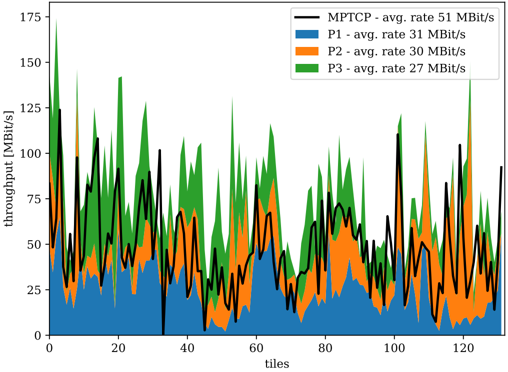
An Odd Couple: Loss-Based Congestion Control and Minimum RTT Scheduling in MPTCP

Ralf Lübben
University of Applied Science Flensburg

Johannes Morgenroth
Robert Bosch GmbH, Corporate Research, Hildesheim

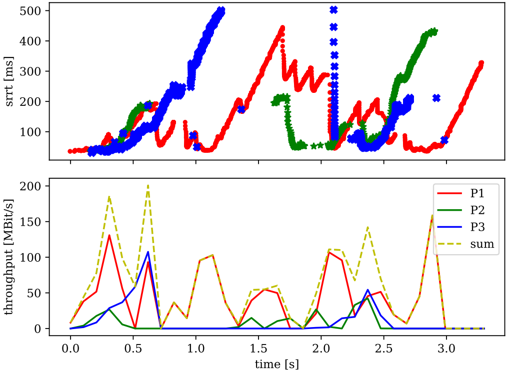
October 16, 2019

Singlepath vs. Multipath TCP Throughput - Drive Test



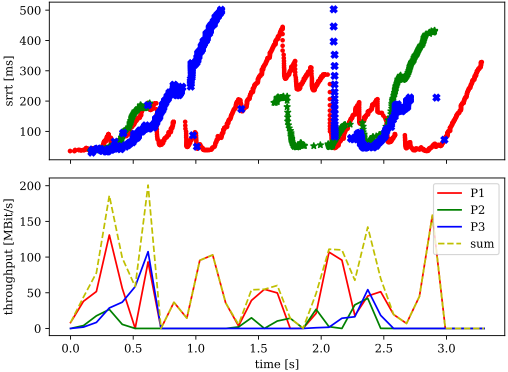
- real-world drive test in cellular networks
- about 30 MBit/s throughput for each provider
- 51 MBit/s aggregated throughput
- MPTCP cannot exploit paths fully

Exemplary Trace



- RTT increase dramatically
- throughput drops to zero for the same time
- occurs in turns between providers
- indicates systematic issue with congestion control and scheduling

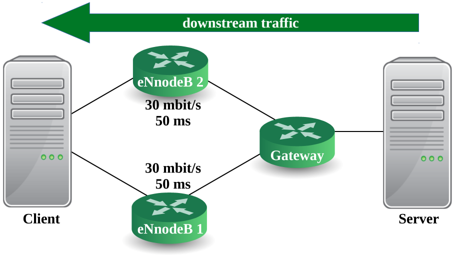
Exemplary Trace



- RTT increase dramatically
- throughput drops to zero for the same time
- occurs in turns between providers
- indicates systematic issue with congestion control and scheduling

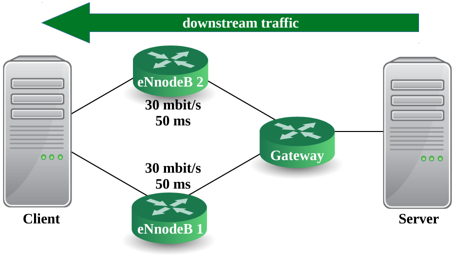
Details in **Application Level Performance Measurements of Multi-Connectivity Options in Cellular Networks for Vehicular Scenarios** presented yesterday as short-paper, poster.

Measurement Setup



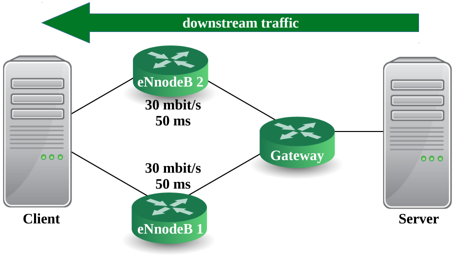
- simplified multi-path scenario
- deep queues (3000 pkts) at nodes close to the client
- reproduce mobile network

Measurement Setup



- simplified multi-path scenario
- deep queues (3000 pkts) at nodes close to the client
- reproduce mobile network
- **congestion control: NewReno**

Measurement Setup



- simplified multi-path scenario
- deep queues (3000 pkts) at nodes close to the client
- reproduce mobile network
- **congestion control: NewReno**
- **scheduling: MinRTT**

Bandwidth Delay Product and MPTCP

- Iperf3 throughput measurements

Bandwidth Delay Product and MPTCP

- Iperf3 throughput measurements
- single path BDP is 366 kByte

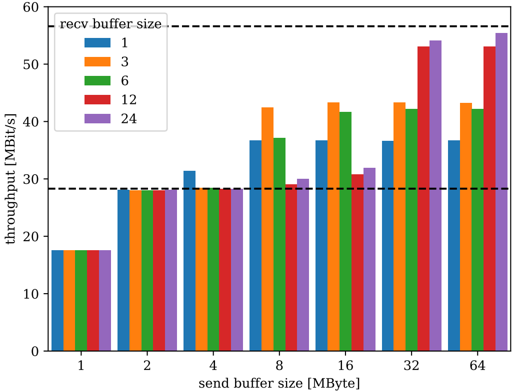
Bandwidth Delay Product and MPTCP

- Iperf3 throughput measurements
- single path BDP is 366 kByte
- send buffer / receiver buffer setting of 1 MByte / 768 kByte achieves full utilization

Bandwidth Delay Product and MPTCP

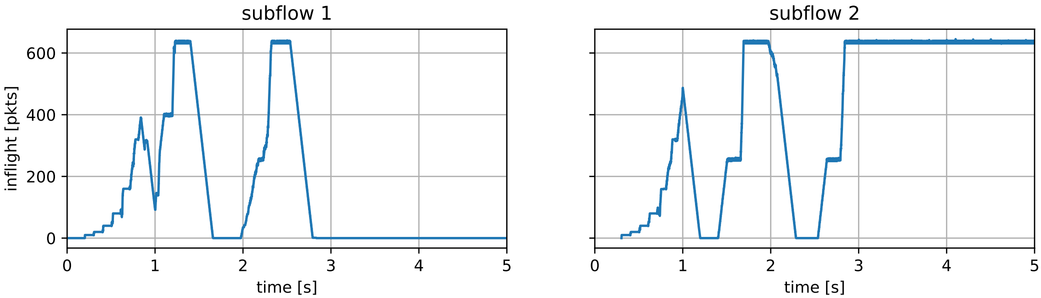
- Iperf3 throughput measurements
- single path BDP is 366 kByte
- send buffer / receiver buffer setting of 1 MByte / 768 kByte achieves full utilization
- the difference in BDP and buffer space is due to overhead in memory consumption and sequence number space

Bandwidth Delay Product and MPTCP



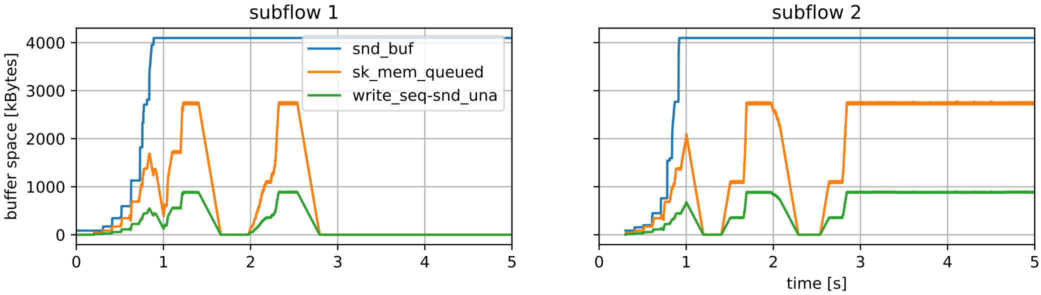
- Iperf3 throughput measurements
- single path BDP is 366 kByte
- send buffer / receiver buffer setting of 1 MByte / 768 kByte achieves full utilization
- the difference in BDP and buffer space is due to overhead in memory consumption and sequence number space
- MPTCP requires much larger buffer sizes for full utilization

Packets in Flight



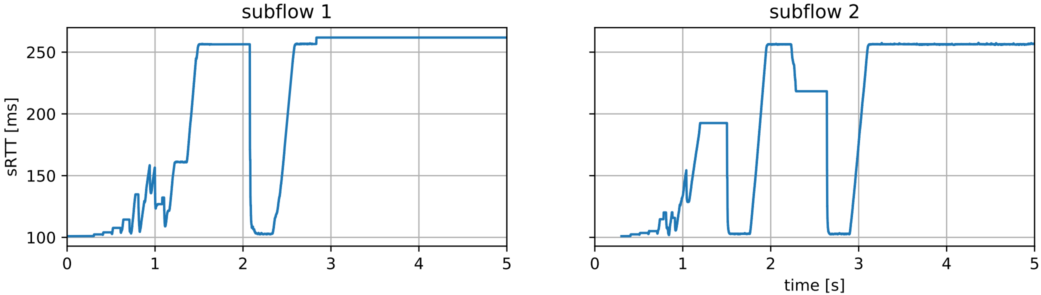
1. both subflows ramp up (slow start)
2. subflows are used alternately
3. one subflow starves

Send Buffer Limitation



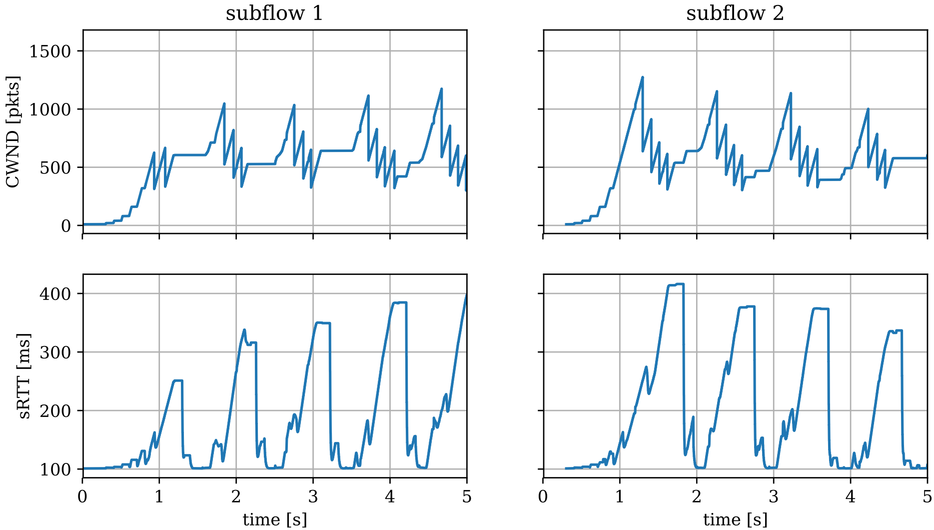
- memory space increases to 4 MByte
- subflow 1: used memory and sequence number space drops to zero
- subflow 2: fully utilizes the memory and sequence number space

RTT



- RTT of subflow 1 is slightly higher
- minRTT scheduling: subflow 1 is not scheduled and RTT is never updated
- subflow 2 cannot increase traffic rate due to buffer limitations
- RTT of subflow 2 never exceeds RTT of subflow 1

Recv Buffer Limitation



Recv Buffer Limitation

MPTCP implements a receive buffer optimization:

- if the recv window limits the transmission, the flow is not selected anymore
- the CWND of the non-selected sub-flow is halved
- decreasing the CWND of the non-selected subflow is safe
- paths are used in turns due to CWND halving

Recv Buffer Limitation

MPTCP implements a receive buffer optimization:

- if the recv window limits the transmission, the flow is not selected anymore
- the CWND of the non-selected sub-flow is halved
- decreasing the CWND of the non-selected subflow is safe
- paths are used in turns due to CWND halving

MPTCP **send buffer optimization**:

- not as straight forward as receive buffer optimization
- here, the active flow is the problem, not the inactive flow

Mitigation

1. Tackling bufferbloat[1]:

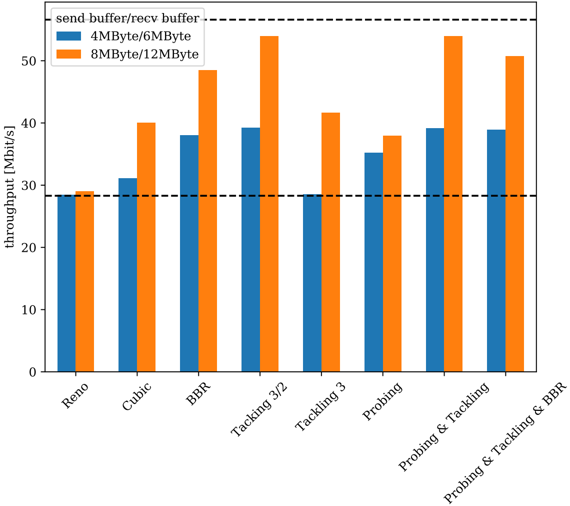
$$CWND = \begin{cases} \lambda \frac{RTT_{min}}{sRTT} CWND, & \text{if } \frac{sRTT}{RTT_{min}} \geq \lambda \\ CWND, & \text{otherwise} \end{cases}$$

   with $\lambda \in \{\frac{3}{2}, 3\}$

2. Tail burst probing: send TCP probe, when no RTT update in minimal RTT
3. Avoid large buffers: use BBR as congestion control

[1] Tackling the challenge of bufferbloat in Multi-Path Transport over heterogeneous wireless networks, Simone Ferlin-Oliveira, Thomas Dreibholz, Özgü Alay

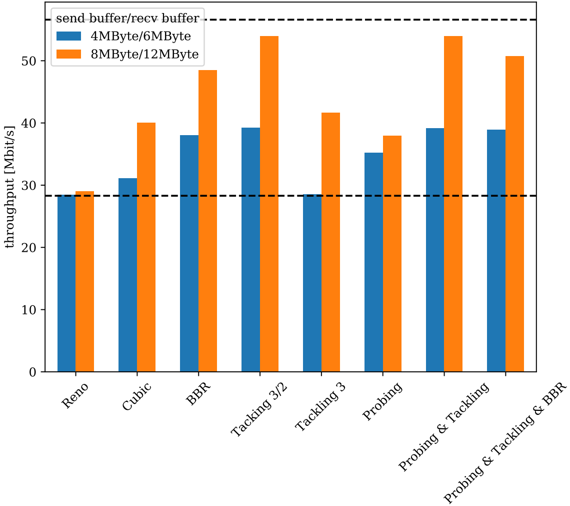
Results



Tackling bufferbloat:

- small buffers: improves for $\lambda = \frac{3}{2}$ but not for $\lambda = 3$
- large buffers: improves for $\lambda = \frac{3}{2}$ also for $\lambda = 3$
- setting λ is challenging

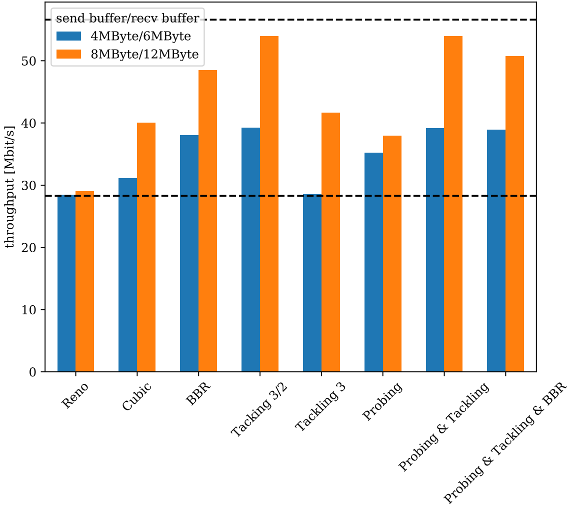
Results



Probing:

- improves the overall throughput
- improvement is limited
- subflows block for short duration before a probe gives a new estimate

Results

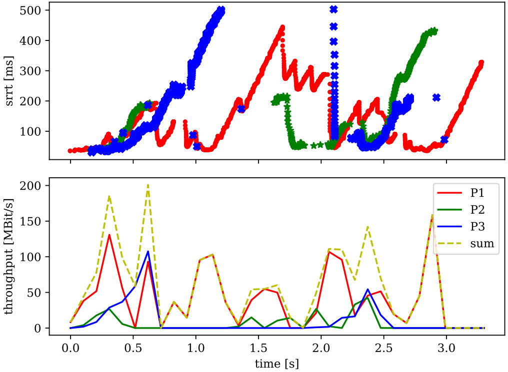


BBR:

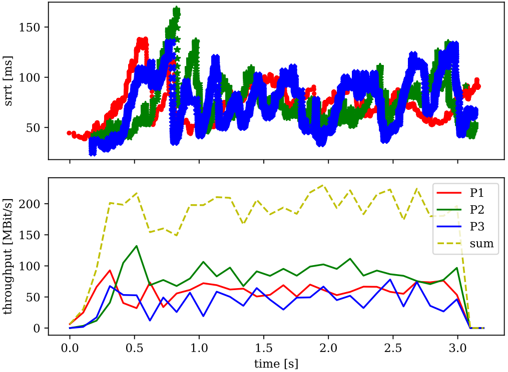
- increases throughput similar to the tackling approach
- avoids filling buffers and network queues by bandwidth estimation
- avoids missleading scheduling decisions
- no direct parameter setting is required
- BBR has other known issues: fairness

Comparision Traffic Trace (Real-World Cellular Network)

Cubic:



BBR:



Conclusion

- MinRTT scheduling and loss-based congestion control interfere with each other.
- Large network queues enforce the negative interference.
- Large send/receive buffers significantly above path BDP are required.
- Tail burst probe, tackling, advanced congestion control mitigate the problem.
- Each approach concerns a different issue.