TCP LoLa – Toward Low Latency and High Throughput Congestion Control

Mario Hock, Felix Neumeister, Martina Zitterbart, Roland Bless

Institute of Telematics (TM)

Motivation

- High Throughput and Low Delay
  - Typically considered as conflicting goals or trade-off
  - Not necessarily so: mitigate this trade-off
  - Approaches: AQM, Tweaks to existing CC (e.g., Alternative Back-off with ECN), New congestion controls

- Investigate how far we can get with a congestion control
  - Low queuing delay
  - High utilization/throughput
  - Scalable (also 10 Gbit/s and beyond)
  - RTT Fairness
  - Should work with regular tail-drop queues
  - Focus: Wide area networks (not Data Center)

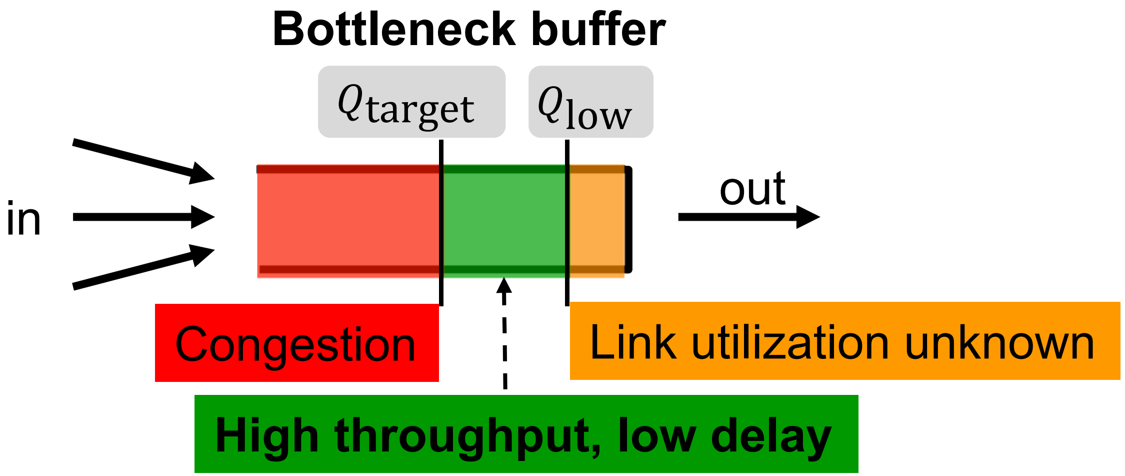
Objective and Challenges

- General goal
  - Determine a suitable amount of inflight data
    - achieving high bottleneck link utilization
    - avoid creating standing queues → keep queuing delay low
  - Configurable fixed target delay value
    - Congestion: persistent queuing delay above fixed target

- Challenge
  - Convergence to fairness: total amount of inflight data okay, but maybe unequal rate shares
  - Increase inflight data of one sender while reducing it for others
  - Interaction with small queue is more difficult
  - Without sacrificing the low delay goal!

TCP LoLa (Low Latency)

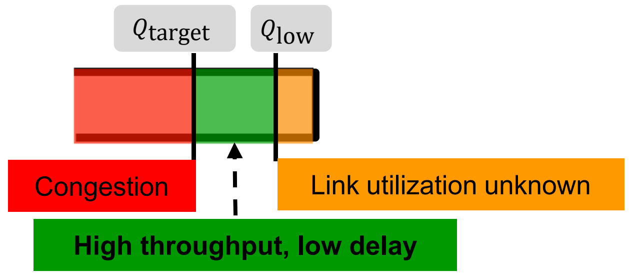
- Scalable approach based on queuing delay thresholds

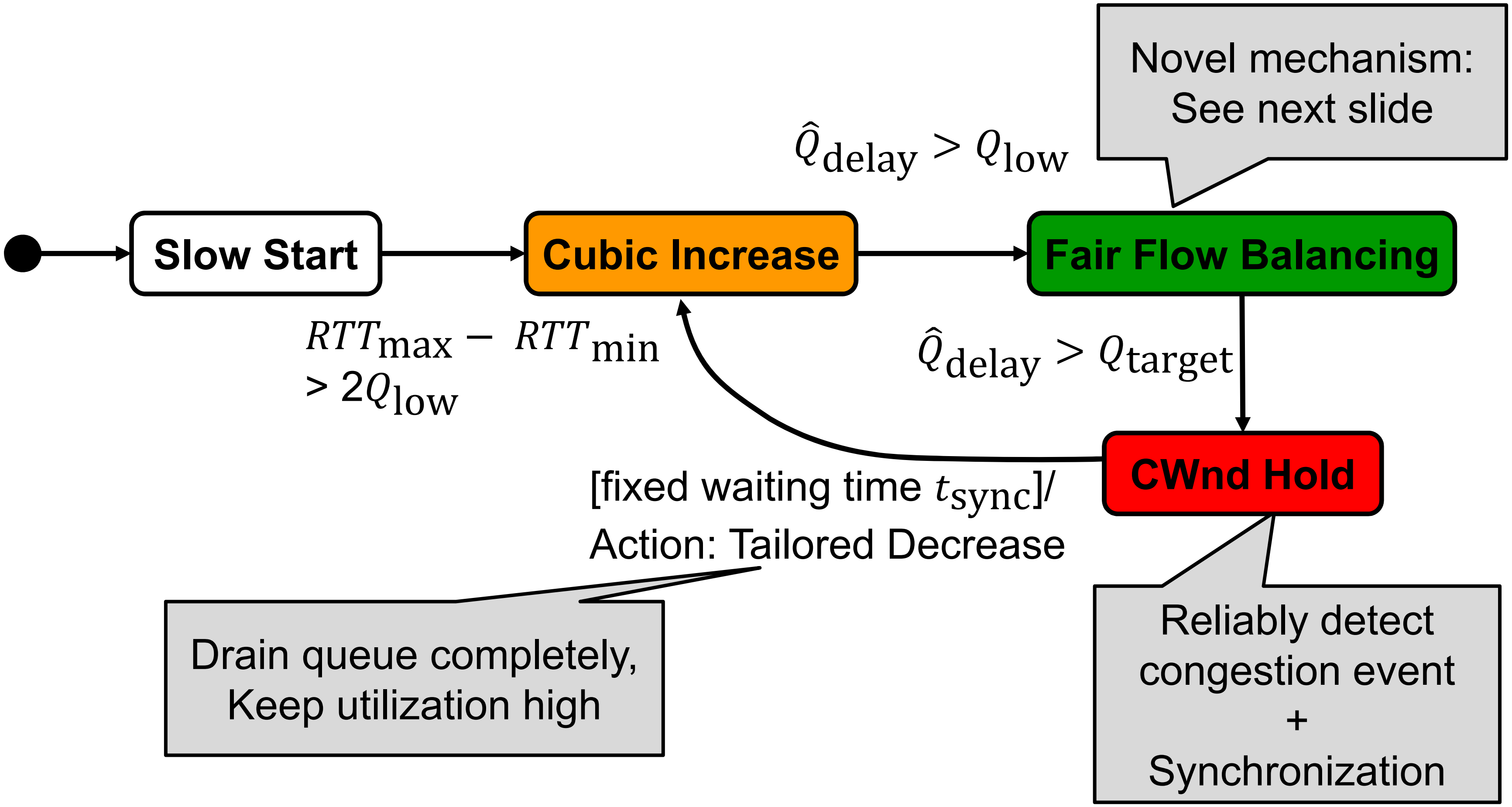


  - e.g., sender sets Q_{target}= 5 ms
- Estimate queuing delay
  - using min filter over fixed time period → measure standing queue
  - heuristic to adapt to network path changes (e.g., increasing RTT_{min})
- Congestion window-based approach
  - packet pacing is beneficial, but not necessary

TCP LoLa (Low Latency)

- Flow States

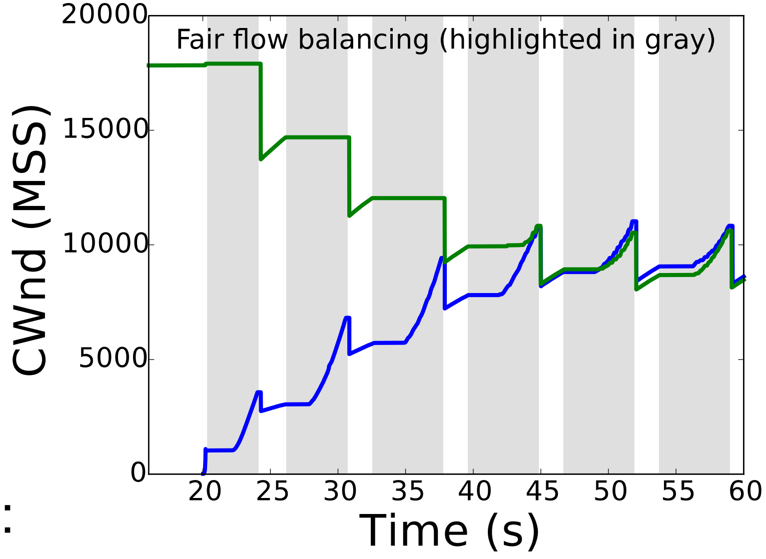




Fair Flow Balancing

- Novel convergence to fairness mechanism
  - Equalize amount of data that each flow may queue at the bottleneck
  - Dynamically scale allowed amount of data w.r.t. given delay target
  - Knowledge about current shares not available

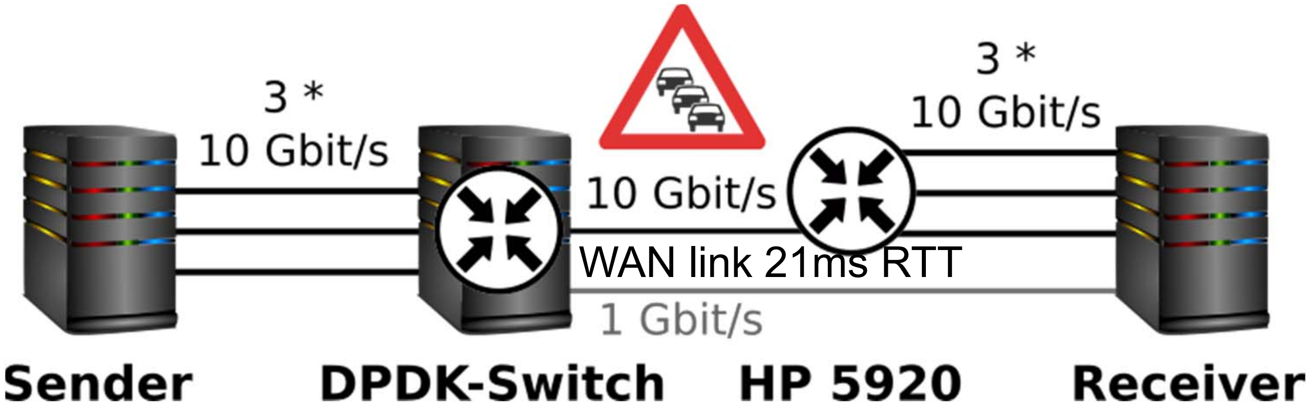
- Key concepts
  - Flow with more than fair share keeps CWnd
  - Flow with smaller than fair share increases CWnd
  - Allowed amount of data in queue X (fair queue share) is time dependent:



$\mathrm{X(t)} = \frac{t^3}{\phi}$, t: time since Q_{target} exceeded, ϕ: constant

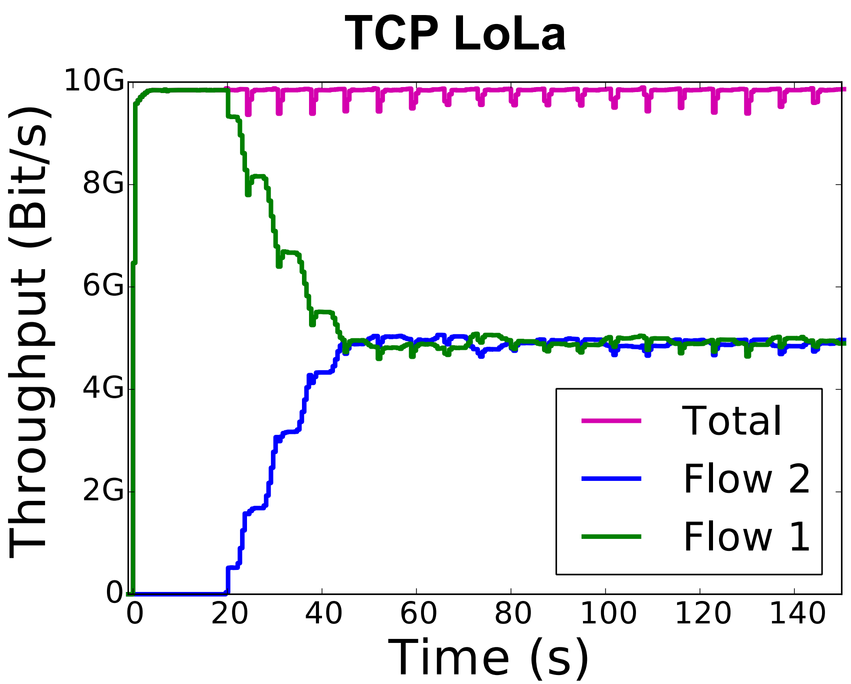
Testbed Setup

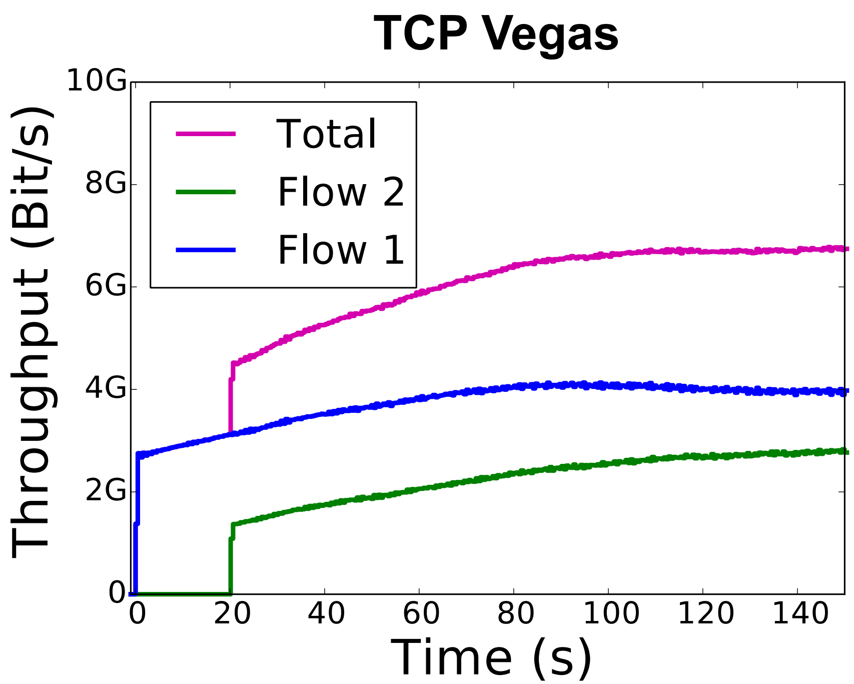
- Implemented as Linux Kernel module (Ubuntu 16.04)



- Packet pacing enabled
- Traffic generated with iperf3
- Q_{low}= 1ms, **Q_{target}= 5ms**, t_{sync}= 250ms, t_{measure}= 40ms

Evaluation Results – Throughput/Fairness

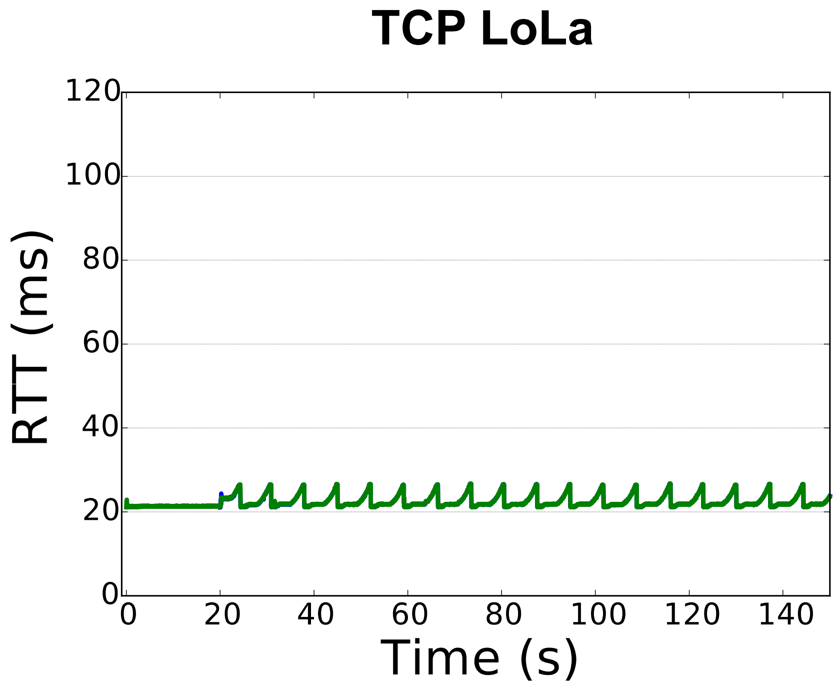


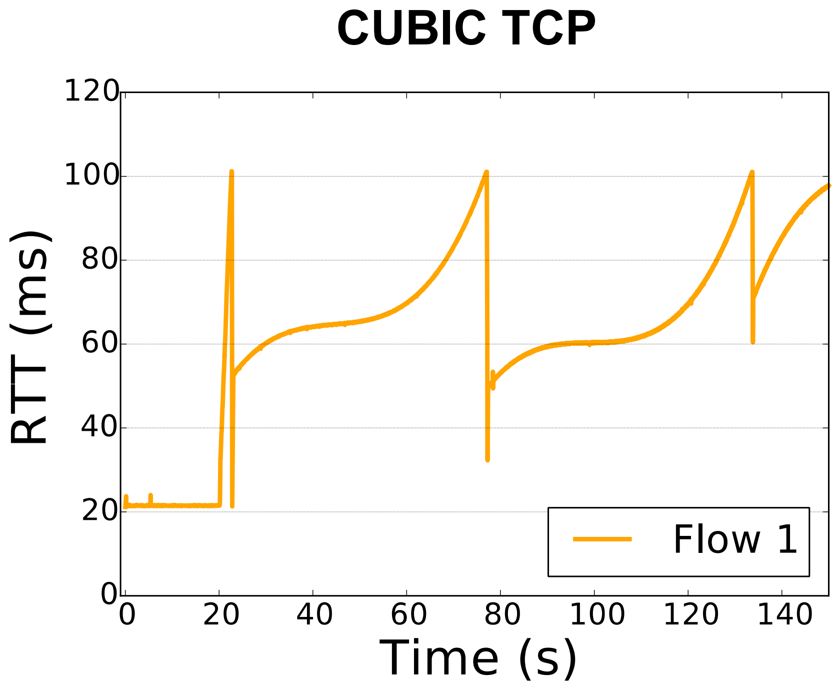


- High Throughput
- Convergence to Fairness

- Link cannot be fully utilized
- Misinterprets jitter as congestion

Low (Queuing) Delay

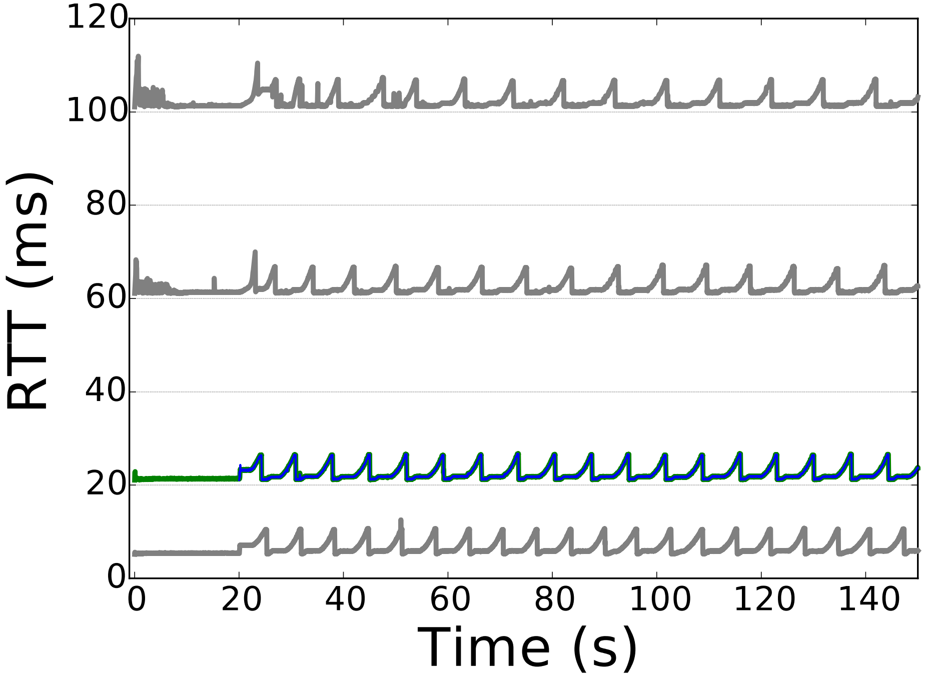




- Delay kept around target (~5ms queuing delay)
- Fills buffer completely

Queuing Delay – RTT Independent

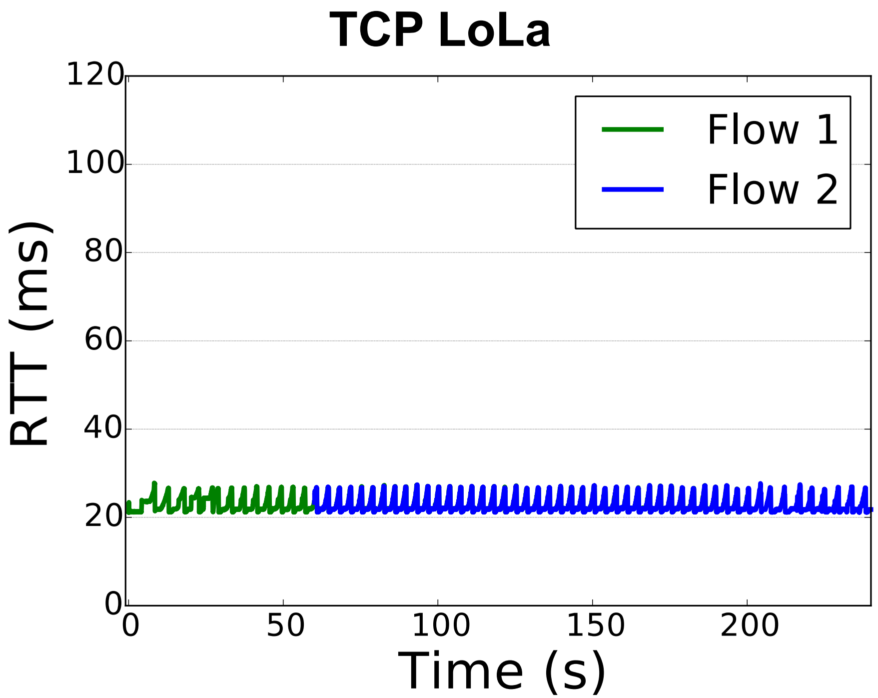
- Single flow, base RTT varied: 5ms, 61ms, 101ms

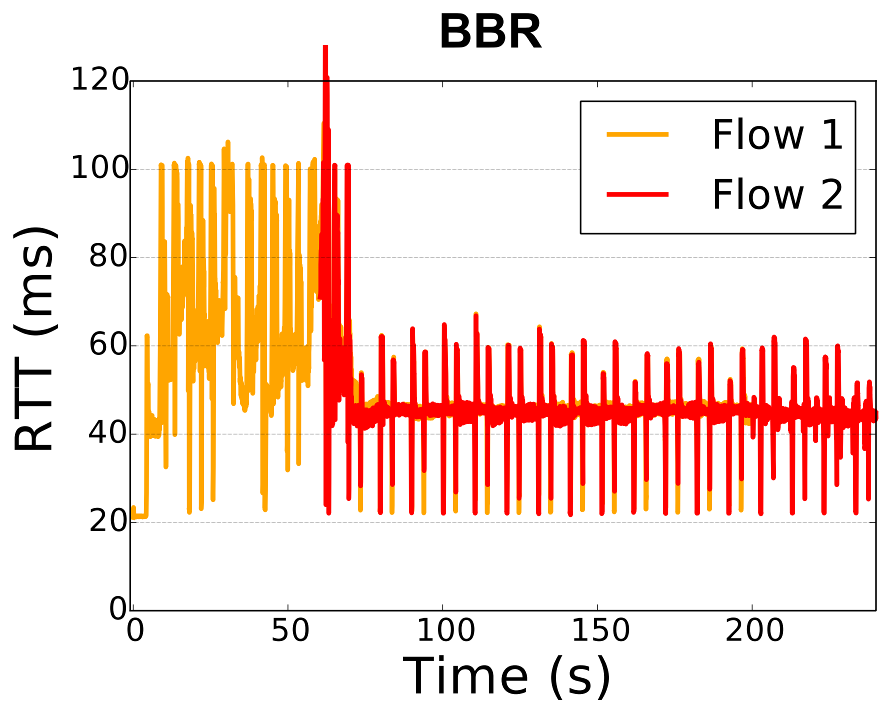


- LoLa keeps delay around target (~5ms queuing delay)
- Queuing delay is independent of
  - Base RTT (and rate – not shown here)
  - Number of senders (next slide)

Several Flows Starting in Succession

- RTTs measured at TCP senders

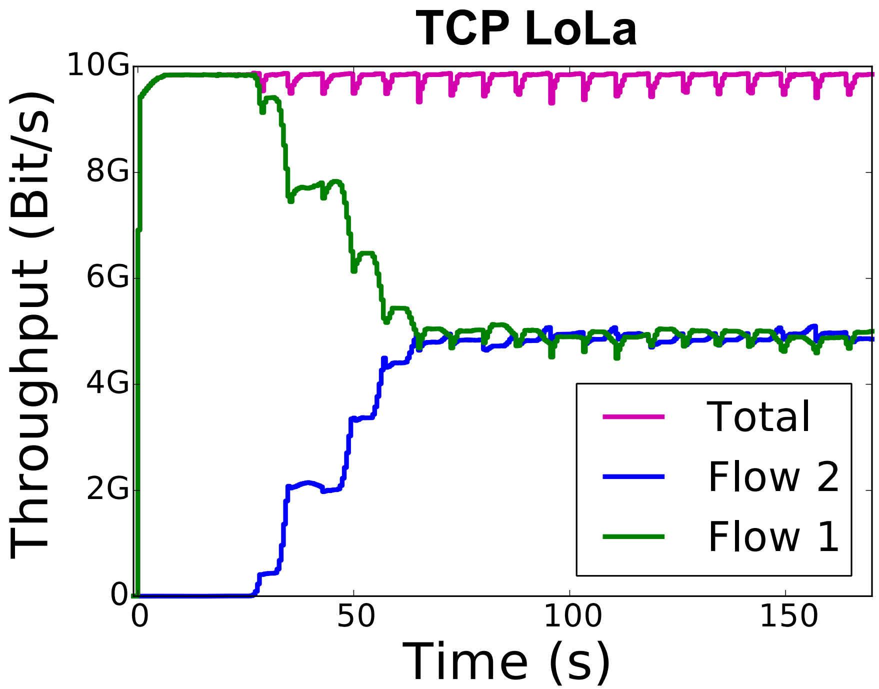


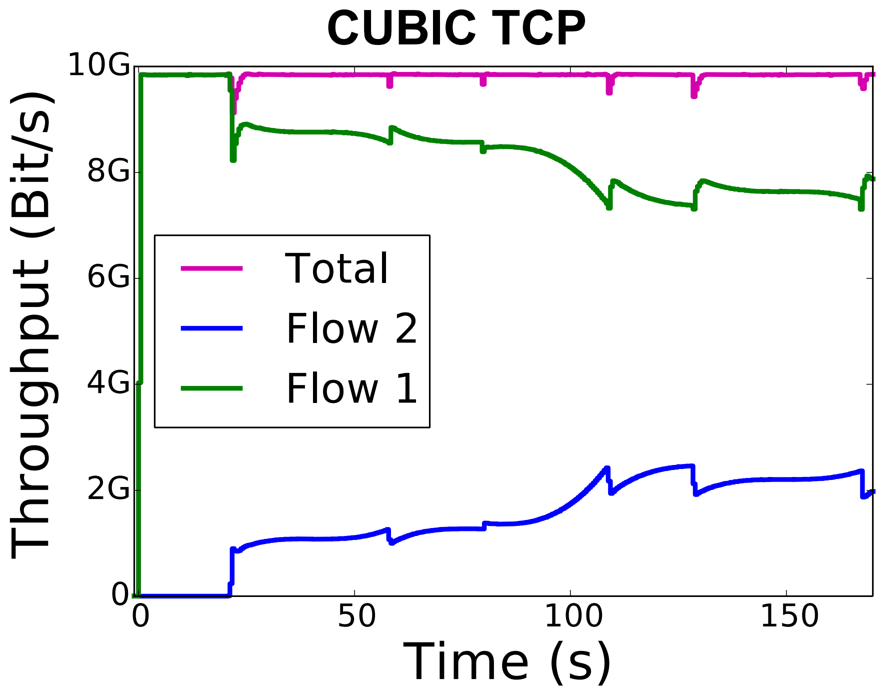


- LoLa still controls overall queuing delay
- No packet loss

- Starting flows fill the buffer
- Delay kept around $2RTT_{min}$
- > 1 Mio retransmissions

RTT Fairness





- Flow 1: 21 ms base RTT
- Flow 2: 101 ms base RTT
- Small buffer: 12.5 Mbyte
- Convergence to fair shares

- RTT unfairness
- Convergence to similar congestion windows
- No convergence to fair rate shares

Conclusion and Outlook

- First tests of the overall concepts [1]
  - Parameters not thoroughly optimized
  - Not a full-fledged TCP variant yet

- Further investigations
  - Use of one-way delay instead of RTT measurements
  - Influence of delayed and compressed ACKs
  - Performance in wireless environments
  - Multiple bottleneck scenarios
  - Coexistence with loss-based variants (separate queues, AQMs, …) [2]
- Planned
  - Use of explicit feedback from the network

References

[1] M. Hock, F. Neumeister, M. Zitterbart, R. Bless: „TCP LoLa: Congestion Control for Low Latencies and High Throughput“, IEEE 42nd Conference on Local Computer Networks, Singapore, Oct 9–12, 2017

[2] M. Hock, R. Bless, M. Zitterbart, „Toward Coexistence of Different Congestion Control Mechanisms”, 2016 IEEE 41st Conference on Local Computer Networks, pp. 567–570, Dubai, United Arab Emirates, November 2016

BACKUP SLIDES

Formulas

- RTT measurements

$$\widehat{RTT}_{now} = \min\{\mathrm{RTT}(t_k) \mid t_k \in [t - t_{\mathrm{measure}}, t]\}$$

$$\widehat{RTT}_{min} := \min(\widehat{RTT}_{now}, \widehat{RTT}_{min})$$

Reset $\widehat{RTT}_{min}$ if $\widehat{RTT}_{now}$ hasn‘t been measured
close $\widehat{RTT}_{min}$ for 100 tailored decreases

$$\hat{Q}_{\mathrm{delay}} = \widehat{RTT}_{now} - \widehat{RTT}_{min}$$

- Cubic Increase / Tailored Decrease

$$\mathrm{CWnd}(t) = C \cdot (t - K)^3 + \mathrm{CWnd}_{\mathrm{max}}$$

$$K = \sqrt[3]{\left(\mathrm{CWnd}_{\mathrm{max}} - \widehat{RTT}_{min} \cdot \frac{\mathrm{CWnd}_{\mathrm{max}}}{\widehat{RTT}_{now}} \cdot \gamma\right) / C}$$

- Fair Flow Balancing

$\mathrm{X}(\mathrm{t}) = \frac{t^3}{\phi}$, t: time since Q_{target} exceeded,
X allowed amount of queued data, ϕ: constant