Congestion Control of MPTCP: Performance Issues and a Possible Solution

Ramin Khalili, T-Labs/TU-Berlin, Germany

R.khalili, N. Gast, M. Popovic, J.-Y Le Boudec,
draft-khalili-mptcp-performance-issues-02
draft-khalili-mptcp-congestion-control-00

Measurement-based study supported by theory

focus on congestion control part of MPTCP [RFC 6356]

- outline:
  1. examples of performance issues
  2. can these problems be fixed in practice?

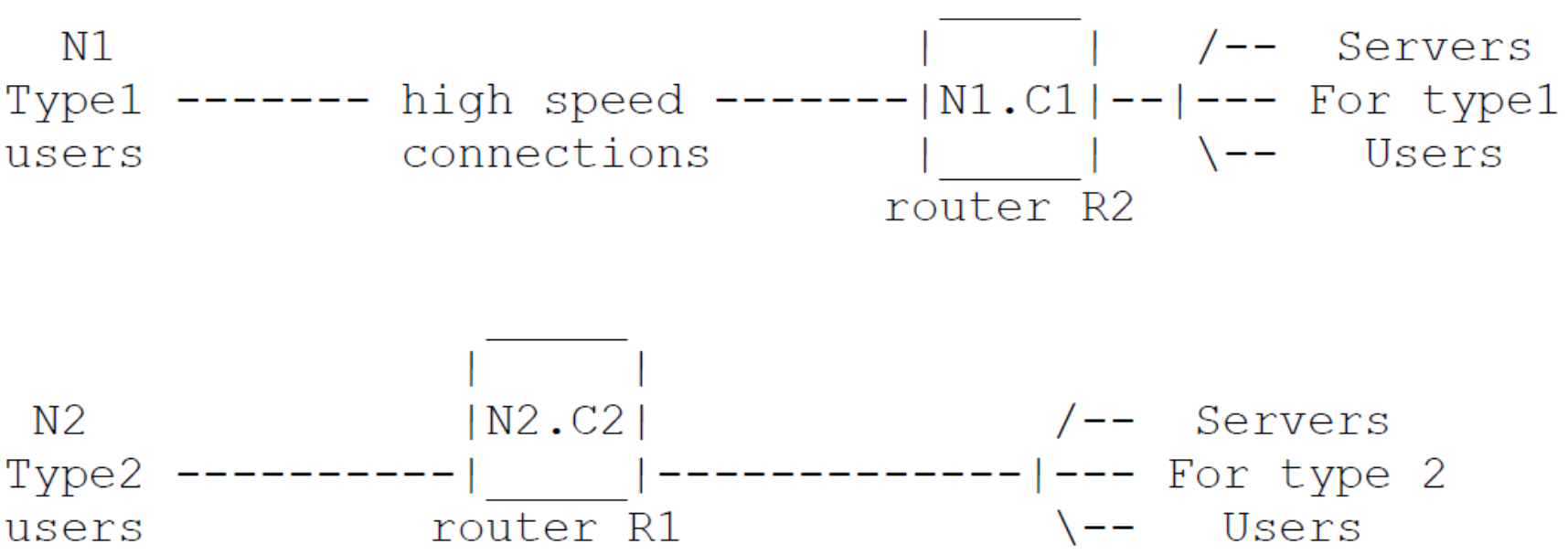
LIA [RFC 6356]: "Linked Increases" Algorithm

- adhoc design based on 3 goals
  1. improve throughput: total throughput ≥ TCP over best path
  2. do not harm: not more aggressive than a TCP over a path
  3. balance congestion while meeting the first two goals
- as also stated in RFC 6356, LIA does not fully satisfy goal 3

MPTCP CAN PENALIZE USERS

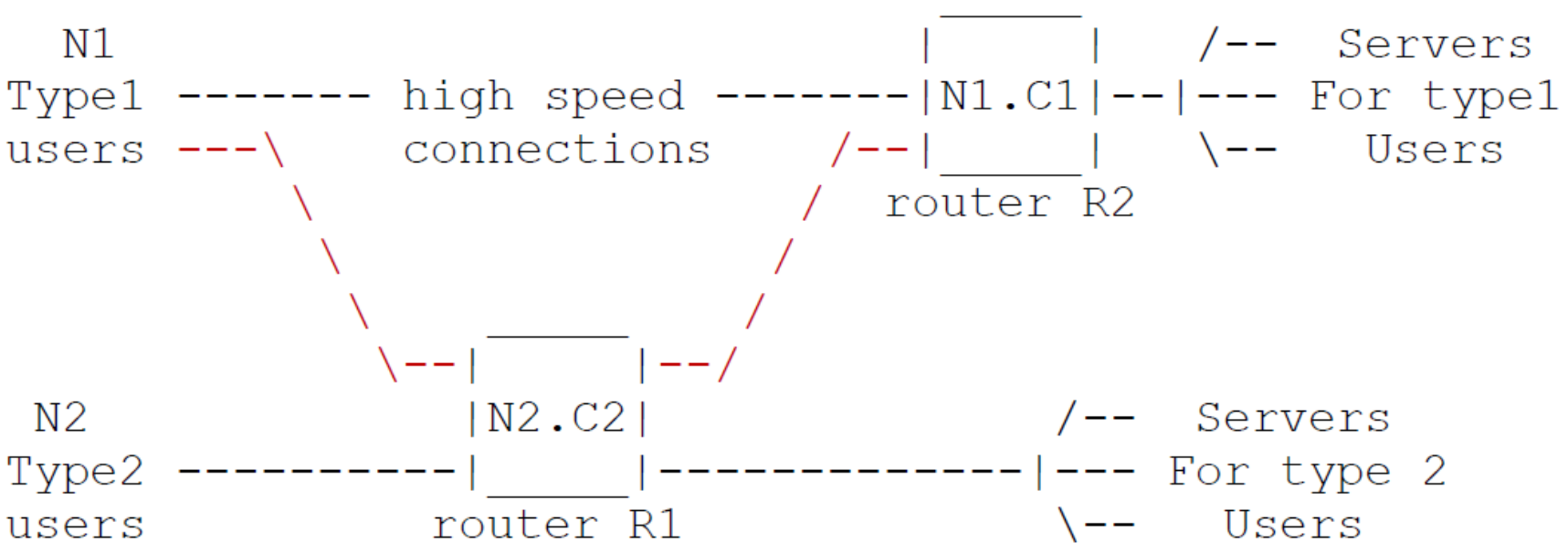
R. Khalili, N. Gast, M. Popovic, J.-Y. Le Boudec, "Performance Issues with MPTCP", draft-khalili-mptcp-performance-issues-02

Scenario A: MPTCP can penalize TCP users



- bottleneck for type 1 user is at the server side
- bottleneck for type 2 users is at the access side

Scenario A: MPTCP can penalize TCP users

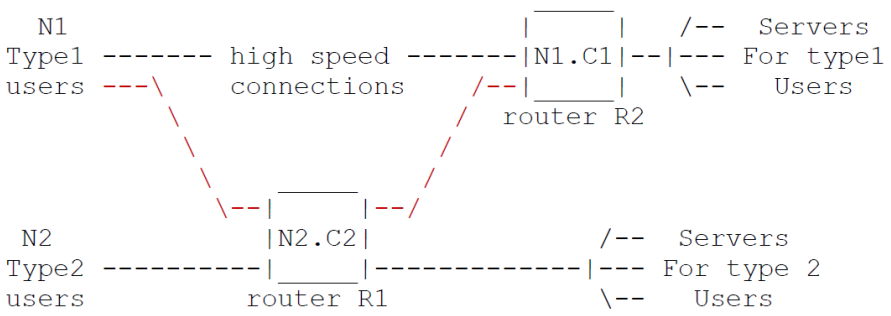


- type 1 users upgrade to MPTCP users
- MPTCP transmits significant traffic over R1: no benefits for type 1 users but hurts R2 users

Throughput of type 2 users reduced without any benefit for type 1 users

$C_1=C_2=1$ Mbps

		Type1 users are single path (measurement)	Type1 users are multipath: MPTCP (meas.)		
N1=10	type1	0.98	0.96		
N2=10	type2	0.98	0.70		
N1=30	type1	0.98	0.98		
N2=10	type2	0.98	0.44		



We compare MPTCP with two theoretical baselines

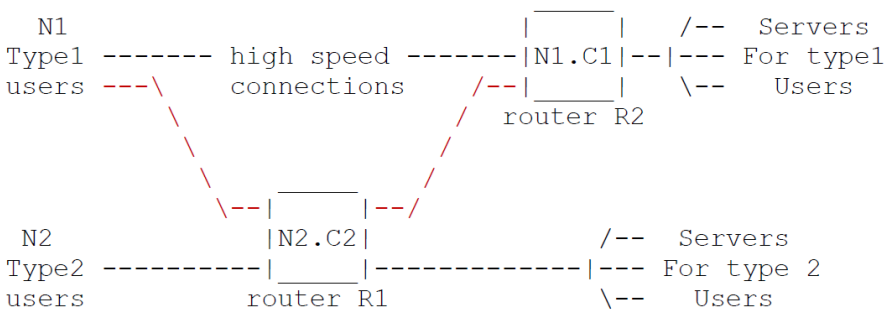
1. optimal algorithm (without probing cost): theoretical optimal load balancing [Kelly,Voice 05]

2. optimal algorithm with probing cost: theoretical optimal load balancing including minimal probing traffic
   - using a windows-based algorithm, a min probing traffic of 1 MSS/RTT is sent over each path

Part of problem is in nature of things, but MPTCP seems to be far from optimal

$C_1=C_2=1$ Mbps

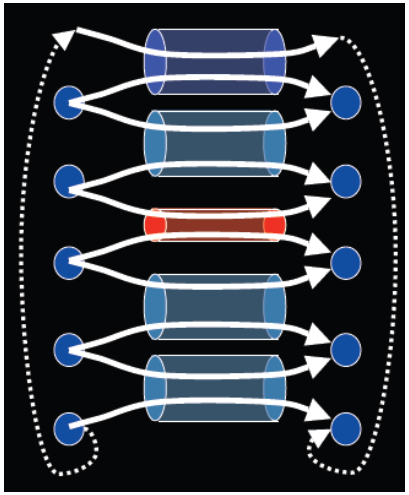
		Type1 users are single path	Type1 users are multipath			
			MPTCP	optimal algorithm		
				with p. cost	w/out p. cost	
		(measurement)	(meas.)	(theory)	(theory)	
N1=10	type1	0.98	0.96	1	1	
N2=10	type2	0.98	0.70	0.94	1	
N1=30	type1	0.98	0.98	1	1	
N2=10	type2	0.98	0.44	0.8	1	



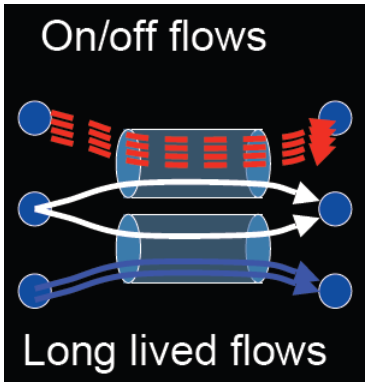
CAN THE SUBOPTIMALITY OF MPTCP WITH LIA BE FIXED IN PRACTICE?

R. Khalili, N. Gast, M. Popovic, J.-Y. Le Boudec, "Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP", draft-khalili-mptcp-congestion-control-00

LIA's design forces tradeoff between responsiveness and congestion balancing



provide congestion balancing



be responsive

optimal congestion balacing
but not responsive

LIA's implementation
(RFC 6356)

responsive but
bad congestion balancing

ε=0 ε=1 ε=2

ε is a design parameter

OLIA: an algorithm inspired by utility maximization framework

- simultaneously provides responsiveness and congestion balancing
- an adjustment of optimal algorithm [Kelly,Voice 05]
  - by adapting windows increases as a function of quality of paths, we make it responsive and non-flappy
- implemented on the MPTCP Linux kernel

Set of collected paths (collected_paths)

- l_r: smoothed estimation of number of bytes transmitted between last two losses
- best_paths: set of paths with max $(l_r * l_r)/rtt_r$
  - paths that are presumubly the bests for the MPTCP connection (based on TCP loss-throughput formula)
- max_w_paths: set of path with max windows
- collected_paths: set of paths in best_paths but not in max_w_paths

OLIA: "Opportunistic Linked-Increases Algorithm"

For each path r:

- **increase part:** for each ACK on r, increase w_r by

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

optimal congestion balancing: adaptation of [kelly, voice 05]

responsiveness; reacts to changes in current windows

- **decrease part:** each loss on r, decreases w_r by $w_r/2$

OLIA reforwards traffic from fully used paths to paths that have free capacity

$\alpha_r(t)$ is calculated as follows:

- if r is in collected_paths, then

$$\alpha_r(t) = \frac{1/number\_of\_paths}{|collected\_paths|}$$

- if r is in max_w_paths and if collected_paths is not empty

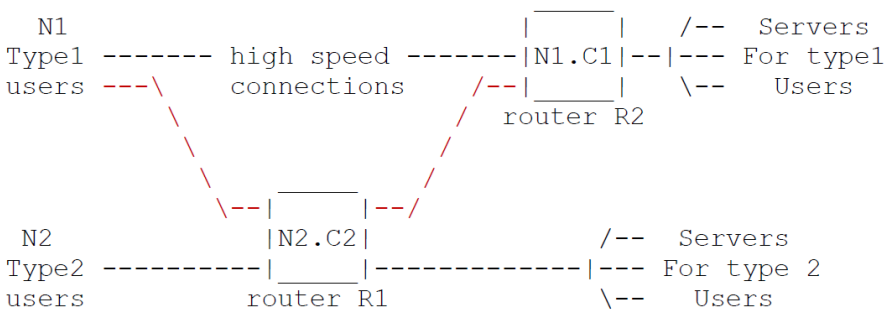
$$\alpha_r(t) = -\frac{1/number\_of\_paths}{|max\_w\_paths|}$$

- otherwise, $\alpha_r = 0$.

Scenario A: OLIA performs close to optimal algorithm with probing cost

$C_1=C_2=1$ Mbps

		Type1 users are single path (measurement)	Type1 users are multipath: MPTCP w. OLIA [with LIA] (measurement)	Type1 users are multipath: optimal algorithm with p. cost (theory)	Type1 users are multipath: optimal algorithm w/out p. cost (theory)
N1=10	type1	0.98	0.98 [0.96]	1	1
N2=10	type2	0.98	0.86 [0.70]	0.94	1
N1=30	type1	0.98	0.98 [0.98]	1	1
N2=10	type2	0.98	0.75 [0.44]	0.8	1



Summary

- MPTCP with LIA suffers from important performance problems
- these problems can be mitigated in practice
- OLIA outperforms LIA in all scenarios we studied [CoNEXT 12]
- suggestion: congestion control part of MPTCP should be revisited by the IETF committees

References

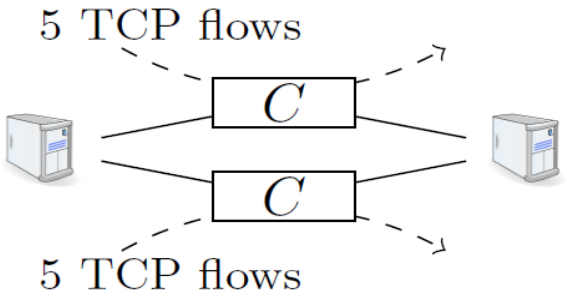
- [RFC 6356]: C. Raiciu, M. Handly, and D. Wischik. “Coupled congestion control for multipath transport protocols”. 2011
- [Kelly, Voice 05]: F. Kelly and T. Voice. “Stability of end-to-end algorithms for joint routing and rate control”. ACM SIGCOMM CCR, 35, 2005.
- [CoNEXT 12]: R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec. “Non pareto-optimality of mptcp: Performance issues and a possible solution”. ACM CoNEXT 2012 (best paper).

BACK UP SLIDES

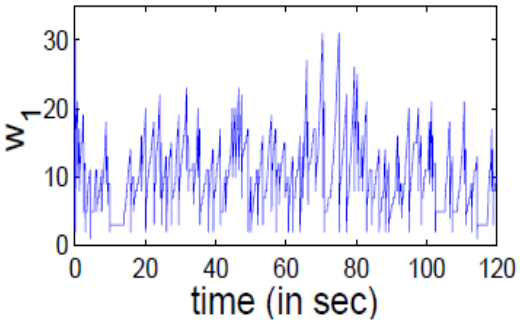
Theoretical results: OLIA solves problems with LIA

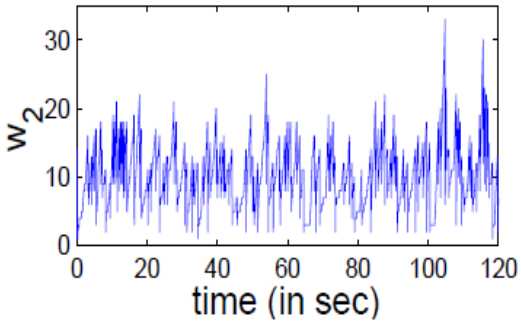
- using a fluid model of OLIA
- Theorem: OLIA satisfies design goals of LIA (RFC 6356)
- Theorem: OLIA is Pareto optimal
- Theorem: when all paths of a user have similar RTTs, OLIA provides optimal load balancing

An illustrative example of OLIA's behavior symmetric scenario

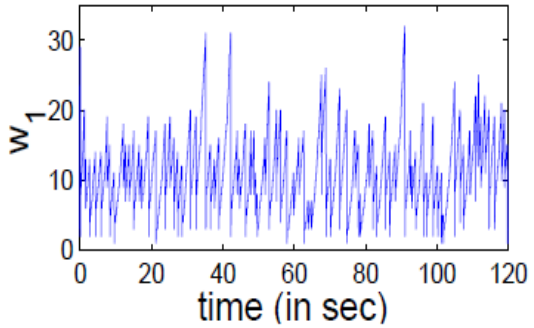


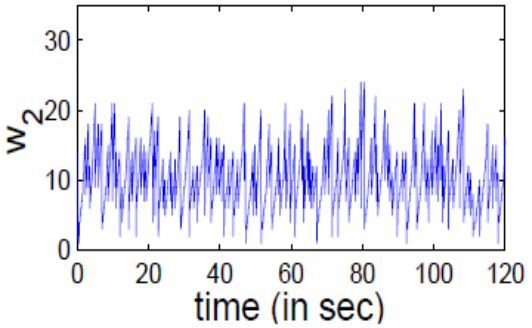
both paths are equally good





MPTCP with OLIA

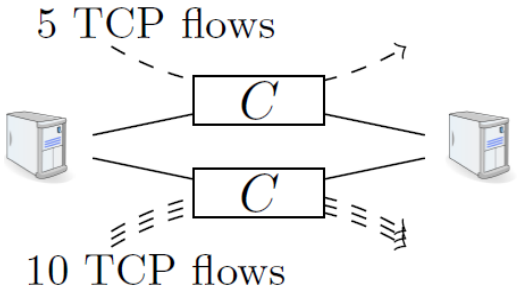




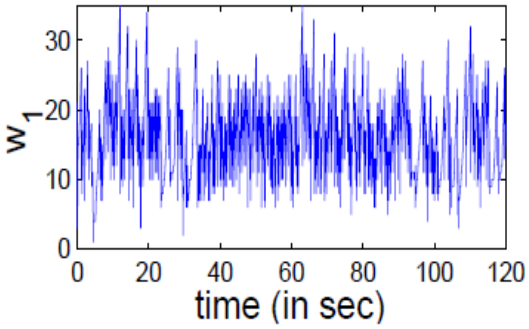
MPTCP with LIA

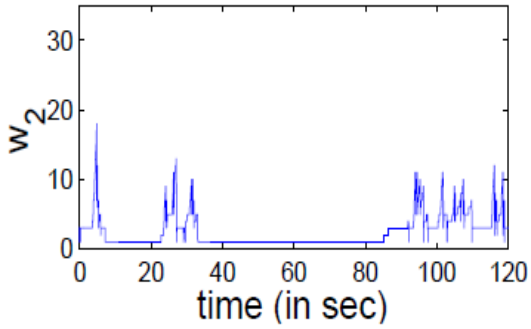
OLIA uses both paths; it is non-flappy and responsive

An illustrative example of OLIA's behavior asymmetric scenario

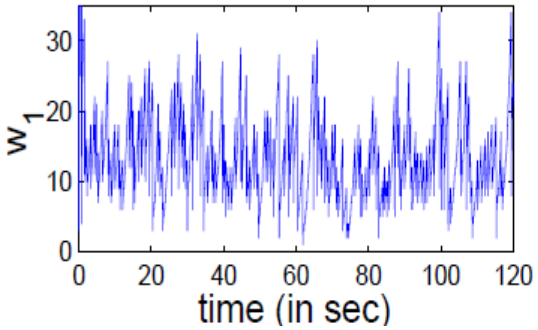


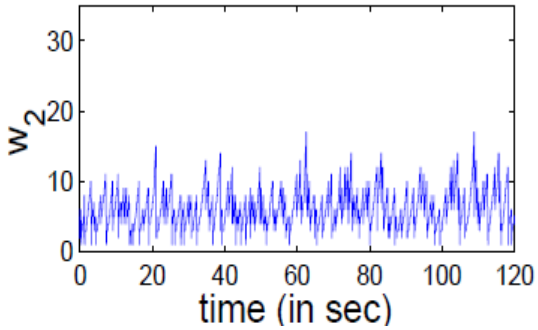
second path is congested





MPTCP with OLIA

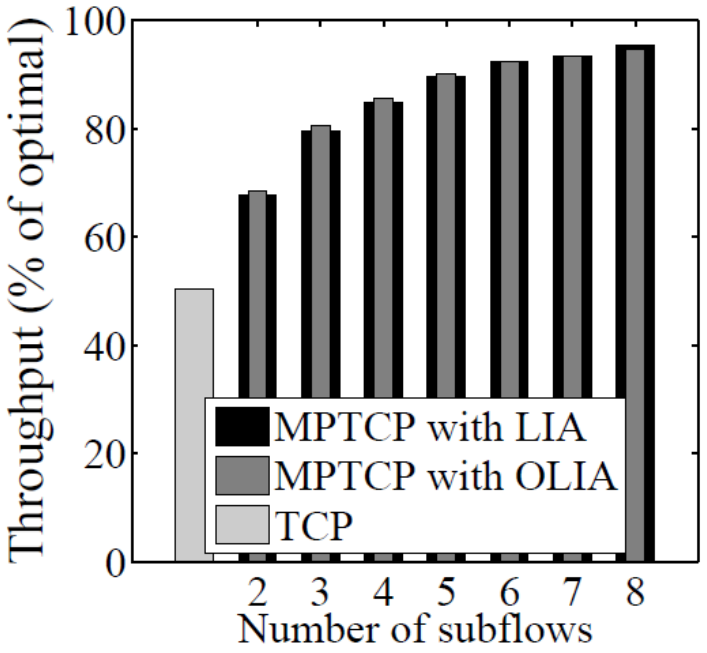




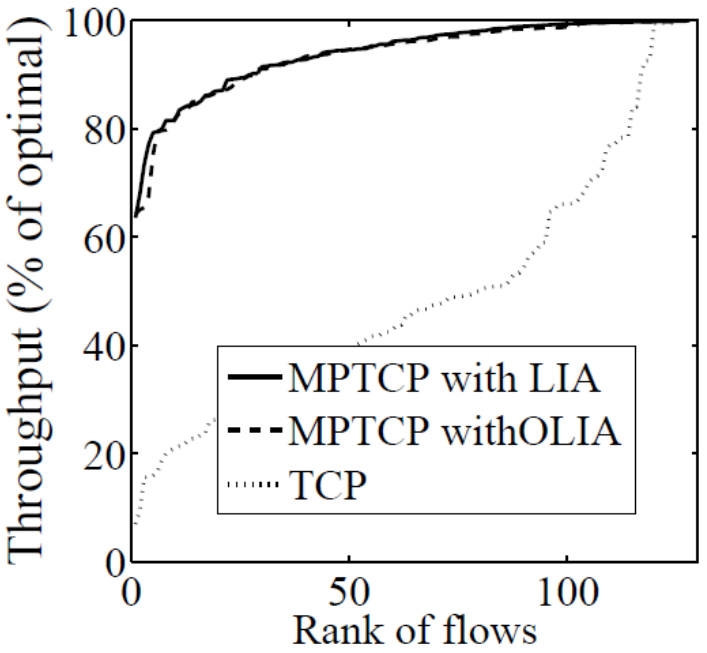
MPTCP with LIA

OLIA uses only the first one; it balances the congestion

Static fat-tree topology: OLIA explores path diversity and show no flappiness



(a) Aggregated throughput.

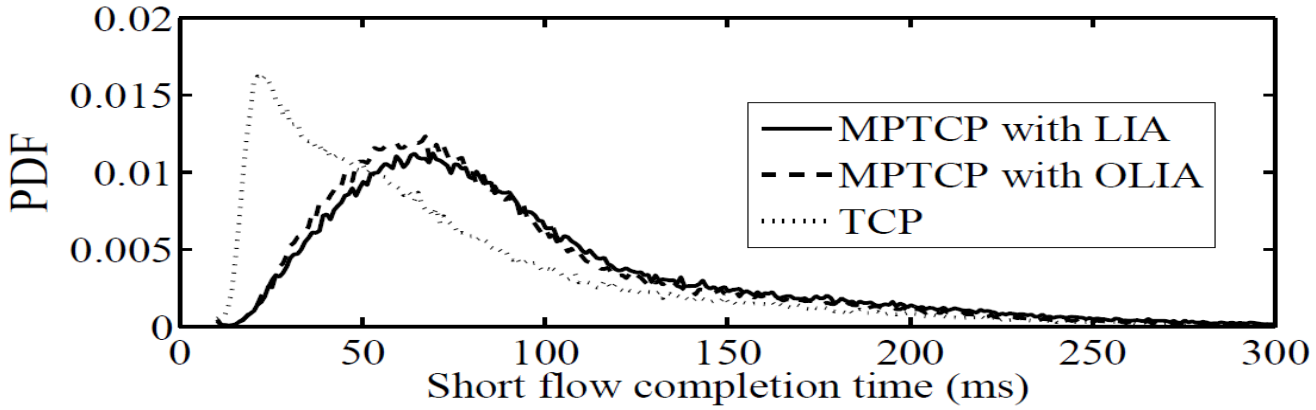


(b) Throughput of users.

a data center with fat-tree topology (similarly to what studied at [MPTCP-Sigcomm 2011])

Highly dynamic setting with short flows

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%



4:1 oversubscribed fat-tree; 1/3 of flows are long flows and 2/3 are short flows (similarly to [MPTCP-Sigcomm 2011])