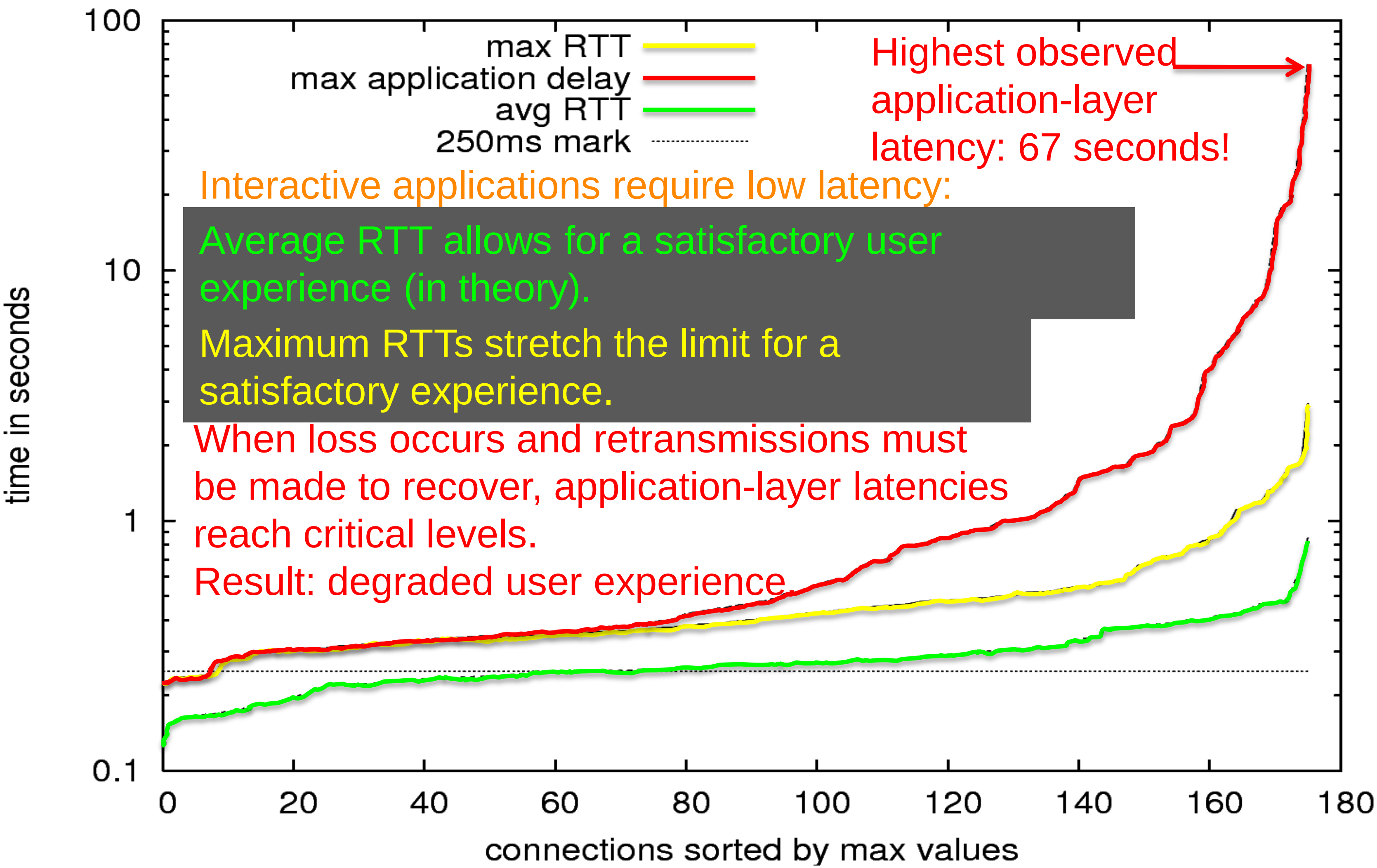
TCP modifications to reduce thin-stream latency

Andreas Petlund

Simula Research Laboratory AS

30.07.2010

Anarchy Online server-side trace analysis



Griwodz et al.: “The fun of using TCP for an MMORPG” (2006)

Interactive thin streams over TCP

application (platform)	payload size (Bytes)			packet interarrival time (ms)						avg. bandwidth requirement	
								percentiles			
	average	min	max	average	median	min	max	1%	99%	(pps)	(bps)
World of Warcraft	26	6	1228	314	133	0	14855	0	3785	3.185	2046
Anarchy Online	98	8	1333	632	449	7	17032	83	4195	1.582	2168
Age of Conan	80	5	1460	86	57	0	1375	24	386	11.628	12375
BZFlag†	30	4	1448	24	0	0	540	0	151	41.667	31370
Casa (sensor network)	175	93	572	7287	307	305	29898	305	29898	0.137	269
Windows remote desktop	111	8	1417	318	159	1	12254	2	3892	3.145	4497
Skype (2 users)†	236	14	1267	34	40	0	1671	4	80	29.412	69296
SSH text session	48	16	752	323	159	0	76610	32	3616	3.096	2825

† Application using TCP fallback due to UDP being blocked by a firewall.

Time-dependent applications

High retransmission latencies

Analysis of TCP for thin streams

Linux TCP flavours (2.6.16) analysed:

– New Reno -SACK -DSACK -FACK
– DSACK+FACK -Westwood -BIC -Vegas

Poor overall performance for interactive thin streams with all tested flavours.

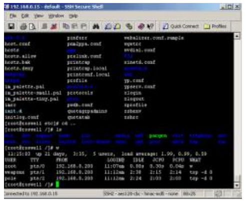
New Reno best "on average" for thin-stream latency.

Griwodz et al.: "The fun of using TCP for an MMORPG" (2006)

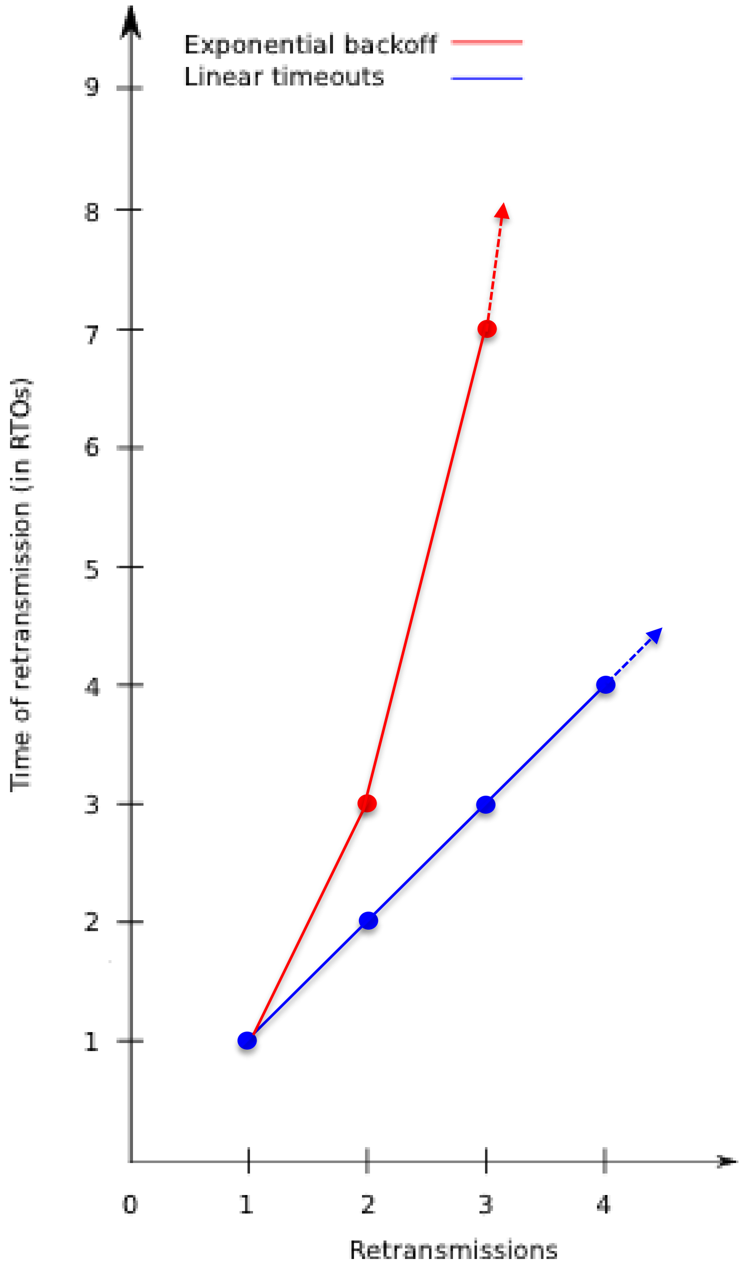
Thin streams need help with latency!

- **Greedy streams (throughput) the driving force in TCP development.**
- **Mechanisms have been suggested that (partially) address the issue (e.g. Early Retransmit - RFC5827)**
- **Thin streams need more help to deal with latency issues.**

Interactive, thin-stream applications that benefit from the thin-stream mechanisms include stock exhange applications, remote control of PCs (like RDP, VNC and SSH), voice over IP and networked games.



Timeouts and exp. backoff



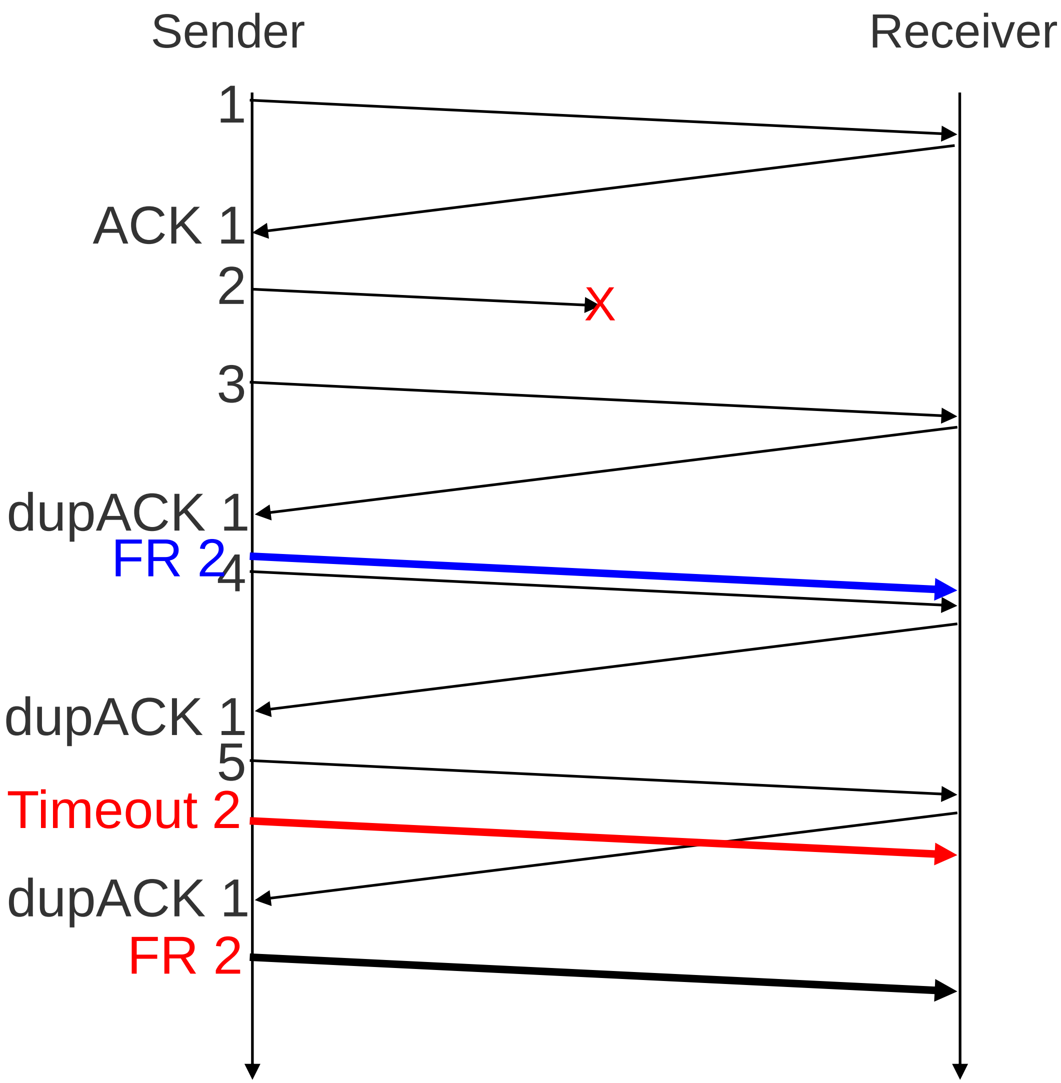
Retransmission time-out (RTO) will double for each consecutive loss.

Use linear timeouts (LT) for thin streams

TCP and SCTP standard RTO_{min}: 1000ms

TCP in Linux uses a 200ms RTO_{min}

Fast retransmissions



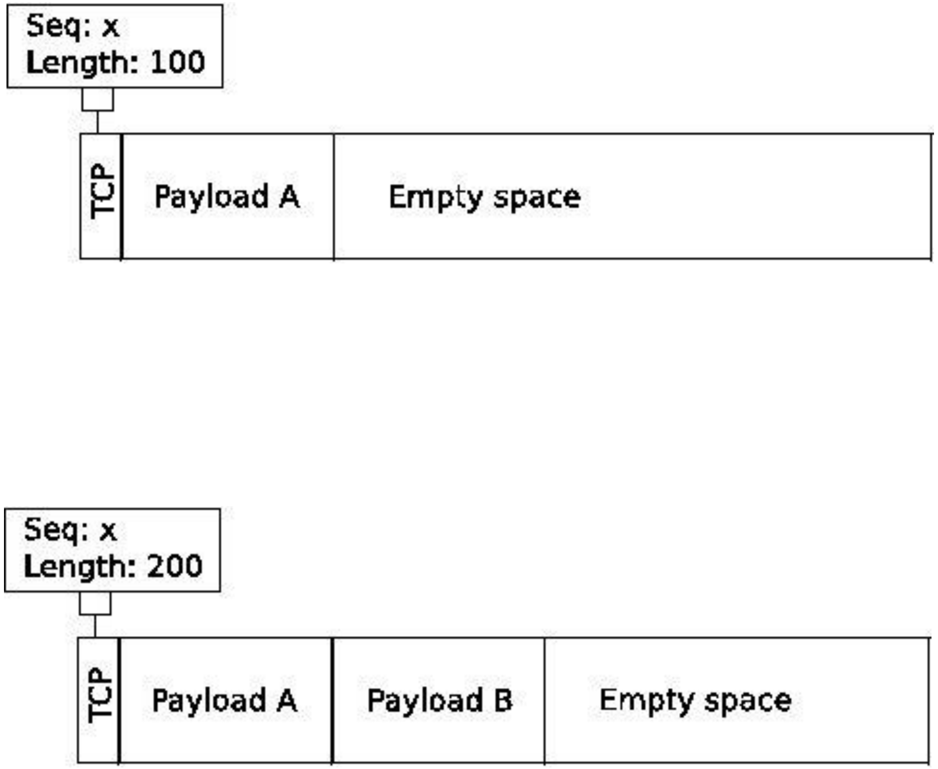
- Thin streams often have < 1 packet per RTT.
- Timeout happens before a fast retransmission can be triggered.
- For thin streams: fast retransmit on first received dupACK (mFR)
- Following scheme from Early Retransmit (but consequently retransmit on first dupACK)

Redundant data bundling

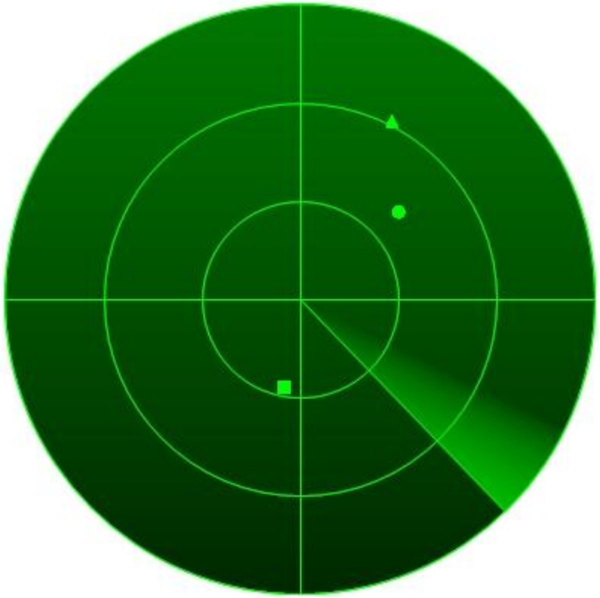
– Preempting the experience of loss.

– Will not increase number of sent packets.

– Introduces inherent redundancy.



Thin-stream detection



Retransmission mechanisms :

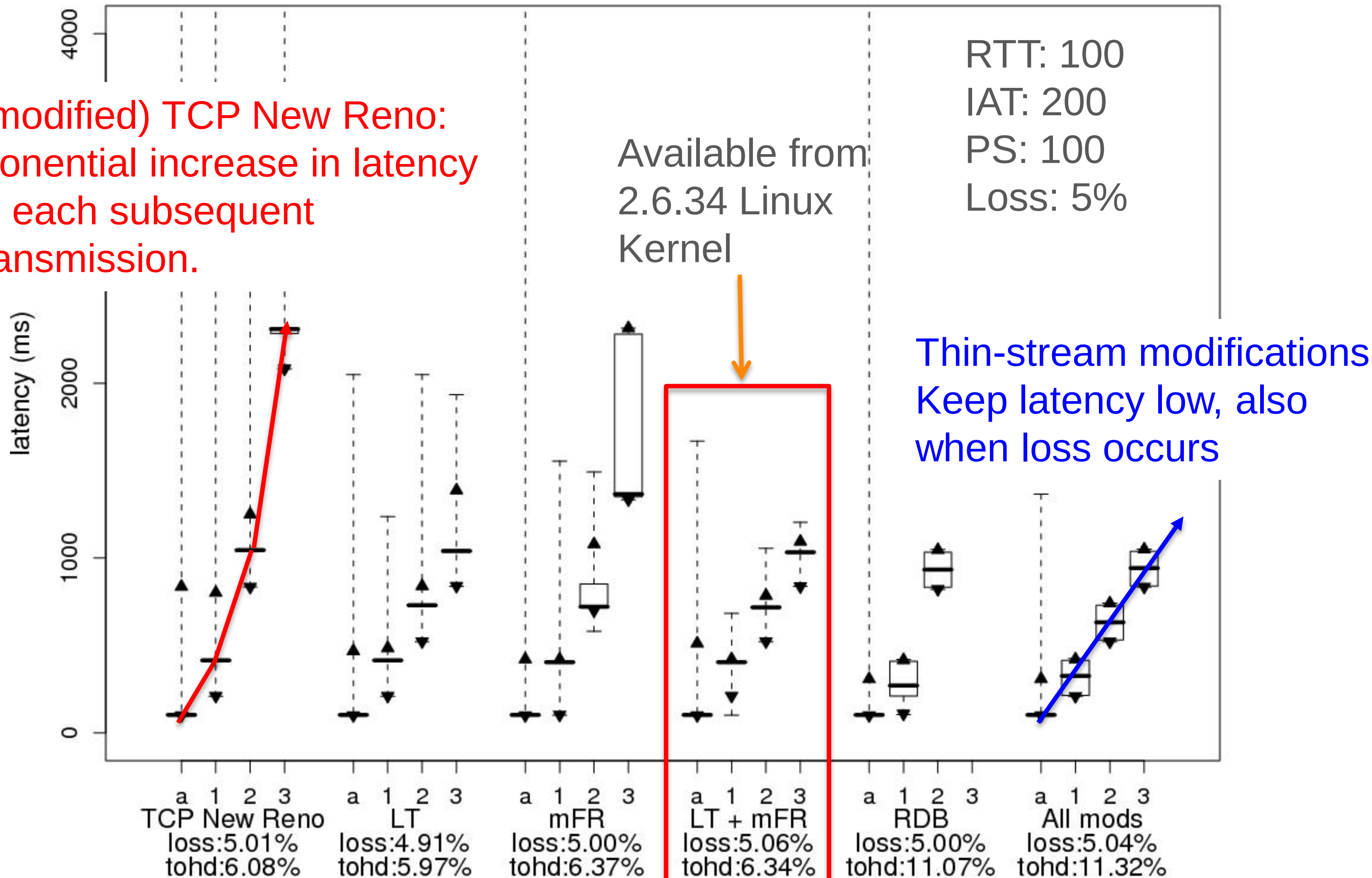
packets in flight (PIF) <= 4

Bundling:

size_unacked(p1) + size(p2) < MSS

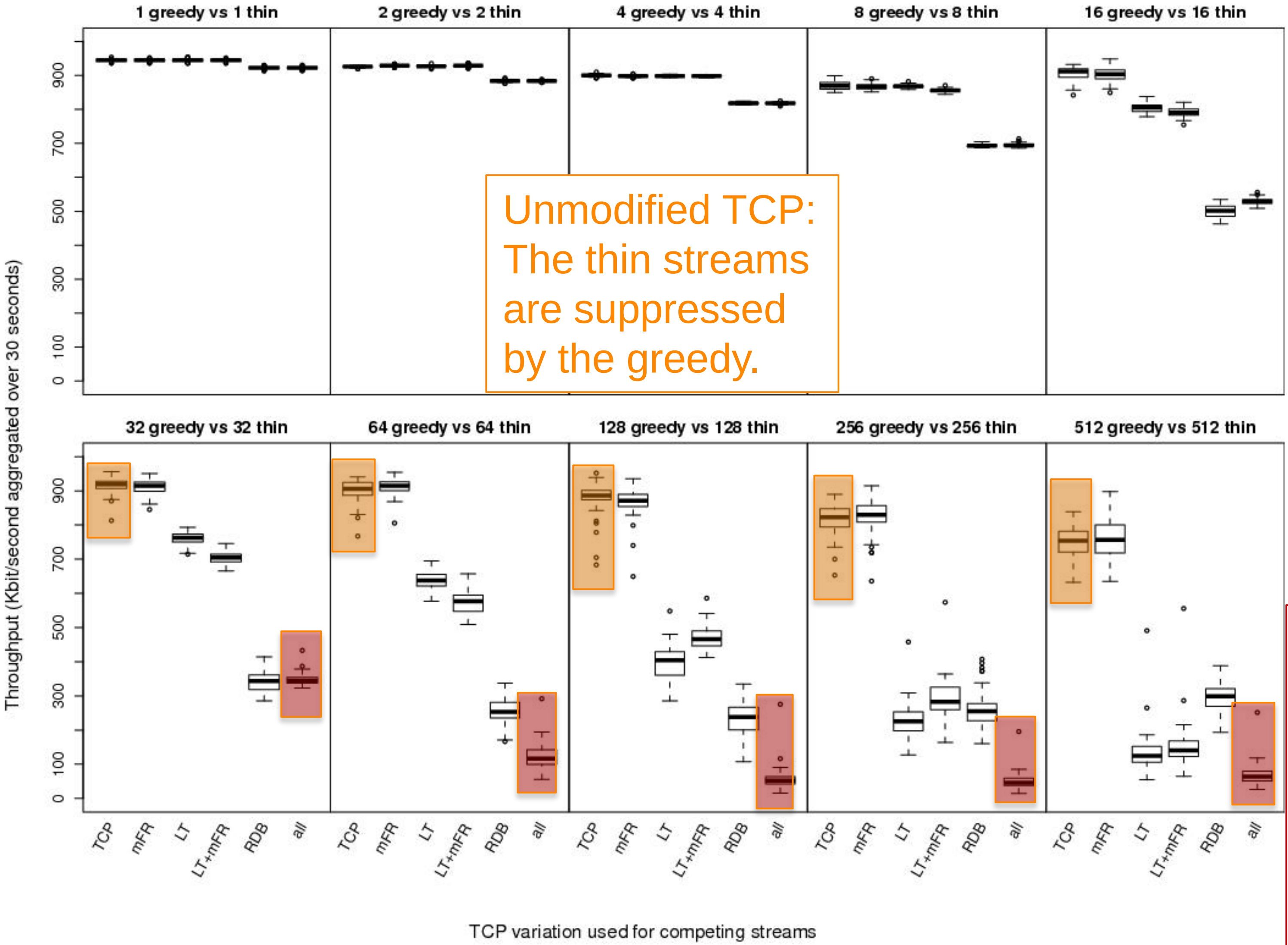
- **Modifications triggered <u>only</u> when these conditions are met.**
- **All modifcations are sender-side only. Tested to work with Windows (XP, Vista, 7), BSD, OSX and Linux as receivers.**

Test results and analysis example



Fairness

Packet-based drop strategy, small buffer

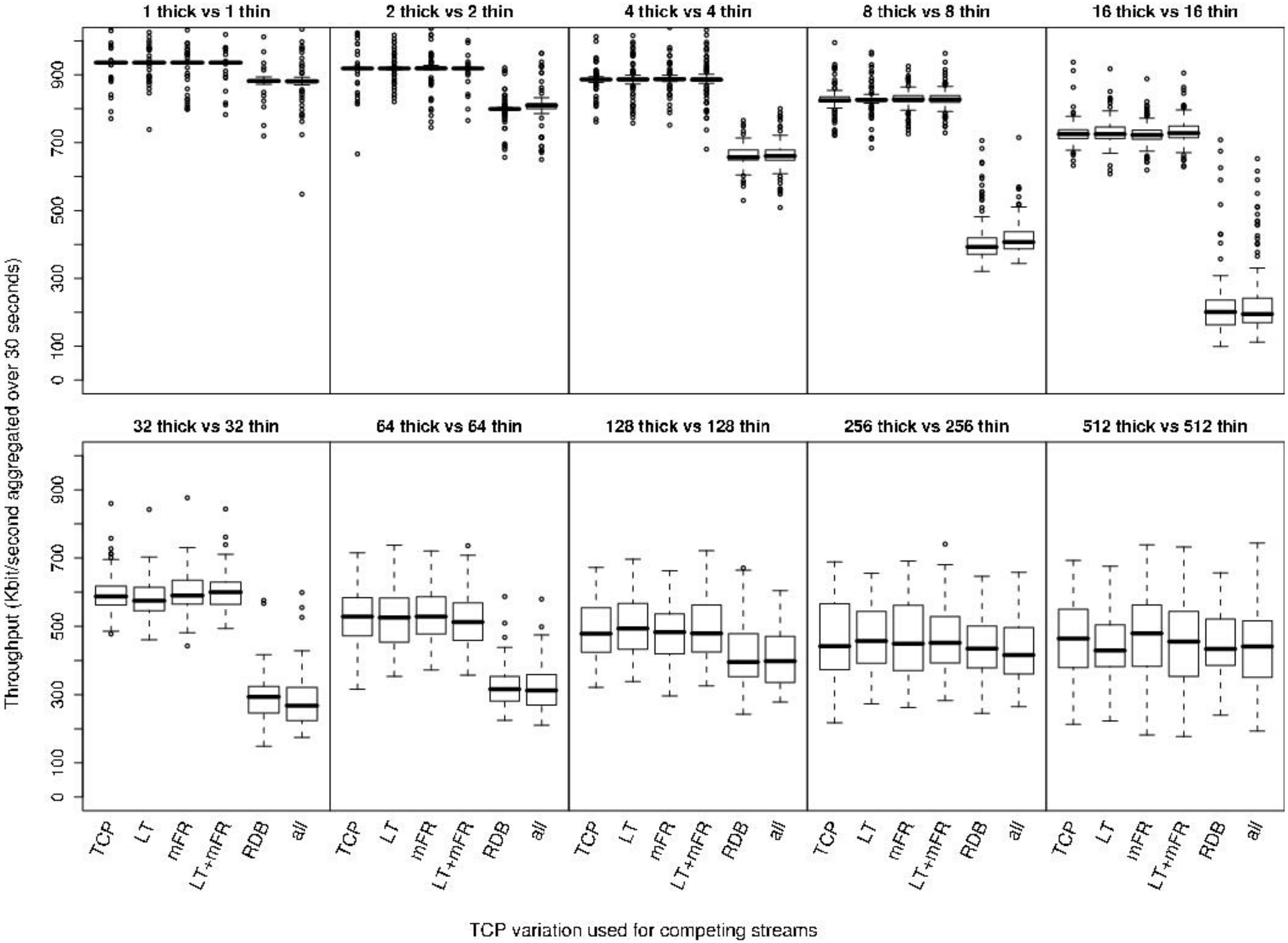


- Greedy stream goodput shown
- 1Mbps bottleneck
- 120 Bytes packets
- RTT 100ms

The basic bundling mechanism is too aggressive in very high congestion scenarios.

Fairness

Byte based drop strategy, large buffer



- •Greedy stream goodput shown
- •1Mbps bottleneck
- •120 Bytes packets
- •RTT 100ms
- •Behaviour depends on drop strategy and queue length.

Questions / Discussion

?

Thin stream vs Thick stream