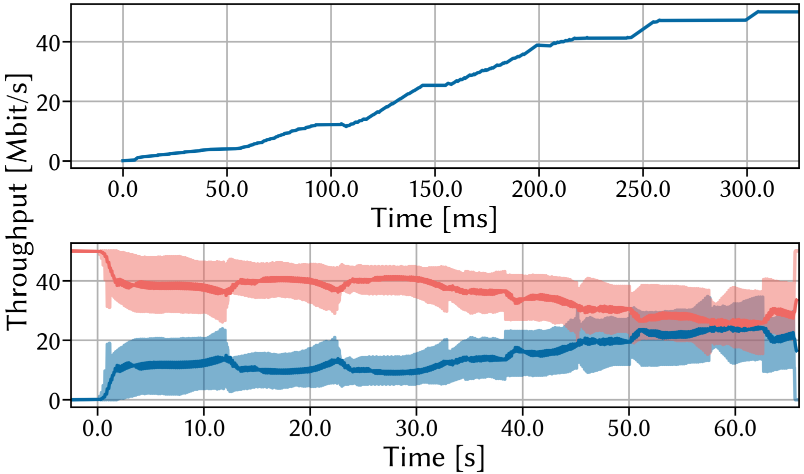
Blitz-starting QUIC connections

Rapidly starting QUIC connections

Jan Rüth, Konrad Wolsing, Martin Serror, Klaus Wehrle, Oliver Hohlfeld

Rapidly starting QUIC connections

- **Short connections are often stuck in slow start**
  - Worse: They often exit it too early



  - Blue flow exits slow start early → slow fairness convergence and bad performance
  - Even regular slow start impeds performance

Replacing slow start

- **Slow start should probe the bandwidth to not cause a collapse**
  - It fails to discover its fair share
- **One must induce congestion to others to gain bandwidth**
  - Push more packets into the network
  - But at a reasonable pace
- **Skip slow start at connection start and go to congestion avoidance**
  - Bootstrap the congestion window with an appropriate rate
- **Let the client signal the rate that it thinks it can handle**
  - Knows other parallel flows
  - Knows access technology

How has this been done before?

- **Statically increase the initial congestion window [Dukkipati et al. CCR10]**
  - IW10 does not fit all networks
- **Riptide [Flores et al., ICDCS16], SmartIW [Nie et al., IWQOS18]**
  - “Learn” IW and customize from past connections
- **Quickstart: Path signaling of bandwidth [Floyd et al., RFC 4782]**
  - Hard to deploy
- **Mobile Throughput Guidance [Jain et al., draft-flinck-mobile-throughput-guidance-04.txt]**
  - Authenticated TCP option inserted on path with bandwidth information
- **Jumpstart: Simply paces out all available packets [Liu et al. PFLD07]**
  - RWIN used to signal bandwidth
- **Halfback: Jumpstart but with opportunistic retrans. [Li et al., CoNEXT15]**
  - Better than RACK?

Client-Side Bandwidth Estimation

- **We assume congestion to happen at the network edge**
  - Not always true

- **At the edge**
  - Wireless: Use current coding scheme to calculate PHY-rate (available on all popular OSes)
  - Mobile: Using band, bandwidth, modulation and MIMO calculation of LTE PHY-rate
  - Ethernet connected at home
    - Past maximum speed measurements
    - Ask local gateway, e.g., TR-064 allows to determine the available link-speed to the next hop

- **We did not investigate how well any of these work!**
  - We will look at what happens if we have a bad estimate

Evaluation

- **Modified Google QUIC**
  - Clients can signal bandwidth as a transport parameter in the initial packet
  - Server uses bandwidth and RTT sample to calculate congestion window and goes to CA
- **Compete against an unmodified Google QUIC**
  - Cubic congestion control with IW32 and pacing (0.5 RTT in SS, 0.75 RTT in CA)
- **We used Mininet to emulate various network conditions**
  - We know the true available bandwidth, buffers and delays...
- **We measure flow completion times for**
  - 70kB, average size of Google landing page in 2017
  - 2MB, video chunk or whole website
  - 10MB, larger files or objects

Evaluation

- **We announce 0.5, 1.0, 1.5, 3.0, 4.0x the true bandwidth to the server**
- **We compare against a single competing elephant flow that is already utilizing the link**
- **We repeat 30 times to gain statistical significance (we show 95% conf.)**
  - We perform an ANOVA test to see if there is statistical difference

Results

x BW estimate	DSL slow RTT=50 ms, BW=25 MBit, BUF=50 ms (104 pkt) 70KB FCT	70KB Loss	2MB FCT	2MB Loss	10MB FCT	10MB Loss
0.5	x1.3 ▲(orange) +47ms	x4.0 ▼(green) -13	x1.1 ▼(green) -170ms	x1.5 ▼(green) -10	x1.0 ●(gray) -104ms	x1.2 ▼(green) -8
1.0	x1.0 ●(gray) +2ms	x1.0 ●(gray) 0	x1.4 ▼(green) -562ms	x3.4 ▲(red) +69	x1.2 ▼(green) -1.2s	x2.7 ▲(red) +79
1.5	x1.1 ▲(orange) +19ms	x1.3 ▲(red) +5	x1.6 ▼(green) -744ms	x7.5 ▲(red) +190	x1.3 ▼(green) -1.6s	x5.8 ▲(red) +219
3.0	x1.2 ▲(orange) +40ms	x2.0 ▲(red) +17	x1.9 ▼(green) -939ms	x25.2 ▲(red) +708	x1.4 ▼(green) -2.2s	x17.0 ▲(red) +735
4.0	x1.3 ▲(red) +52ms	x2.3 ▲(red) +23	x2.0 ▼(green) -998ms	x37.2 ▲(red) +1061	x1.4 ▼(green) -2.3s	x25.0 ▲(red) +1105

x BW estimate	DSL fast RTT=50 ms, BW=50 MBit, BUF=50 ms (208 pkt) 70KB FCT	70KB Loss	2MB FCT	2MB Loss	10MB FCT	10MB Loss
0.5	x1.0 ●(gray) -3ms	x1.5 ▼(green) -3	x1.4 ▼(green) -407ms	x1.3 ●(gray) +9	x1.2 ▼(green) -905ms	x1.6 ▲(red) +17
1.0	x1.0 ●(gray) +6ms	x1.9 ▲(red) +7	x1.9 ▼(green) -664ms	x7.2 ▲(red) +176	x1.5 ▼(green) -1.7s	x7.4 ▲(red) +195
1.5	x1.1 ●(gray) +10ms	x2.2 ▲(red) +10	x2.2 ▼(green) -773ms	x15.6 ▲(red) +412	x1.7 ▼(green) -2.0s	x15.8 ▲(red) +449
3.0	x1.1 ●(gray) +10ms	x2.9 ▲(red) +16	x2.8 ▼(green) -893ms	x37.8 ▲(red) +1036	x2.1 ▼(green) -2.7s	x42.4 ▲(red) +1254
4.0	x1.1 ▲(orange) +14ms	x3.2 ▲(red) +18	x2.9 ▼(green) -915ms	x40.5 ▲(red) +1115	x2.3 ▼(green) -2.9s	x57.0 ▲(red) +1696

x BW estimate	3G RTT=90 ms, BW=8 MBit, BUF=200 ms (140 pkt) 70KB FCT	70KB Loss	2MB FCT	2MB Loss	10MB FCT	10MB Loss
0.5	x1.1 ●(gray) -51ms	x3.8 ▼(green) -16	x1.0 ●(gray) +454ms	x3.5 ▼(green) -22	x1.0 ●(gray) +979ms	x2.0 ▼(green) -20
1.0	x1.0 ●(gray) +4ms	x1.0 ●(gray) +1	x1.3 ▼(green) -2.2s	x1.2 ▲(red) +7	x1.1 ▼(green) -2.3s	x1.2 ▲(red) +9
1.5	x1.0 ●(gray) +27ms	x1.3 ▲(red) +7	x1.5 ▼(green) -3.2s	x2.4 ▲(red) +45	x1.2 ▼(green) -4.1s	x2.2 ▲(red) +47
3.0	x1.1 ▲(red) +110ms	x1.8 ▲(red) +18	x1.9 ▼(green) -4.3s	x8.4 ▲(red) +230	x1.3 ▼(green) -6.7s	x6.8 ▲(red) +233
4.0	x1.1 ▲(red) +106ms	x1.9 ▲(red) +19	x2.1 ▼(green) -4.8s	x13.1 ▲(red) +376	x1.5 ▼(green) -9.3s	x10.5 ▲(red) +383

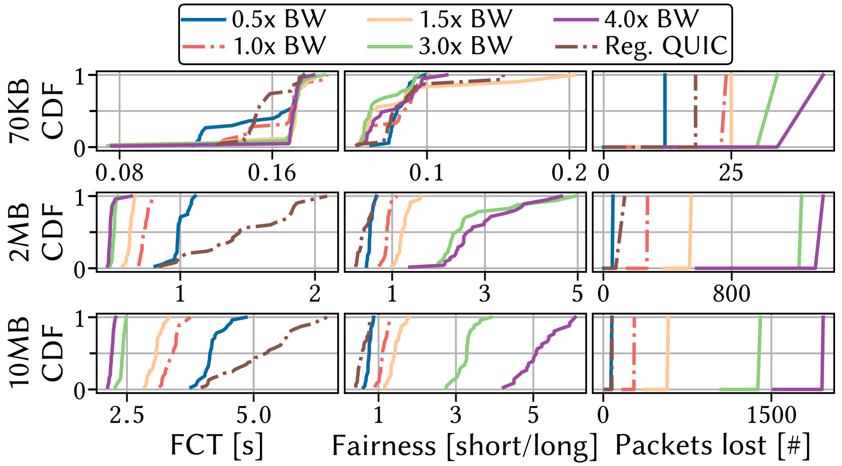
x BW estimate	LTE RTT=70 ms, BW=32 MBit, BUF=200 ms (560 pkt) 70KB FCT	70KB Loss	2MB FCT	2MB Loss	10MB FCT	10MB Loss
0.5	x1.8 ▼(green) -214ms	x1.1 ●(gray) 0	x1.5 ▼(green) -1.9s	x1.9 ●(gray) +5	x1.1 ●(gray) -1.4s	x1.3 ●(gray) -4
1.0	x1.8 ▼(green) -223ms	x2.2 ●(gray) +2	x2.5 ▼(green) -3.4s	x9.5 ▲(red) +45	x1.6 ▼(green) -6.4s	x3.1 ▲(red) +33
1.5	x1.9 ▼(green) -235ms	x3.2 ▲(red) +3	x3.0 ▼(green) -3.8s	x26.8 ▲(red) +137	x2.1 ▼(green) -8.7s	x8.8 ▲(red) +122
3.0	x2.1 ▼(green) -259ms	x6.5 ▲(red) +7	x3.9 ▼(green) -4.2s	x114.5 ▲(red) +602	x2.9 ▼(green) -10.7s	x44.2 ▲(red) +678
4.0	x1.8 ▼(green) -224ms	x7.9 ▲(red) +9	x3.9 ▼(green) -4.2s	x195.1 ▲(red) +1029	x3.3 ▼(green) -11.3s	x71.6 ▲(red) +1109

- ▲: An improvement
- ▼: A deterioration
- ▲ ▼: Changes within the same RTT
- ●: No significant difference in the 95% confidence intervals

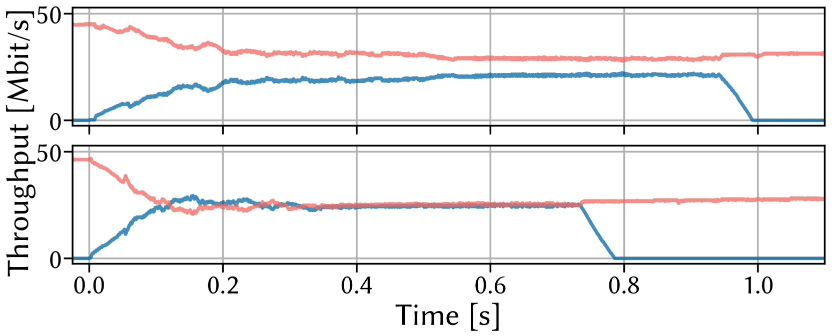
- **We can trade little additonal bandwidth for increased performance**
  - If the bandwidth estimate is good!

Results

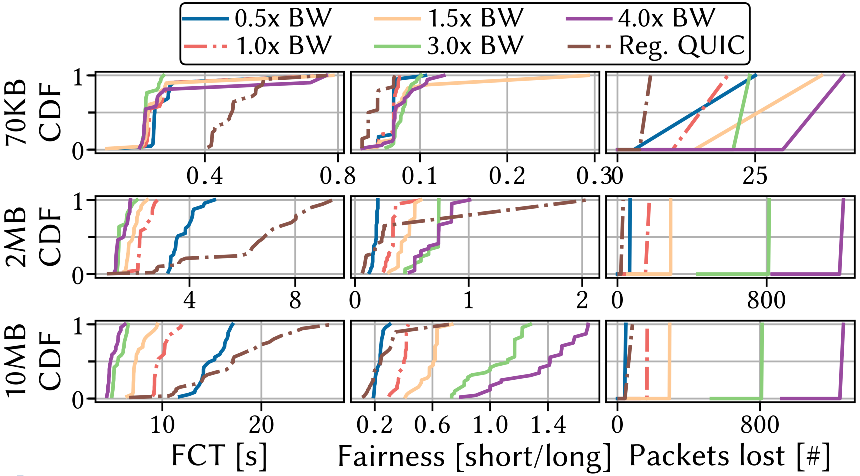
- **DSL Fast**
  - 70kB too few bytes to get close to fairness
  - FCT CDFs steeper than slow start
    - Consistent results!
  - Overestimations lead to significant loss



- **2MB DSL fast**
  - Best regular QUIC vs. median blitz-started
  - Much better fairness
  - Constant over flow's lifetime
  - Faster finish



Results



- **LTE setting in detail**
  - Fairness hard to reach
  - Large buffers set off the estimate
    - CUBIC will fill the buffers and overestimate the cwnd
  - A slight overestimation whould be fine here

Discussion

- **Initial results look nice but**
  - Only one flow competes
    - How realistic?
    - Would we still announce the full bandwidth?
  - What if multiple flows do this in parallel?
    - We only tested against a single elephant flow
    - What if resources are discovered on a webpage and are then loaded using Blitzstart?
  - How do other congestion control algorithms react, e.g., BBR?
  - How accurate are the bandwidth estimates that one would get in reality?
    - What if the client lies about her bandwidth? Could a sender detect lies?
      - How well does it fit to a previous connections?
    - Are bandwidth estimates a privacy concern? Could you fingerprint a user?
  - Could we use the bandwidth estimate differently?

Conclusion

- **We revived an old idea that can easily be deployed with QUIC**
  - Works well in easy settings
  - Yields good fairness and fast transmission times

- **Real-world applicability?**
  - Must be tested with more flows
  - When more flows are starting
  - How accurate are bandwidth estimates?

- **Lots of future work ☺**