Balia (<u>Ba</u>lanced <u>li</u>nked <u>a</u>daptation)

A new MPTCP congestion control algorithm

Anwar Walid
Jaehyun Hwang

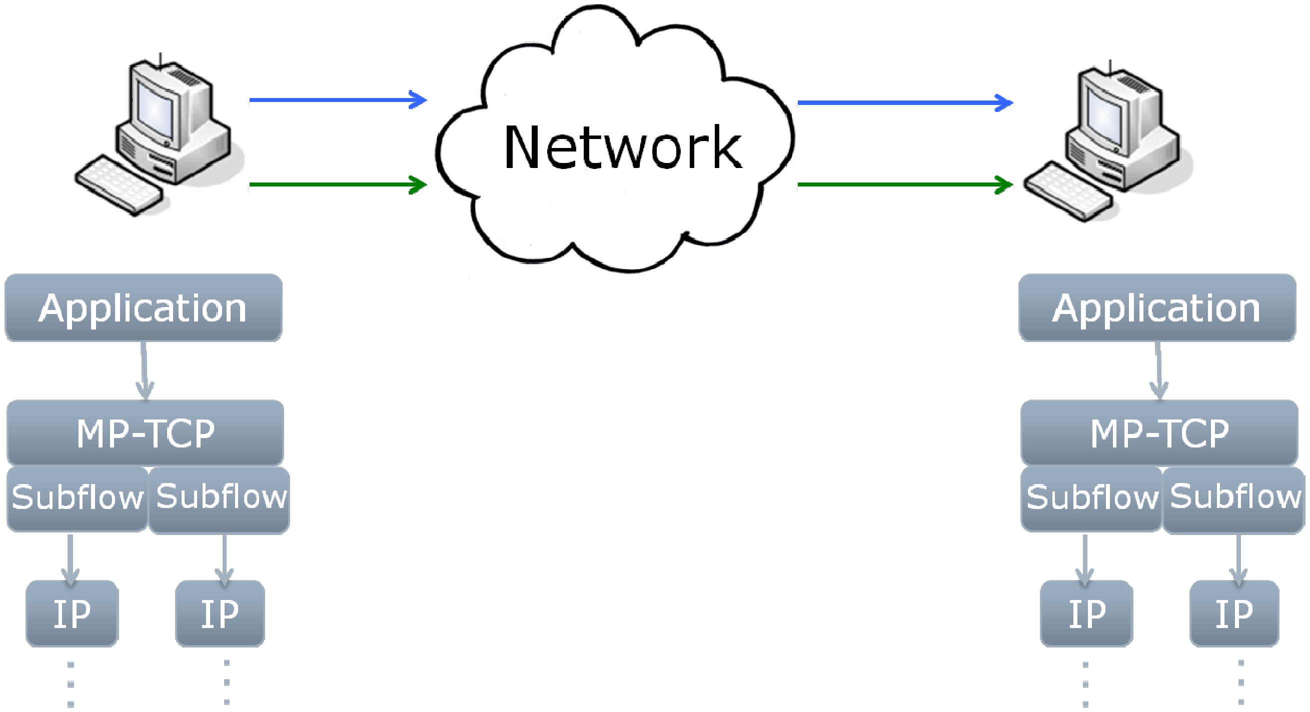
Bell Labs

Qiuyu Peng
Steven Low

Caltech

July 2014

MPTCP Congestion Control



(IETF RFC 6824)

How to control congestion over multiple paths?

Desirable Control Properties: Increase throughput and robustness to link failure while remaining

- TCP friendly
- Responsive

Key message

Tradeoff between <u>friendliness</u> (to single path TCP) & <u>responsiveness</u> (to network changes)

- **is inevitable, but**
- **can be systematically balanced**

New Algorithm Balia explicitly balances this tradeoff

- **based on a new design framework**

Prior Proposals

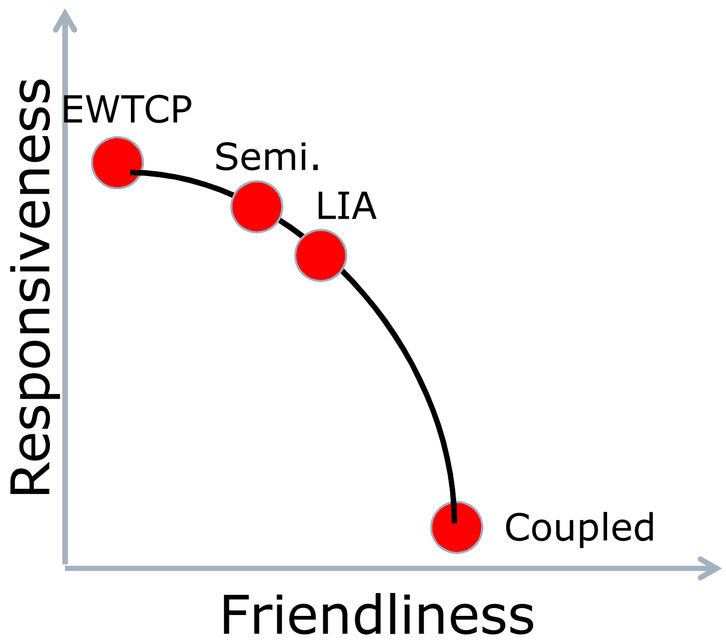
- **EWTCP**
  - [Iyengar’ 06] [Honda’ 09]
- **Coupled MPTCP**
  - [Kelly’ 05] [Han 06]
- **Semi-coupled MPTCP**
  - [Wischik’ 11]
- **LIA (RFC6356) MPTCP**
  - [Wischik’ 11]
- **OLIA MPTCP**
  - [Khalili’ 12]

Desirable properties

Increase throughput and robustness to link failure while remaining

- TCP friendly
- Responsive

Unfortunately …
there is provably an inevitable tradeoff

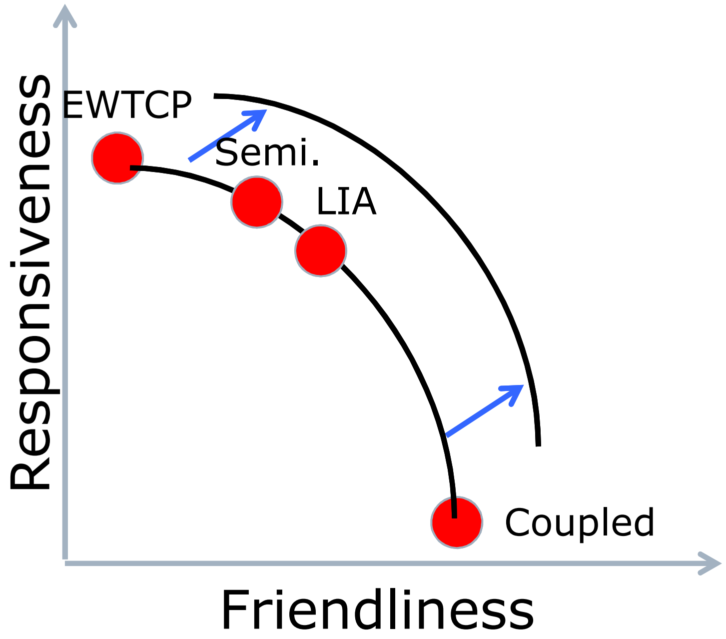


Two questions

1. Have prior algorithms achieved the best tradeoff possible?

   - Bad news: No !

… but significant improvement is possible

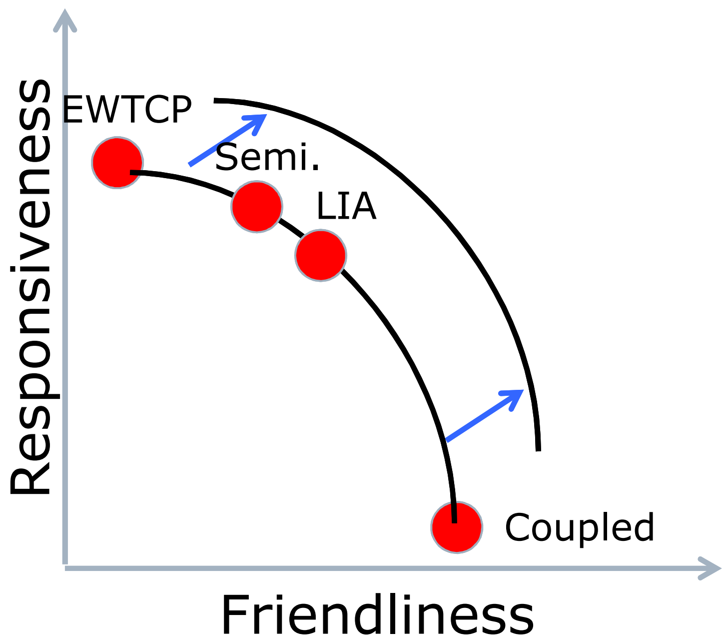


Two questions

2. Can we systematically design this inevitable tradeoff ?

   - Good news: Yes !

… a new framework to better understand & design



First question first …

… **let's first understand some problems with LIA and OLIA**
… and then look at a solution

Problem with LIA (RFC6356)

LIA can be

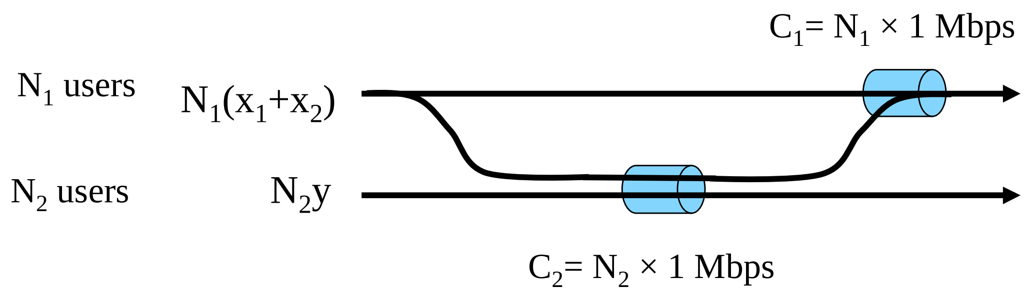
- unfriendly to single path TCP (SPTCP)

LIA can be unfriendly to SPTCP

- … even when its own throughput is max'ed out !

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

		MPTCP with LIA (measurement)	optimal with probing cost (theory)	optimal w/out probing cost (theory)
N_1=10	x_1+x_2	0.96	1	1
N_2=10	y	0.7	0.94	1
N_1=30	x_1+x_2	0.96	1	1
N_2=10	y	0.4	0.8	1



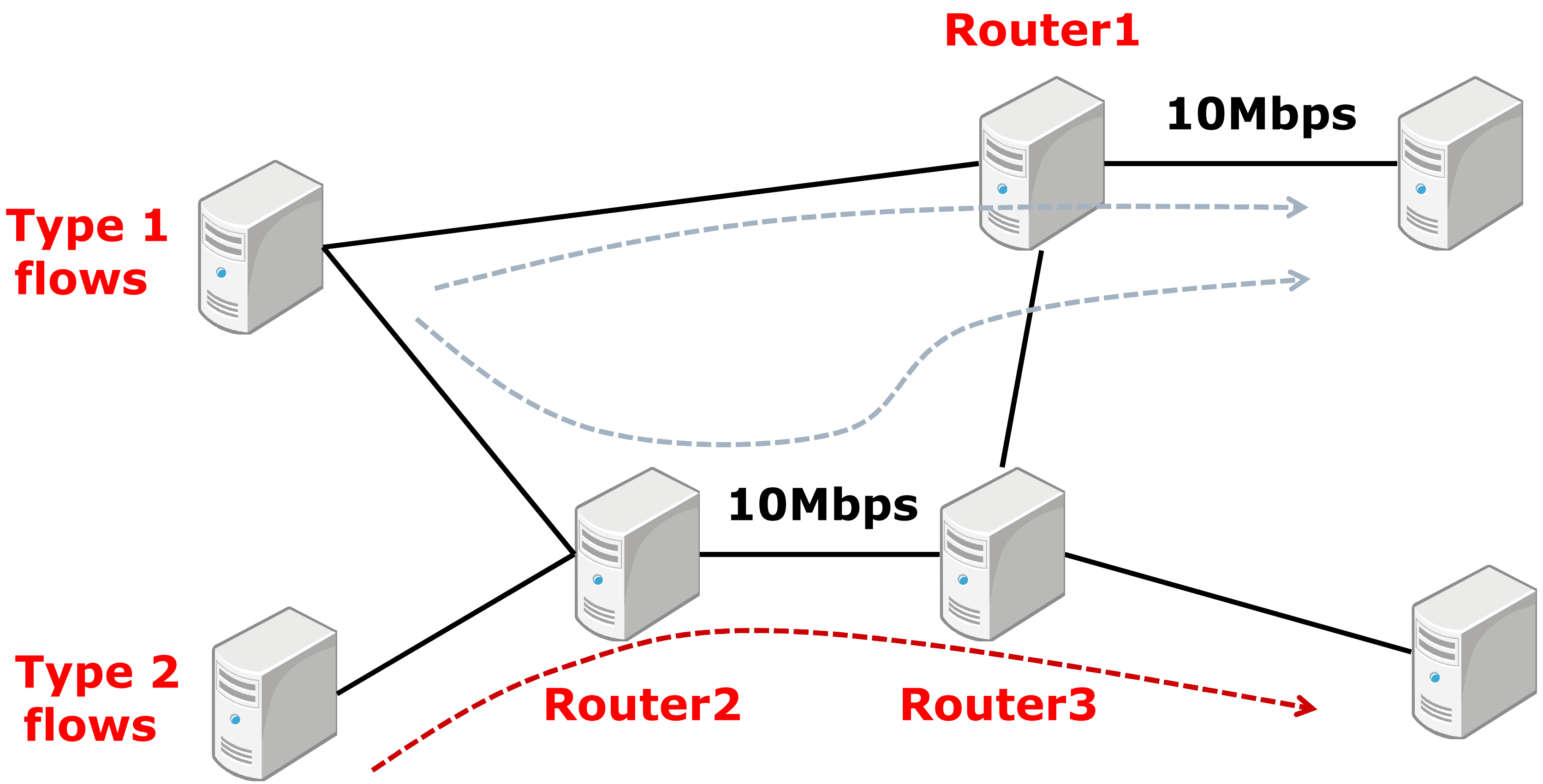
10"

SPTCP is worse off than optimal

- by 26%
- by 50%

[Source: Khalili iccrg presentation on OLIA]

We have confirmed Khalili's discovery with our own testbed



Type 2 flows are SPTCP.

When all flows are SPTCP, they achieve capacity on each path

		aggregate throughput: type 1 flows are SPTCP
N1=5 N2=5	type 1	9.47
	type 2	9.29
N1=15 N2=5	type 1	9.39
	type 2	9.29

C1=C2=10Mbps

When type 1 users are MPTCP, LIA starves SPTCP

- … even when LIA throughput is max'ed out !

		type 1 flows are SPTCP	Type 1 flows are MPTCP
			LIA
N1=5	type 1	9.47	9.26
N2=5	type 2	9.29	7.55
N1=15	type 1	9.39	8.96
N2=5	type 2	9.29	6.94

C1=C2=10Mbps

SPTCP's are worse off

- by 19%
- by 25%

Two better designs

- **OLIA**
- **Balia**

		type 1 flows are SPTCP	Type 1 flows are MPTCP		
			LIA	OLIA	Balia
N1=5	type 1	9.47	9.26	9.25	9.25
N2=5	type 2	9.29	7.55	8.13	8.32
N1=15	type 1	9.39	8.96	8.93	9.02
N2=5	type 2	9.29	6.94	7.41	7.98

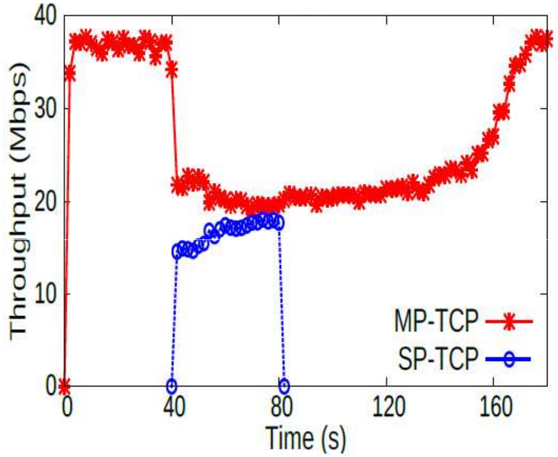
C1=C2=10Mbps

Let’s now examine OLIA in more detail

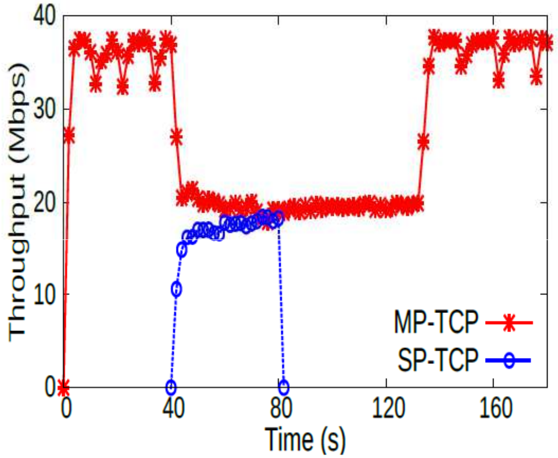
- Is OLIA responsive to network changes?

OLIA can be unresponsive to network changes

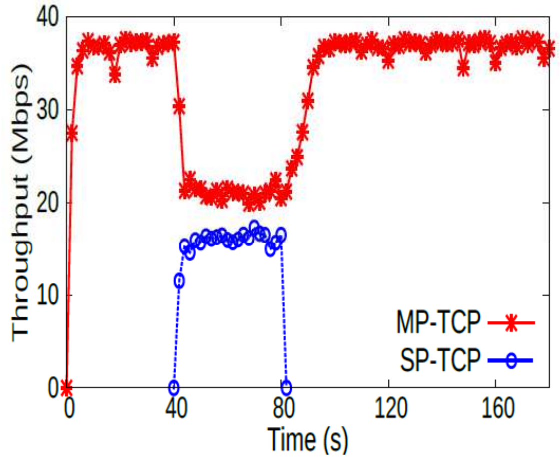
Coupled



OLIA

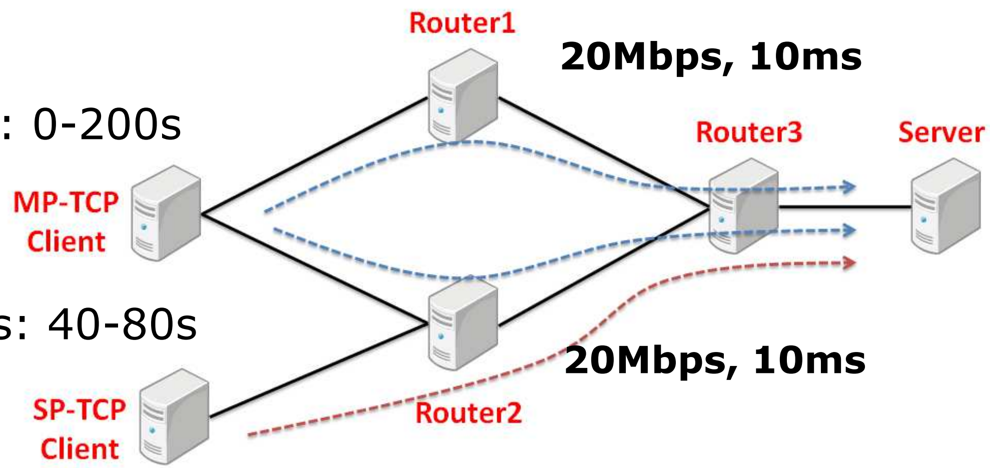


LIA

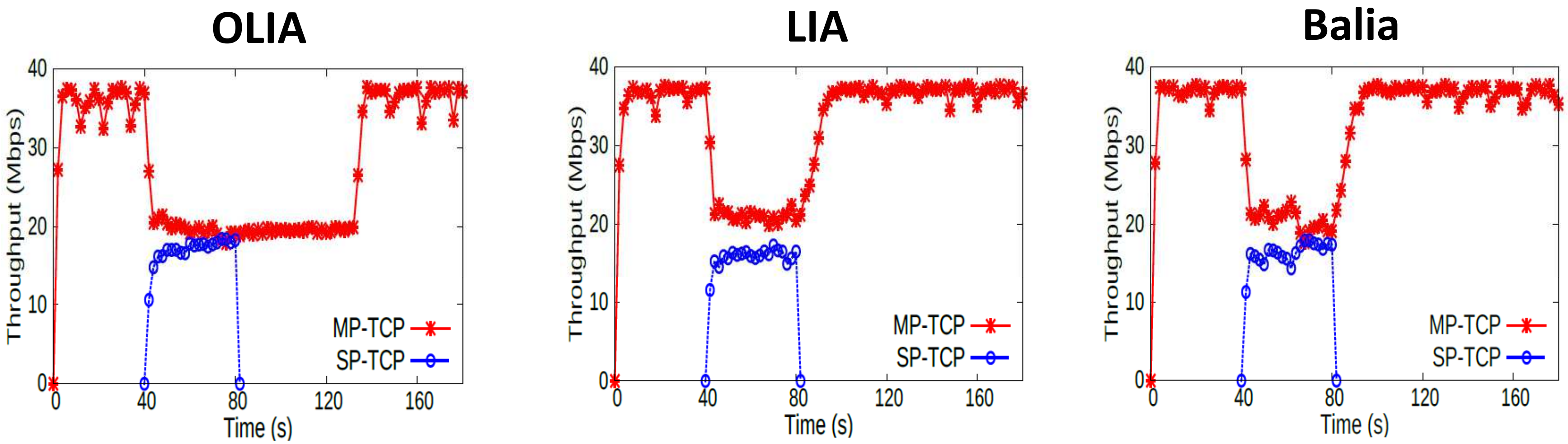


1 MP-TCP flow: 0-200s

5 SP-TCP flows: 40-80s

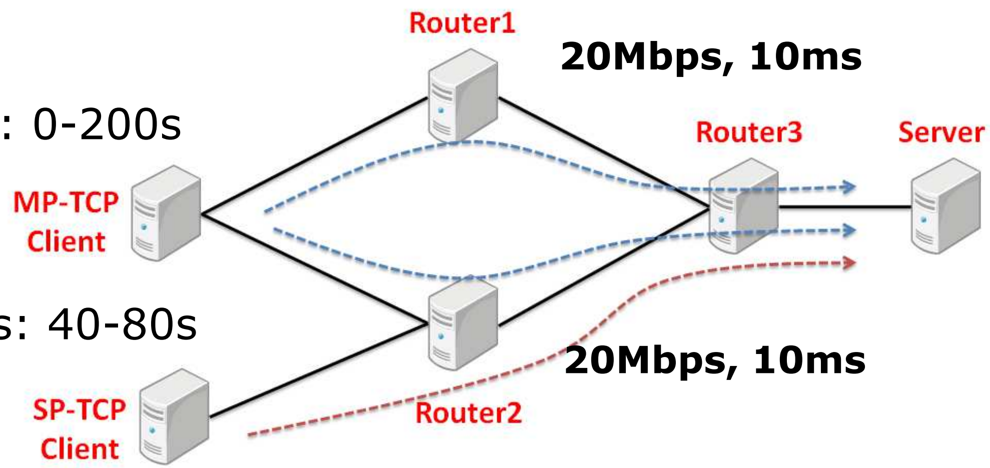


Balia is responsive to network changes



1 MP-TCP flow: 0-200s

5 SP-TCP flows: 40-80s



So LIA can be unfriendly, while OLIA can be unresponsive

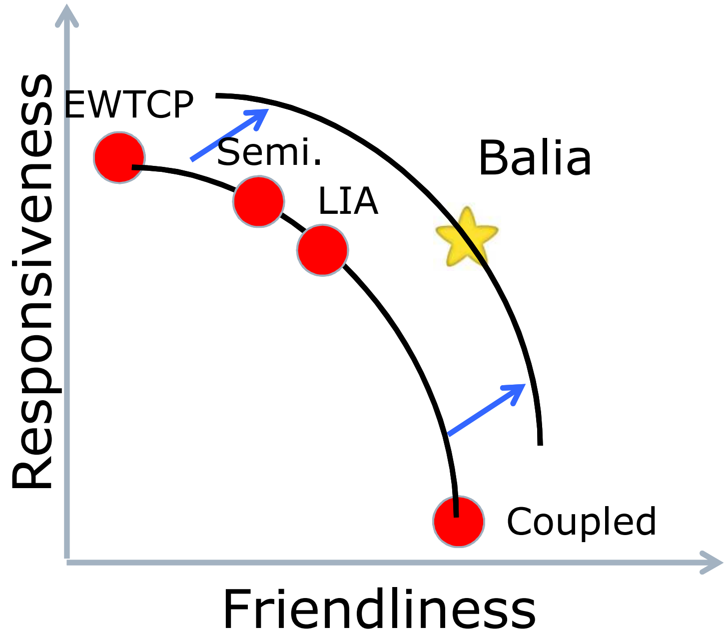
Second question now …

… what is the nature of this inevitable tradeoff
… how does Balia design this tradeoff

Balia

Generalized MPTCP algorithm that strikes a good balance

- Friendly
- Responsive



Balia: Balanced linked adaptation

Balia

Generalized MPTCP algorithm that strikes a good balance

- Friendly
- Responsive

… designed based on a new theoretical framework
… that allows better understanding of this tradeoff

Balia

- For each ACK on path *r, increment window by*:

$$\frac{x_r}{\tau_r\left(\sum x_k\right)^2}\cdot\left(\frac{1+\alpha_r}{2}\right)\cdot\left(\frac{4+\alpha_r}{5}\right)$$

- For each Loss on path *r, decrement window by*:

$$\frac{w_r}{2}\cdot\min\{\alpha_r,\ 1.5\}$$

$x_r := w_r/\tau_r$ $\quad$ $\alpha_r := \frac{\max\{x_k\}}{x_r}$ $\quad$ τ_r is the round trip time

On a single path, α_r=1 and Balia reduces to Reno

Key message

Tradeoff between friendliness & responsiveness

- is inevitable, but
- can be systematically balanced

Balia explicitly balances this tradeoff

- based on a new design framework

Current status

- Linux implementation
  - Working on approval to make our code part of the UCLouvain's MPTCP implementation
- Documents
  - Paper: "Multipath TCP: Analysis, Design and Implementation" (http://arxiv.org/abs/1308.3119v2)
  - To be submitted: draft-walid-mptcp-congestion-control-00
- Experiment plans
  - NorNet: Multi-homed research testbed
  - Small-scale data center testbed
  - Mobile testbed with WiFi/3G/LTE

Back up slides

Theoretical Framework for Design

- "Multipath TCP: Analysis, Design and Implementation," Peng, Walid, Hwang and Low (http://arxiv.org/abs/1308.3119v2). Earlier version appeared in ACM Sigmetrics, 2013.

Unified MPTCP model

- Dynamics of throughput x_r on path *r*:

Algorithm

Congestion price

$$\dot{x}_r = k_r(\mathbf{x}_s)(\phi_r(\mathbf{x}_s) - q_r)$$

Different designs: different κ_r and ϕ_r

- TCP Reno (Jacobson 1988)
- EWTCP (Honda et al 2009)
- Coupled MPTCP (Kelly & Voice 2005, Han et al 2004)
- Semicoupled MPTCP (Wischik et al 20011)
- LIA MPTCP (Wischik et al 2011)
- Balia (Peng et al 2013)

Provable properties

Theorem

Balia has a unique equilibrium point

Theorem

The unique equilibrium point is asymptotically stable

Theorem

(Almost) all MPTCP algorithms face an inevitable tradeoff between

- TCP friendliness
- Responsiveness

Design of the tradeoff

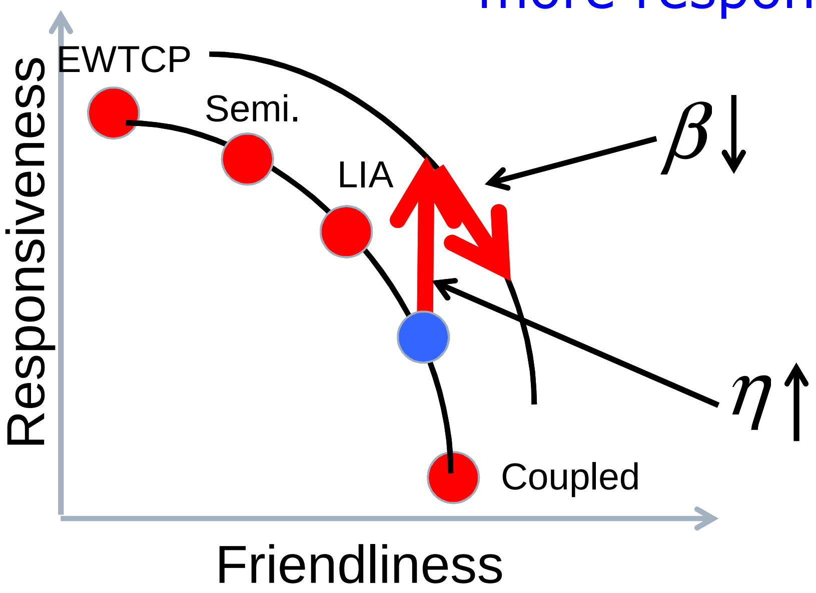
1. **Explicitly parameterize friendliness and responsiveness**

$$k_r(\mathbf{x}_s) = x_r \cdot \left(x_r + \boxed{\eta}\,(x_s^{\max} - x_r)\right)$$

$$\phi_r(\mathbf{x}_s) = f_r(\mathbf{x}_s) \cdot \left(x_r + \boxed{\beta}\,(x_s^{\max} - x_r)\right)$$

more friendly: small β

more responsive: large β, η



$$f_r(\mathbf{x}_s) := \frac{1}{\tau_r^2 x_r \|\mathbf{x}_s\|_1^2}$$

Design of the tradeoff

2. Choose parameters judiciously

$$k_r(\mathbf{x}_s) = x_r \cdot \left(x_r + \boxed{\eta}\,(x_s^{\max} - x_r)\right)$$

$$\phi_r(\mathbf{x}_s) = f_r(\mathbf{x}_s) \cdot \left(x_r + \boxed{\beta}\,(x_s^{\max} - x_r)\right)$$

balanced tradeoff

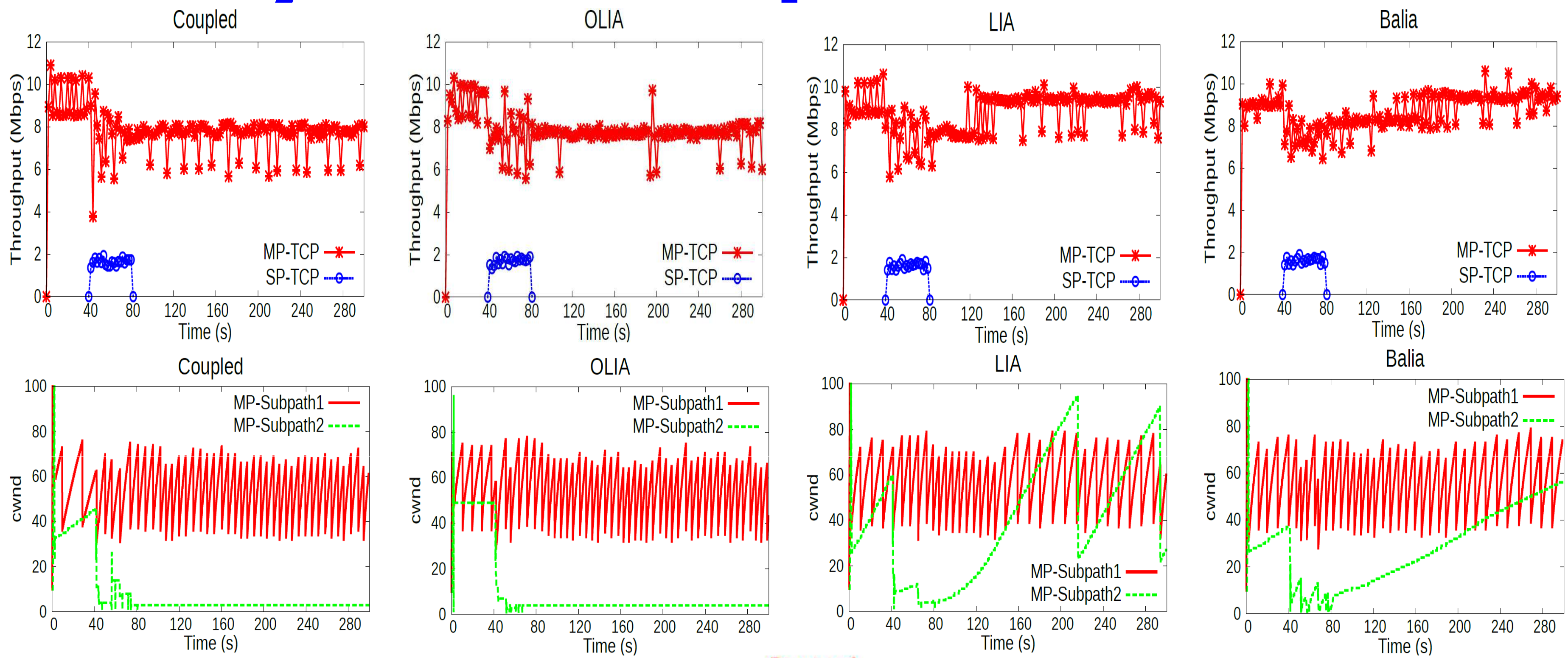
LIA: $(\eta, \beta) = (0, 1)$

Balia: $(\eta, \beta) = (0.5, 0.2)$

more responsive

more friendly

Asymmetric paths



1 MP-TCP flow: 0-300s

5 SP-TCP flows: 40-80s