\mathrm{rtt}$. As $x_2 \geq \mathrm{MSS}/\mathrm{rtt}$, a fair allocation should allocate x_2 such that:

- if $C_X \leq C_T - N\mathrm{MSS}/\mathrm{rtt}$, we should have $x_1+x_2 = y_1+y_2 = (C_T+C_X-N\mathrm{MSS}/\mathrm{rtt})/(2N)$;
- if $C_X \geq C_T - N\mathrm{MSS}/\mathrm{rtt}$, Blue users should transmit the minimal traffic $x_2 = \mathrm{MSS}/\mathrm{rtt}$ over their second link.

Thus, using this optimal algorithm with probing cost, each Blue user will get a rate x_1+y_2 and each Red user will get a rate y_1+y_2 where

$$x_1+x_2 = \max\left(\frac{C_X}{N}, \frac{C_T+C_X}{2N}-\frac{\mathrm{MSS}}{2\mathrm{rtt}}\right) \tag{13}$$
$$y_1+y_2 = \min\left(\frac{C_T}{N}-\frac{\mathrm{MSS}}{\mathrm{rtt}}, \frac{C_X+C_T}{2N}-\frac{\mathrm{MSS}}{2\mathrm{rtt}}\right). \tag{14}$$

Compared to (11) and (12), the rates obtained by (13) and (14) are strictly smaller. The agregate throughput of all users decreases by $N\mathrm{MSS}/\mathrm{rtt}$.

3) Illustrations for Two Values of RTT: Fig. 17 depicts the throughput reduction when upgrading Red users to multipath for an optimal algorithm with probing cost. The values are shown for $C_X = 27$ Mb/s, $C_T = 36$ Mb/s, and $N_{\mathrm{B}} = N_{\mathrm{R}} = 15$ users. The values of the MSS is 1500 B. As the minimal probing traffic sent over a link is $\mathrm{MSS}/\mathrm{rtt}$, a lower value of the RTT means a higher reduction of throughput.

C. Construction of the Differential Inclusion

1) Brief Introduction on Differential Inclusions: In this section, we briefly recall some definitions and results about differential inclusions and their relation to stochastic systems that have discontinuous drifts.

A set-valued function $F : \mathbb{R}^d \to \mathcal{S}(\mathbb{R}^d)$ is a function that associates to each vector $x \in \mathbb{R}^d$ a set of vectors $F(x) \subset \mathbb{R}^d$. We say that a function $x : [0,T] \to \mathbb{R}^d$ is a solution of the differential inclusion $dx/dt \in F(x)$ on the interval $[0,T]$ if there exists a function $f : [0,T] \to \mathbb{R}^d$ such that

$$\forall t \in [0,T] : x(t) = x(0) + \int_0^t f(s)ds \quad \text{with} \quad f(t) \in F(t).$$

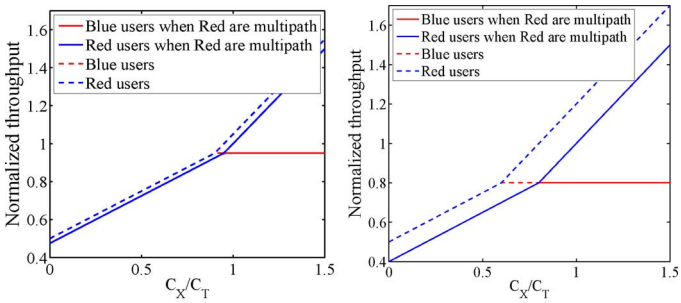


Fig. 17. Illustration of the optimal allocation with probing for scenario B for two values of the RTT. We set $C_T = 36$ Mb/s and $N_B = N_R = 15$ users. (a) RTT $= 100$ ms. (b) RTT $= 25$ ms.

In particular, this implies that x is differentiable for almost every t and its derivative x' satisfies $x'(t) \in F(x(t))$.

Differential inclusion provide a natural way to represent differential equation with discontinuous right-hand side. Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a single-valued function. Following [25], we define the set-valued function F corresponding to f

$$F(x) = \bigcap_{\varepsilon \to 0} \text{convex\_closure}\left\{f(x) : \|x - w\| \le \varepsilon\right\}.$$

This definition guarantees that the differential inclusion $dx/dt \in F(x)$ has at least one solution. Moreover, it has been shown in [25] that the solution of differential inclusions are a good approximation of the stochastic systems with discontinuous drift, such as (7).

2) Computation of (9): In this section, we show how to obtain the conditions on α_r given by (9) and how to compute the differential inclusion (8) from the differential (7).

The only noncontinuous part of the ODE (7) is due to α_r. The set-valued function $\bar{\alpha}$ corresponding to α is

$$\bar{\alpha}(w) = \bigcap_{\varepsilon \to 0} \text{convex\_closure}\left\{\alpha(x) : \|x - w\| \le \varepsilon\right\}.$$

The computation of $\bar{\alpha}$ can be done by a careful inspection of Fig. 18. For a route r, the set $\bar{\alpha}_r$ corresponds to the convex closure of the values that α_r can take when all the points $(w_r, p_r/\text{rtt}_r)$ move in a small neighborhood. We detail the computation for a link $r \in \mathcal{B}_u \backslash \mathcal{M}_u$. The other cases ($r \in \mathcal{M}_u \backslash \mathcal{B}_u$ and $r \in \mathcal{M}_u \cap \mathcal{B}_u$ and $r \notin \mathcal{M}_u \cup \mathcal{B}_u$) are similar.

Let r be a route in $\mathcal{B}_u \backslash \mathcal{M}_u$. Let first assume that there are two or more best paths (*e.g.,* this is the case for the route r_4 of Fig. 18), then if all points move in a small neighborhood (represented by the dotted circles around nodes on Fig. 18), then there are some situations for which this route will be the only route in $\mathcal{B}_u \backslash \mathcal{M}_u \neq \emptyset$, and α_r will be $1/|\mathcal{R}_u|$ in that case. In other situations, the only best route can be route r_1, and in that case $\alpha_r = 0$. Since this route cannot become a route with maximum window size, α_r can take any value in $[0, 1/|\mathcal{R}_u|]$.

On the other hand, if there is only one best path and if $r \in \mathcal{B}_u \backslash \mathcal{M}_u$, then r is the best path (this would be the case for the route r_4 on Fig. 18 if the node r_1 did not exist). In that case, r will always be in $\mathcal{B}_u \backslash \mathcal{M}_u$ and $\alpha_r = 1/|\mathcal{R}_u|$.

This shows the first line of (9): For $r \in \mathcal{B}_u \backslash \mathcal{M}_u$

$$\bar{\alpha}_r \in [0, 1/|\mathcal{R}_u|] \text{ if } |\mathcal{B}_u| \neq 1 \quad \text{and} \quad \bar{\alpha}_r = 1/|\mathcal{R}_u| \text{ otherwise.}$$

The proofs of the other cases of (9) as they are very similar.

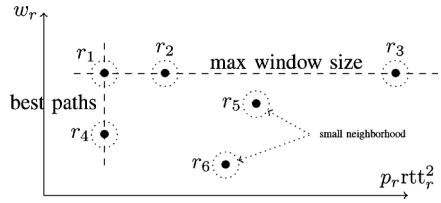


Fig. 18. State of the routes of a user that has 6 routes at a given time. Each route is represented by a point •, its x-coordinate being the ratio $p_r \text{rtt}_r^2$ of a link (the inverse of its hypothetical rate for a single regular TCP flow) and its y-coordinate being its window size. The dotted circles around each node represent a small neighborhood of the points. In this example, the routes r_1, r_2, r_3 are routes with maximum window size: $r_1, r_2, r_3 \in \mathcal{M}_u$. The routes r_1, r_4 are best routes: $r_1, r_4 \in \mathcal{B}_u$. Finally, we have $r_5, r_6 \notin \mathcal{B}_u \cup \mathcal{M}_u$.

D. Proof of Theorem 1

Let x be a nondegenerate fixed point of our algorithm. Recall that a fixed point of the congestion control algorithm (8) is a rate allocation vector x such that there exists $\bar{\alpha}_r$ satisfying (9) such that the quantity dx_r/dt defined by (8) is null.

Proof of (i): Let x be a nondegenerate fixed point of OLIA. For any path $p \in \mathcal{R}_u$, the equation dx_p/dt contains two terms, denoted term A and term B in the following equation:

$$0 = \frac{dx_p}{dt} = x_p^2 \underbrace{\left(\frac{1/\text{rtt}_p^2}{\left(\sum_{s \in \mathcal{R}_u} x_s\right)^2} - \frac{p_p}{2} \right)}_{\text{term } A} + \underbrace{\bar{\alpha}_p}_{\text{term } B}. \qquad (15)$$

Assume that there exists a nonbest path $r \notin \mathcal{B}_u$ such that $x_r > 0$. We show that this leads to a contradiction.

Equation (15) shows that the term A is positive for r and hence is strictly positive for any best paths (by definition of best path). If $\mathcal{B}_u \cap \mathcal{M}_u \neq \emptyset$, there exists a best path p with maximum window size. Thus, we have $x_p \neq 0$, which implies that $dx_p/dt > 0$ as $\bar{\alpha}_p$ is nonnegative. If $\mathcal{B}_u \cap \mathcal{M}_u = \emptyset$, then there exists $p \in \mathcal{B}_u$ such that $\bar{\alpha}_p > 0$, which implies that $\bar{\alpha}_p > 0$ and thus $dx_p/dt > 0$. In both cases, we have $dx_p/dt > 0$, which contradicts that $dx_p/dt = 0$.

This shows that for any nonbest path $r \notin \mathcal{B}_u$, we must have $x_r = 0$.

Proof of (ii): Because of *(i)*, for all routes $r \notin \mathcal{B}_u$, we have $x_r = 0$. This means that for all routes $r \notin \mathcal{B}_u$, we have $r \notin \mathcal{M}_u$ and $\bar{\alpha}_r = 0$. The best paths are the set of paths p with minimum $p_p \text{rtt}_p^2$. Therefore, the term A of (15) is of the same sign for all best paths. This implies that the term $\bar{\alpha}_p$ is of the same sign for all p. As $\sum_{p \in u} \bar{\alpha}_p = \sum_{p \in \mathcal{B}_u} \bar{\alpha}_p = 0$, this implies that $\bar{\alpha}_p = 0$ for all paths $p \in \mathcal{R}_u$.

Therefore, the fixed point x satisfies

$$x_r = 0 \quad \text{or} \quad \sum_{p \in \mathcal{R}_u} x_p = \frac{1}{\text{rtt}_r} \sqrt{\frac{2}{p_r}}. \qquad (16)$$

By assumption, x is nondegenerate, which means that there exists a route $r \in \mathcal{R}_u$ such that $x_r \neq 0$. Because of *(i)*, r is a necessarily a best path. Hence, we have

$$\sum_{p \in \mathcal{R}_u} x_p = \frac{1}{\text{rtt}_r} \sqrt{\frac{2}{p_r}} = \max_p \frac{1}{\text{rtt}_p} \sqrt{\frac{2}{p_p}}.$$

This concludes the proof of *(ii)*. □

E. Proof of Corollary 2

Point *(ii)* of Theorem 1 implies that OLIA satisfies goal 1): The total rate that OLIA gets ($\sum_{r\in\mathcal{R}_u} x_r$) is the same as the rate that a regular TCP would get on its best link ($\max_{r\in\mathcal{R}_u} \sqrt{2/p_r}/\mathrm{rtt}_r$).

Moreover, as OLIA uses only its best paths, it does not transmit more than a regular TCP does on any of its paths and satisfies goal 2). Finally, as OLIA uses only its best path, it perfectly balances congestion and satisfies goal 3). □

F. Proof of Pareto Optimality (Theorem 3)

Let x^* be a fixed point of the algorithm and define the utility function $V^*(x)$ as

$$\sum_{u\in\text{users}} -\frac{1}{\tau_u^2 \sum\limits_{r\in\mathcal{R}_u} \frac{x_r}{\mathrm{rtt}_r^2}} - \frac{1}{2}\sum_{\ell\in\text{links}} \int_0^{\sum\limits_{r\in\ell} x_r} p_\ell(x)dx \qquad (17)$$

where τ_u is defined by: $\tau_u = (\sum_{r\in\mathcal{R}_u} x_r^*)/(\sum_{r\in\mathcal{R}_u} x_r^*/\mathrm{rtt}_r^2)$.

The function V^* is a nonpositive function. Moreover, using that $p_\ell(x)$ is increasing, it goes to $-\infty$ when $x\to\infty$. Therefore, it has a maximum, attained for a finite x. By concavity of V^*, a necessary and sufficient condition for a point x to be a maximizer of $\mathcal{U}$ is that for every route r

$$\frac{\partial V^*}{\partial x_r}(x) \le 0 \quad \text{and} \quad \frac{\partial V^*}{\partial x_r}(x) = 0 \text{ or } x_r = 0.$$

By definition of V^*, this implies that for every r

$$\frac{1}{\tau_u^2}\frac{1/\mathrm{rtt}_r^2}{\left(\sum\limits_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2\right)^2} - \frac{p_r}{2} \le 0 \qquad (18)$$

$$\frac{1}{\tau_u^2}\frac{1/\mathrm{rtt}_r^2}{\left(\sum\limits_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2\right)^2} - \frac{p_r}{2} = 0 \text{ or } x_r = 0. \qquad (19)$$

By definition of τ_u and as x^* satisfies point *(i)* of Theorem 1, (18) holds. Moreover, (16) comes directly from (19).

This shows that x^* is a maximum of the function V^*. Since $V^*(x)$ is an increasing function of $\sum_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2$ and a decreasing function of the congestion cost $C(x)$, it is impossible to increasing $\sum_{r\in\mathcal{R}_u} x_r/\mathrm{rtt}_r^2$ for some users without decreasing it for others or increasing the cost. □

G. Proof of TCP-Compatibility (Theorem 4)

Theorem 4 assumes that all the paths belonging to user u have the same round-trip time rtt_u. Recall that V is

$$V(x) = \sum_{u\in\mathcal{U}} -\frac{1}{\mathrm{rtt}_u^2 \sum\limits_{r\in\mathcal{R}_u} x_r} - \frac{1}{2}\sum_{l\in\mathcal{L}} \int_0^{\sum\limits_{r\in l} x_r} p_\ell(x)dx.$$

By construction of F, there exists at least one solution of the differential inclusion given by (8) (see Appendix-C.1). Let x be one of these solutions. There exists a function $\bar\alpha(t) = (\bar\alpha_1(t)\dots\bar\alpha_{|\mathcal{R}|}(t))$ satisfying (9) for all t and such that dx_r/dt satisfies (8)

$$\frac{dx_r}{dt} = x_r^2\left(\frac{1/\mathrm{rtt}_r^2}{\left(\sum\limits_{p\in\mathcal{R}_u} x_p\right)^2} - \frac{p_r}{2}\right) + \frac{\bar\alpha_r(t)}{\mathrm{rtt}_r^2}.$$

When running the algorithm, the derivative of $V(x(t))$ w.r.t. time satisfies $dV/dt = \sum_{u,r}(\partial V/\partial x_r)(dx_r/dt)$. Thus

$$\frac{d}{dt}V = \sum_{u\in\mathcal{U}}\sum_{r\in\mathcal{R}_u}\frac{\partial V}{\partial x_r}\frac{dx_r}{dt}$$

$$= \sum_{u\in\mathcal{U}}\sum_{r\in\mathcal{R}_u} x_r^2\left(\frac{1}{\mathrm{rtt}_u^2\left(\sum\limits_{p\in\mathcal{R}_u} x_p\right)^2} - \frac{p_r}{2}\right)^2 \qquad (20)$$

$$+ \sum_{u\in\mathcal{U}}\sum_{r\in\mathcal{R}_u}\left(\frac{1}{\mathrm{rtt}_u^2\left(\sum\limits_{p\in u} x_p\right)^2} - \frac{p_r}{2}\right)\frac{\bar\alpha_r}{\mathrm{rtt}_u^2}. \qquad (21)$$

By definition of $\bar\alpha$, we have $\sum_{r\in\mathcal{R}_u}\bar\alpha_r = 0$. Moreover, when all rtt are equal, the best paths are the paths with minimal probability loss and $\bar\alpha_r \le 0$ for such paths. Thus

$$\sum_{r\in\mathcal{R}_u}\bar\alpha_r p_r = \sum_{r\in\mathcal{B}_u}\bar\alpha_r p_r + \sum_{r\notin\mathcal{B}_u}\bar\alpha_r p_r \le \sum_r \bar\alpha_r p_{\min} = 0.$$

These two properties together show that the term (21) is nonnegative. Since (20) is also nonnegative, this shows that $dV(x(t))/dt \ge 0$ for all t. Thus, the function V is nondecreasing. Since V is nonpositive, this shows that $\lim_{t\to\infty} dV(x(t))/dt = 0$.

Let x^* be a limit point of $x(t)$, which exists since $x(t)$ remains in a compact set. Since $\lim_{t\to\infty} dV(x(t))/dt = 0$, this implies that (20) and (21) are equal to 0 for this x^*. In particular, this implies that for all $r\in\mathcal{R}_u$:

$$\frac{1}{\mathrm{rtt}_p^2\left(\sum\limits_{p\in\mathcal{R}_u} x_p^*\right)^2} = \frac{p_r}{2} \text{ or } (x_r = 0 \text{ and } \bar\alpha_r = 0).$$

This shows that x^* is a fixed point of the algorithm. When the RTTs of all paths of a user u are equal to rtt_u, the quantity τ_u defined in the proof of Theorem 3 is equal to rtt_u^2. Thus, the function V^* of the proof of Theorem 3 is equal to V. In particular, V^* does not depend on x^*. Since x^* is a fixed point of the algorithm, x^* is a maximizer of V. □

REFERENCES

[1] M. Allman, V. Paxon, and E. Blanton, "TCP congestion control," RFC 5681, Sep. 2009.

[2] F. P. Kelly, "Mathematical modelling of the Internet," in *Mathematics Unlimited—2001 and Beyond*. Berlin, Germany: Springer-Verlag, 2001, pp. 685–702.

[3] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.
[4] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, 2005.
[5] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity in the Internet," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1260–1271, Dec. 2006.
[6] W. H. Wang, M. Palaniswami, and S. H. Low, "Optimal flow control and routing in multi-path networks," *Perform. Eval.*, vol. 52, no. 2–3, pp. 119–132, 2003.
[7] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handly, "Improving datacenter performance and robustness with multipath TCP," in *Proc. ACM SIGCOMM*, 2011, pp. 266–277.
[8] D. Wischik, M. Handly, and C. Raiciu, "Control of multipath TCP and optimization of multipath routing in the Internet," in *Proc. NetCOOP*, 2009, pp. 204–218.
[9] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handly, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. USENIX NSDI*, 2011, p. 8.
[10] C. Raiciu, M. Handly, and D. Wischik, "Coupled congestion control for multipath transport protocols," RFC 6356 (Experimental), 2011.
[11] UCL, Louvain-la-Neuve, Belgium, "MultiPath TCP—Linux kernel implementation," 2013 [Online]. Available: http://mptcp.info.ucl.ac.be/
[12] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," Univ. Massachusetts, Tech. Rep. UM-CS-2013-018, 2013.
[13] "Multipath TCP (mptcp)," 2013 [Online]. Available: http://datatracker.ietf.org/wg/mptcp/
[14] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development," RFC 6182 (Informational), 2011.
[15] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," IETF Internet Draft, 2011.
[16] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," *Comput. Commun. Rev.*, vol. 38, no. 5, pp. 47–52, 2008.
[17] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," Univ. College London, London, U.K., Tech. Rep., 2009.
[18] C. Cetinkaya and E. W. Knightly, "Opportunistic traffic scheduling over multiple network paths," in *Proc. IEEE INFOCOM*, 2004, pp. 1928–1937.
[19] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," in *Proc. USENIX*, 2004, p. 8.
[20] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda, "Multipath congestion control for shared bottleneck," in *Proc. PFLDNeT Workshop*, 2009.
[21] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.
[22] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. SIGCOMM*, 2000, pp. 151–160.
[23] V. Jacobson, "Congestion avoidance and control," *Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, 1988.
[24] N. Gast and B. Gaujal, "Mean field limit of non-smooth systems and differential inclusions," *Perform. Eval. Rev.*, vol. 38, no. 2, pp. 30–32, 2010.
[25] N. Gast and B. Gaujal, "Markov chains with discontinuous drifts have differential inclusion limits," *Perform. Eval.*, vol. 69, no. 12, pp. 623–642, 2012.
[26] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *Oper. Syst. Rev.*, vol. 42, pp. 64–74, 2008.
[27] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *Comput. Commun. Rev.*, vol. 33, no. 2, pp. 83–91, 2003.
[28] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 689–702, 2003.

Ramin Khalili received the B.S. degree in electrical engineering from the University of Shiraz, Shiraz, Iran, in 1999, the M.S. degree in telecommunication from Sharif University of Technology, Tehran, Iran, in 2001, and the doctorate degree in computer science from University de Pierre et Marie Curie, Paris, France, in 2005.

He is a Senior Scientist Researcher with T-Labs/TU-Berlin, Berlin, Germany. From 2007 to 2008, he was a Postdoctoral Researcher with the University of Massachusetts, Amherst, MA, USA. After that, from 2008 to 2012, he was a Senior Researcher with LCA2-EPFL, Lausanne, Switzerland. His interests include topics in computer networks and wireless communications, with an emphasis on design and optimization of networking protocols.

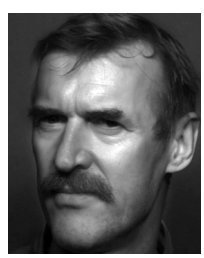
Dr. Khalili has been on the Technical Program Committee of IEEE INFOCOM since 2009. He received the Best Paper Award at ACM E-energy 2011 and ACM CoNEXT 2012.

Nicolas Gast received the Aggregation in Mathematics from École Normale Supérieure, Paris, France, in 2007, and the Ph.D. degree in computer science from the University of Grenoble and INRIA, Grenoble, France, in 2010.

He is currently a Post-Doctoral Fellow with the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. His main interests are in stochastic modeling and control of large systems, using approximation techniques to build online and distributed optimization algorithms. His contributions span on various domains, including communication networks, distributed computing systems, and energy management.

Miroslav Popovic received the master's degree in computer science from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2010, and is currently pursuing the Ph.D. degree under the supervision of Prof. Jean-Yves Le Boudec at EPFL.

His main research interests are multipath data transfer protocols and various aspects of communication networks for smart grid.



Jean-Yves Le Boudec (M'89–SM'01–F'05) received the Aggregation in Mathematics from École Normale Supérieure de Saint-Cloud, Saint-Cloud, Paris, in 1980, and the doctorate degree from the University of Rennes, Rennes, France, in 1984.

He is a Full Professor with École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. From 1984 to 1987, he was with INSA/IRISA, Rennes, France. In 1987, he joined Bell Northern Research, Ottawa, ON, Canada, as a scientific staff. In 1988, he joined the IBM Zurich Research Laboratory, Zurich, Switzerland, where he was Manager of the Customer Premises Network Department. In 1994, he joined EPFL as an Associate Professor. In 1984, he developed analytical models of multiprocessor, multiple bus computers. In 1990 he invented the concept called "MAC emulation," which later became the ATM forum LAN emulation project, and developed the first ATM control point based on OSPF. He proposed in 1998 the first solution to the failure propagation that arises from common infrastructures in the Internet. He contributed to network calculus, a recent set of developments that forms a foundation to many traffic control concepts in the Internet, and coauthored a book on this topic. He is also the author of the book *Performance Evaluation* (EPFL Press, 2010). His interests are in the performance and architecture of communication systems.

Prof. Le Boudec is or has been on the program committee or Editorial Board of many conferences and journals, including ACM SIGCOMM, ACM SIGMETRICS, IEEE INFOCOM, *Performance Evaluation*, and the IEEE/ACM TRANSACTIONS ON NETWORKING. He received the IEEE Millenium Medal, the IEEE INFOCOM 2005 Best Paper Award, the CommSoc 2008 William R. Bennett Prize, and the 2009 ACM SIGMETRICS Best Paper Award.