facebook

INFRASTRUCTURE

Comparing COPA with CUBIC, BBR for live video upload

Nitin Garg

Software Engineer

Videos Infra, Facebook

Outline

- Motivation
- Experiment Setup and results
- Brief overview of COPA
- Conclusion and future work

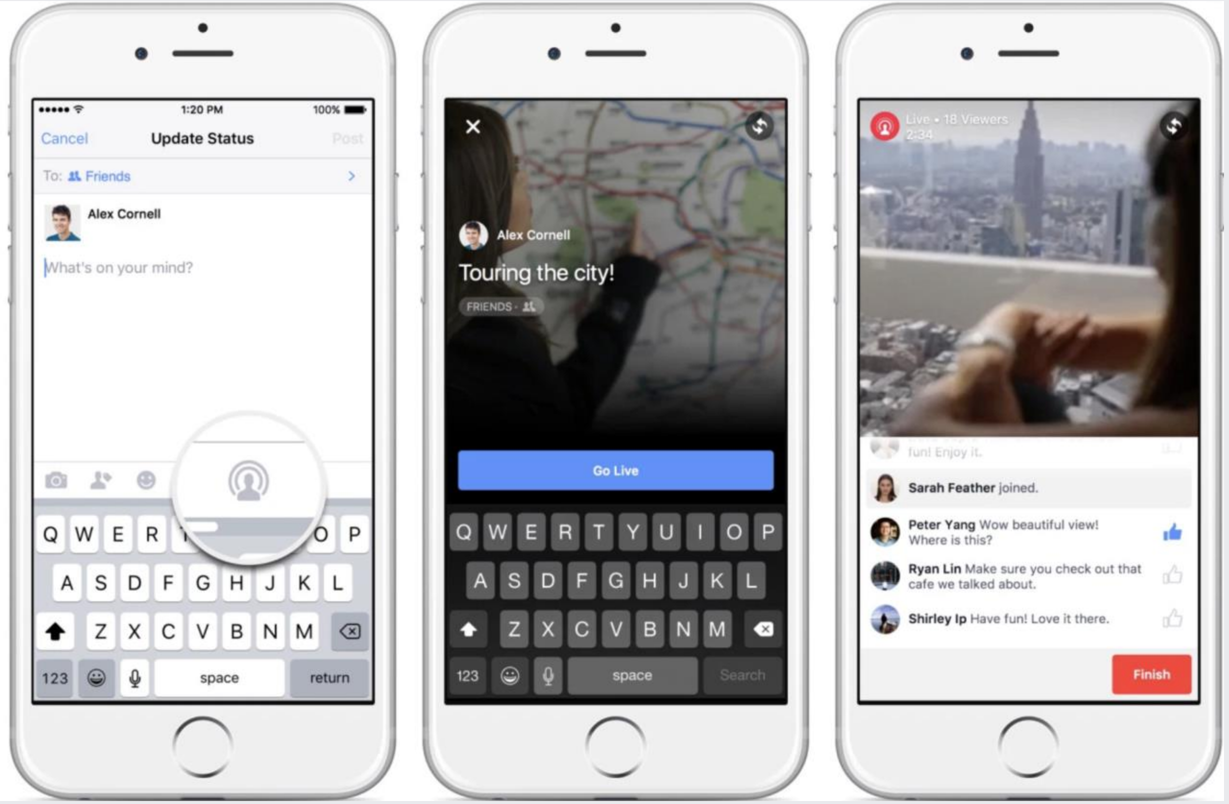
Motivation

- Difference video experiences require different latency vs quality trade off
    - Interactive applications require low end to end latencies.
    - Several other applications can tolerate latency but require high quality.
- Is it possible to have a single Congestion control algorithm compatible with all scenarios?
    - Dial for application to set the throughput vs delay trade-off
- COPA : Delay based congestion control, with a param delta to control delay sensitivity.

An interactive video experience

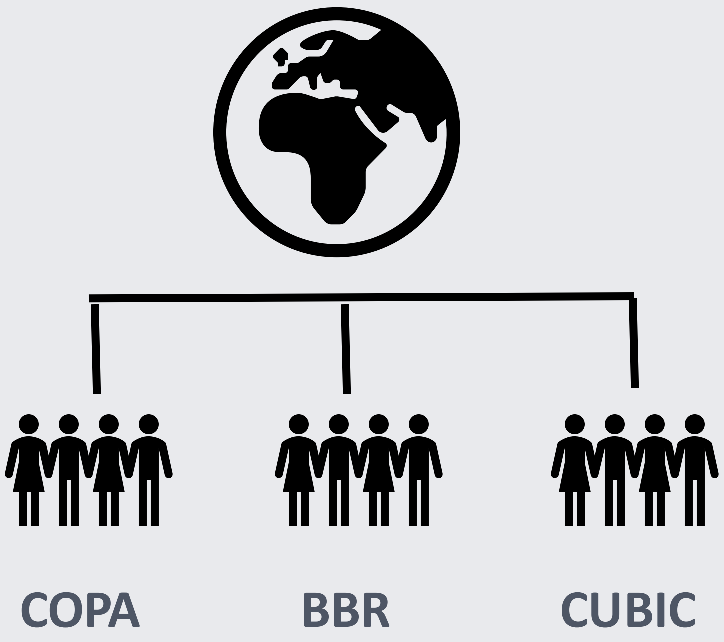
Experiment setup

- COPA vs BBR vs CUBIC
  - Tuned COPA to optimize for throughput at the expense of delay.
- Tested with FB Live stream application on mobile.
  - Long running flows, P50 duration is around 3 min.
- ABR to change bitrate in response to the network conditions.
- Congestion control algorithms implemented in the Facebook QUIC library.



Experiment setup

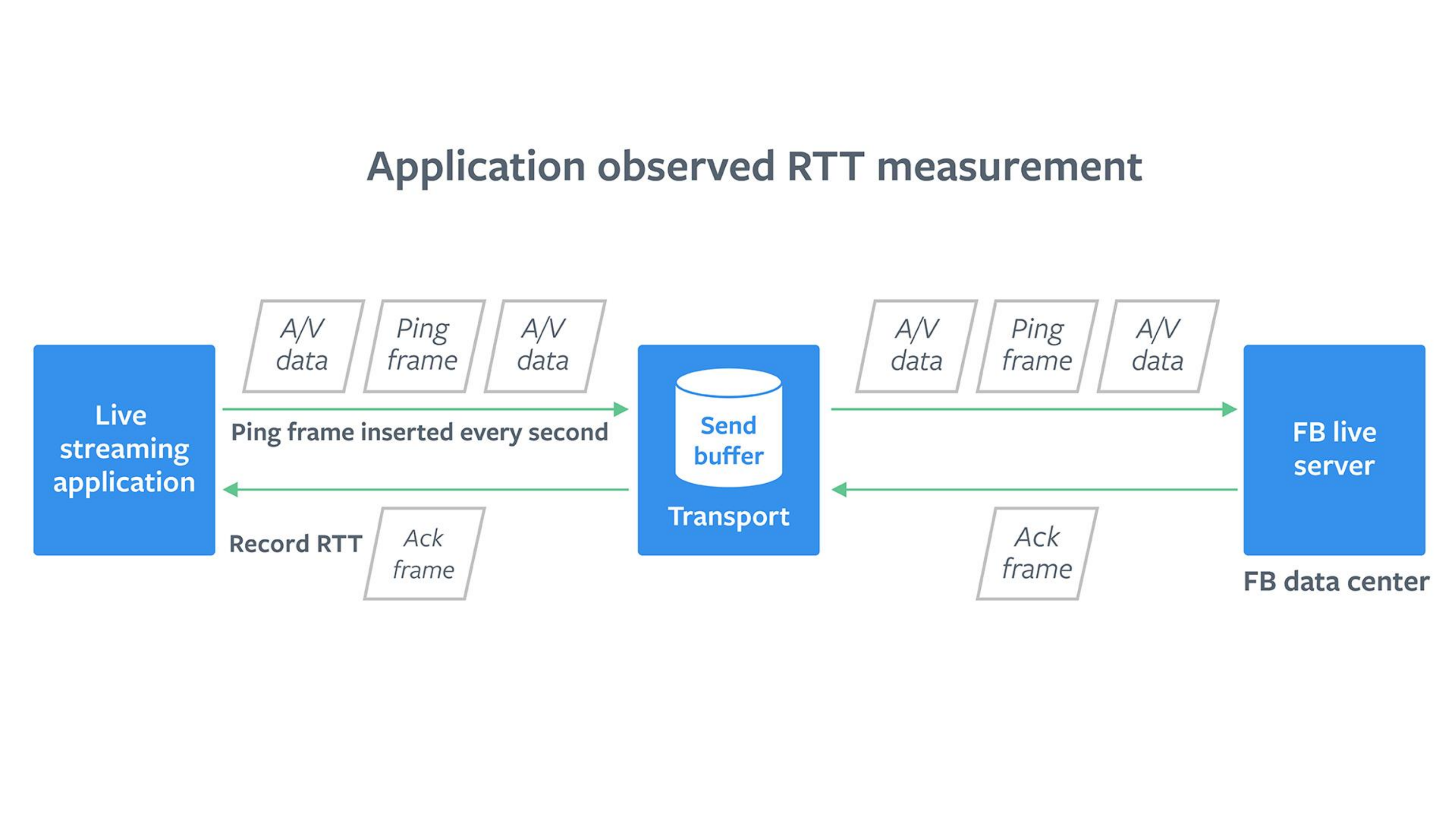
- Metrics collected from broadcasts from Android devices all over the world: US, Mexico, India, Vietnam, Indonesia, Thailand and so on.
- Used A/B testing framework to randomly divide users into 3 equal groups.
- Collected roughly 4 million samples for each group over 2 weeks.



Application metrics

Avg Goodput: Num of application bytes successfully sent / duration of broadcast.

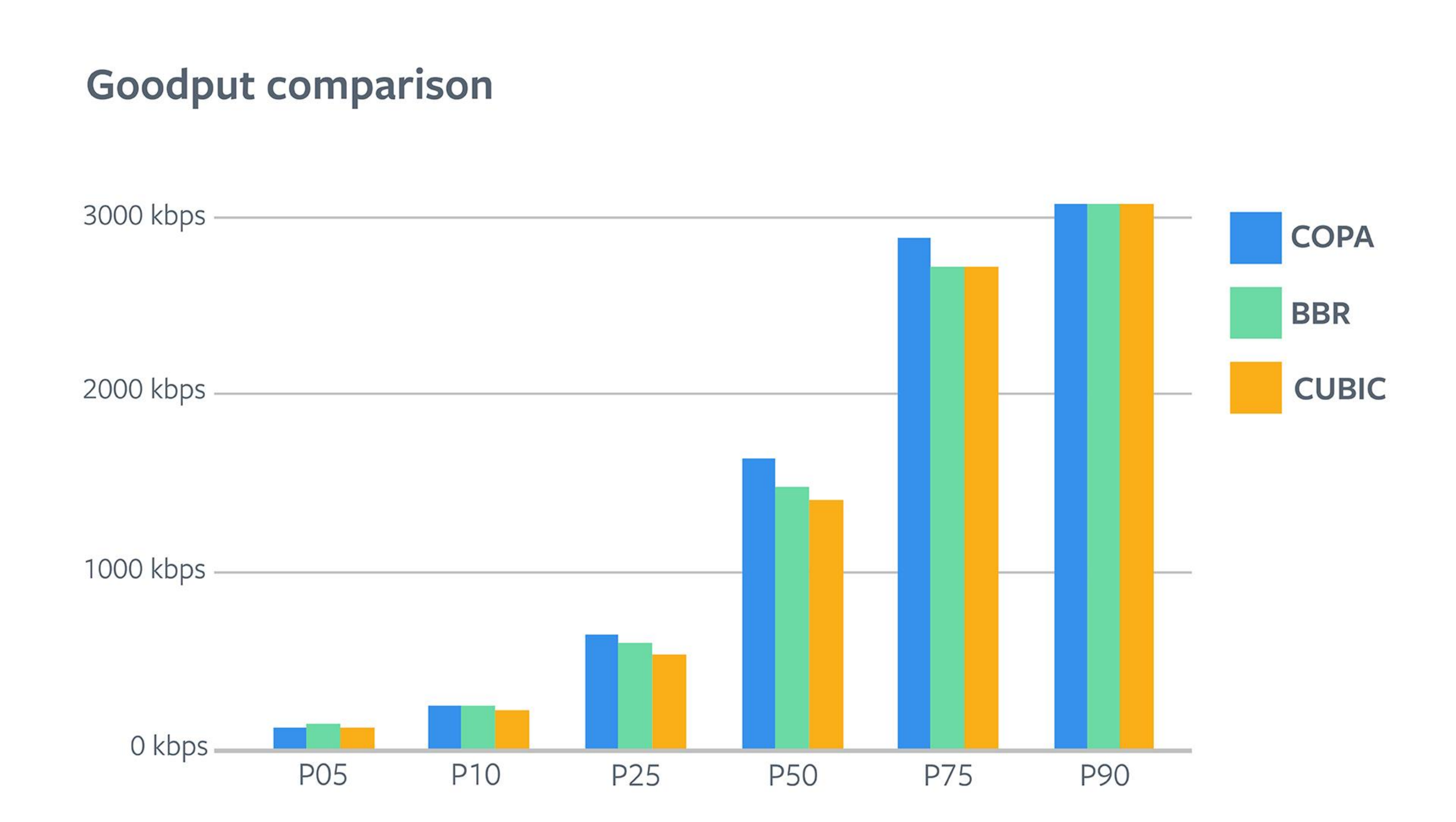
Avg Application Observed RTT: Avg of round-trip time taken by ping frames sent by application to FB Live server.



Results: Goodput

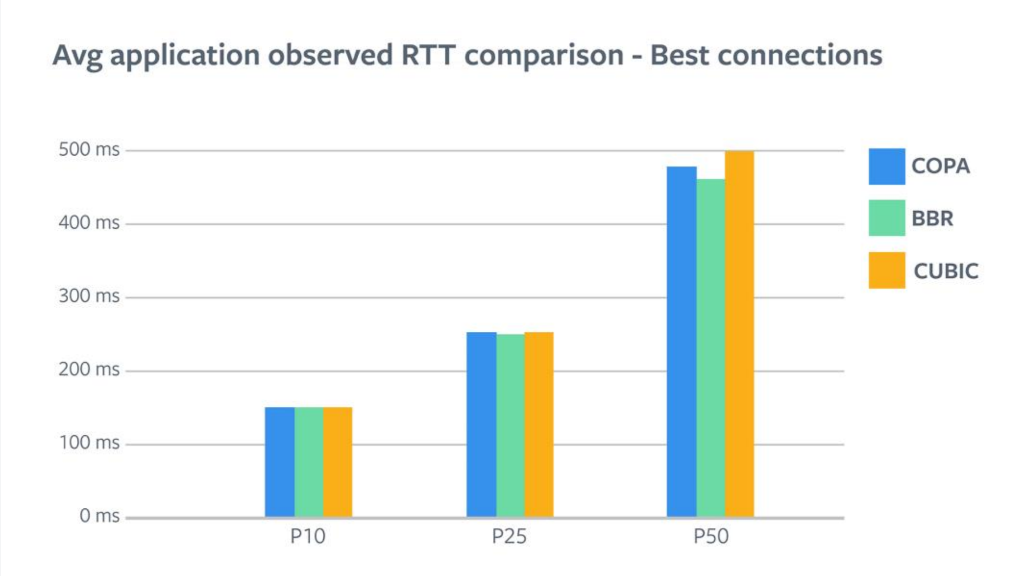
Avg Goodput: Num of application bytes successfully sent / duration of broadcast

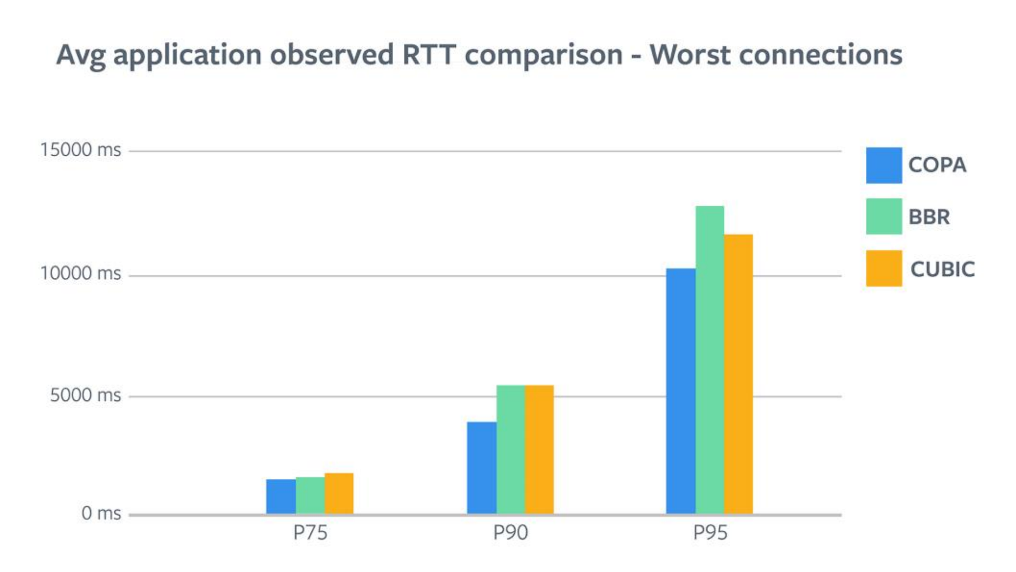
- P50: Copa +16%, BBR: +5%
- P10: Copa +6%, BBR: +4%
- Positive impact on top-line video watch time / engagement metrics



Results: Video ingest latency

Avg Application Observed RTT: Avg of round-trip time taken by ping frames sent by application to origin.

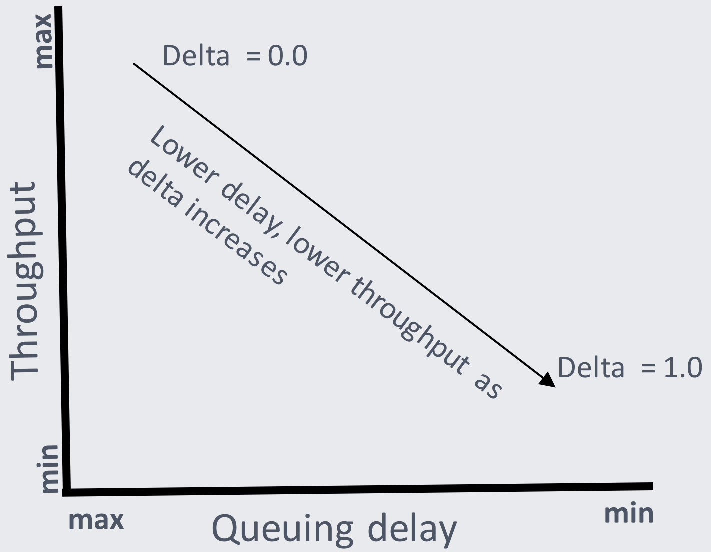


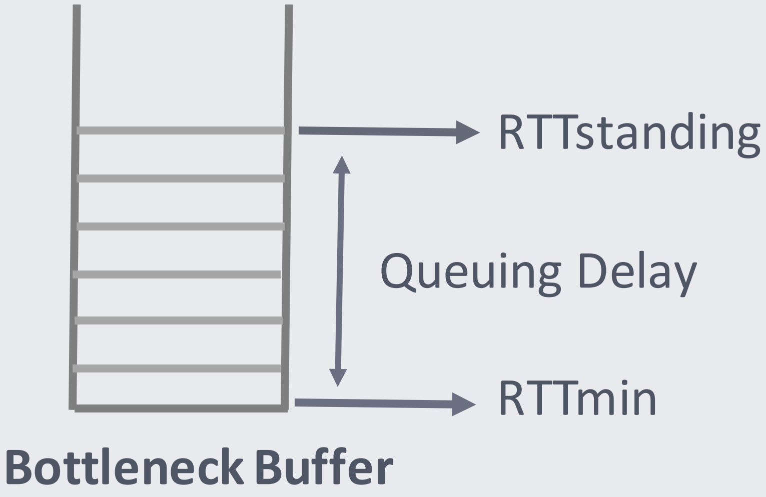


- BBR reduced App RTT most for P50 and below. P50 : BBR 8% , COPA 4%
- COPA reductions were highest for the worst connections. P90: BBR 0%, COPA 27%
- COPA impacted Application ABR to produce more bytes, but kept latencies low.

COPA

Tunable Delay based congestion control

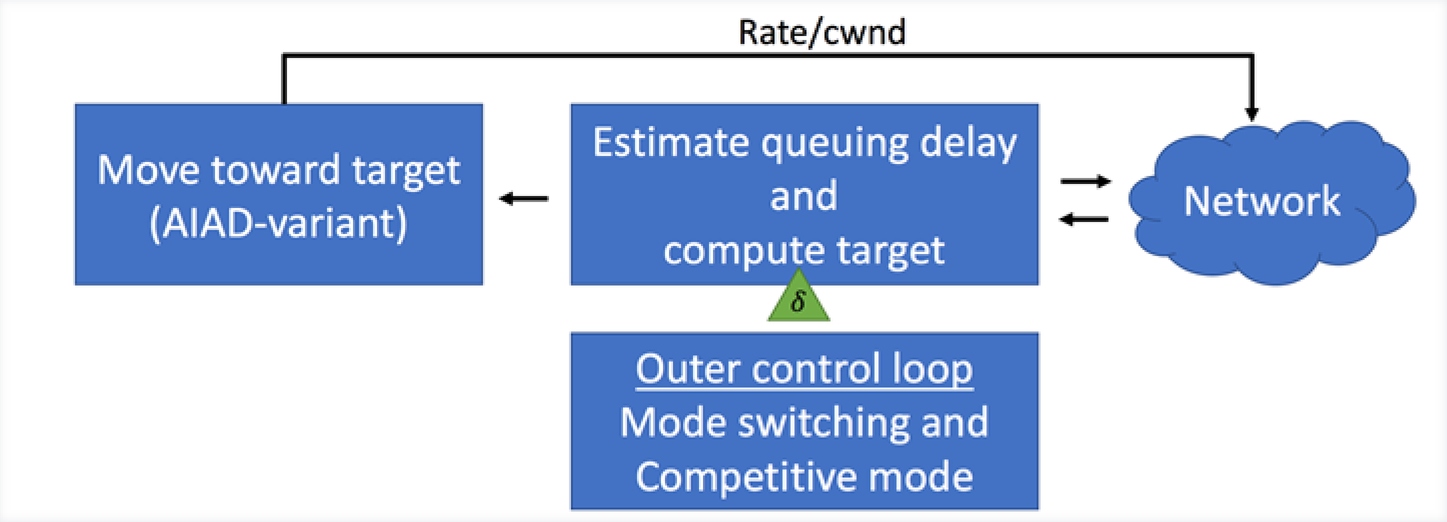




- **Delta param**: Determines sensitivity to delay.
- Uses RTT variation as a signal for congestion.
- **RTTmin** : min RTT over 10 seconds. Estimate of two way propagation delay for the network path.
- **RTTstanding** : min over srtt/2. Current Round trip time including queueing delays.
- **Queuing delay** = RTTstanding - RTTmin

COPA

Tunable Delay based congestion control



Once every ack

- Compute Queueing delay Dq = RTTstanding - RTTmin
- Calculate Target rate = 1 / (delta * Dq)
- Adjust cwnd = cwnd +/- (v / (delta*cwnd))
- Change in one RTT = (v / delta) packets
- V = velocity parameter, 1 by default.

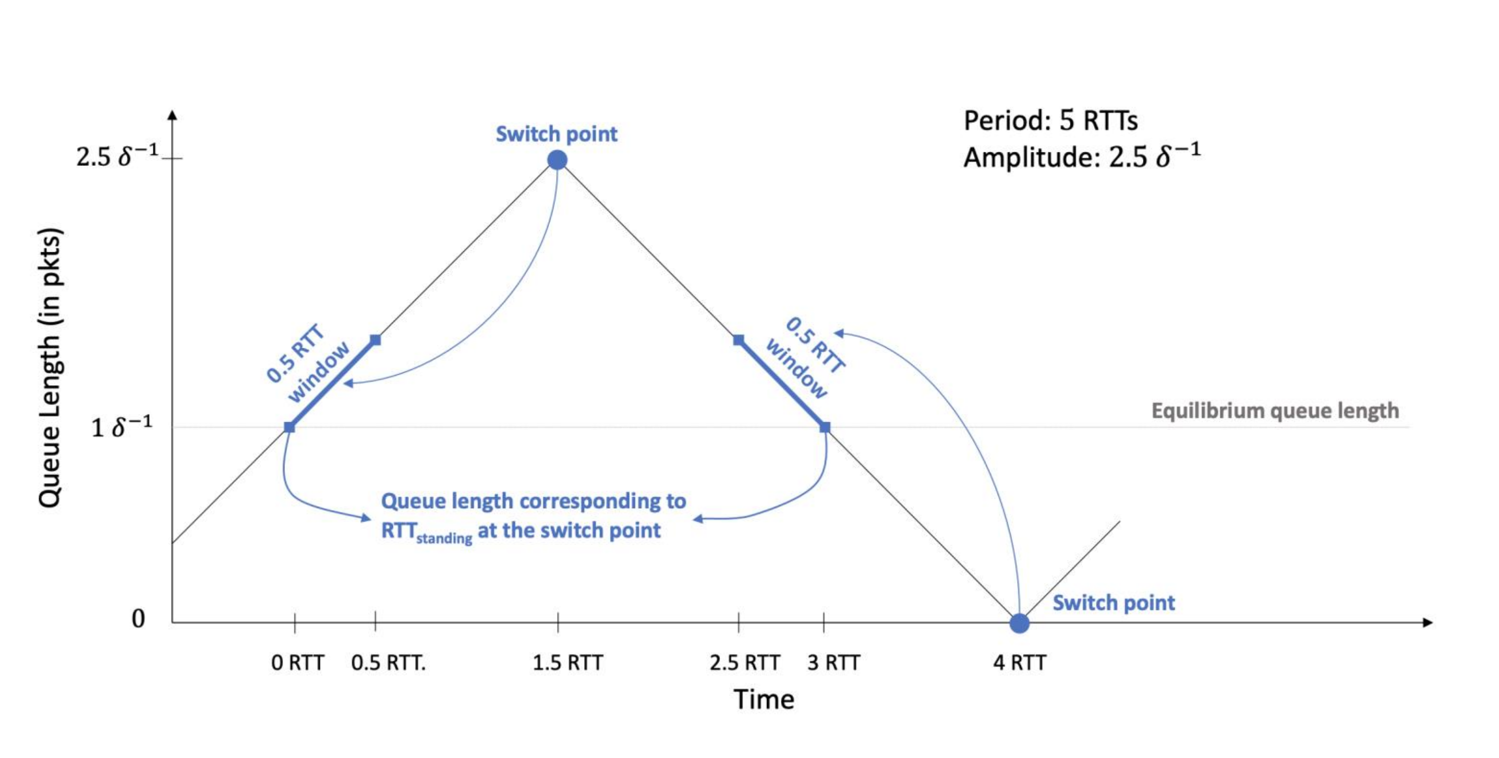
Competitive mode

- Normally delay based CCA lose to buffer filling flows.
- Competitive mode detects presence of buffer filling flows and adjusts delta to be more aggressive.
- We tested without competitive mode

COPA

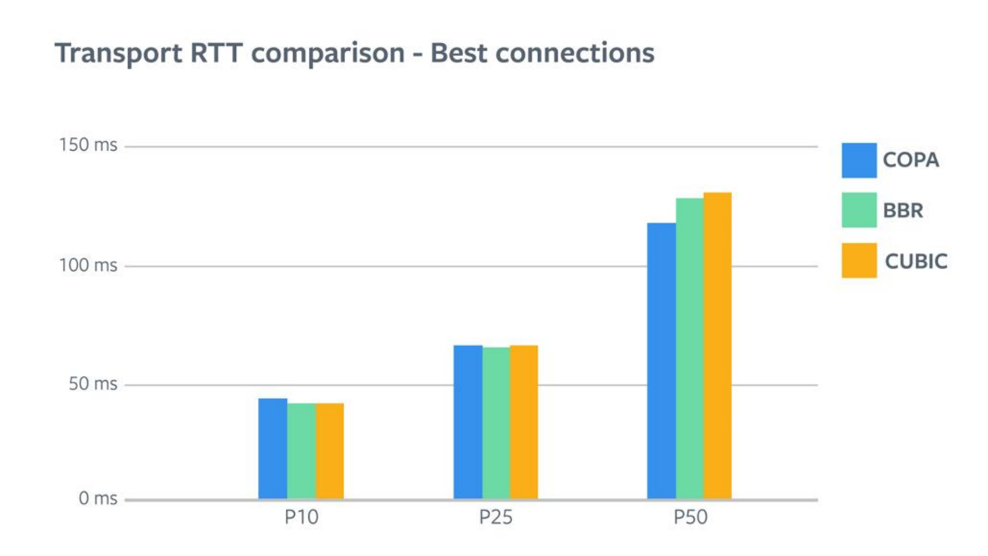
Steady State Dynamics

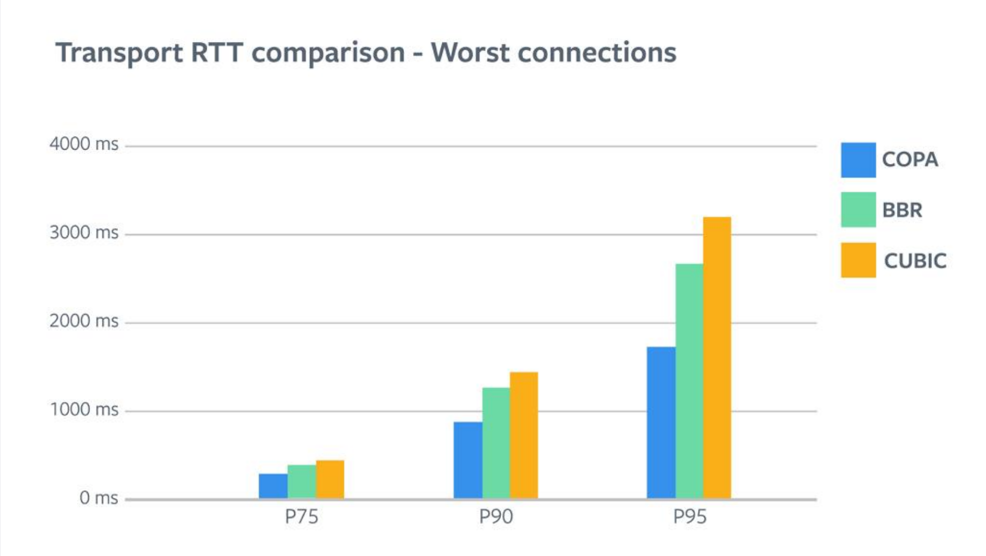
- Bottleneck queue varies from 0 to 2.5 / delta.
- Cycle repeats itself every 5 RTT
- Steady state queue length = 1/delta.
    - 25 packets at delta = 0.04



Avg Transport RTT comparison

Avg Transport RTT: Avg of QUIC RTT measurements over the duration of broadcast



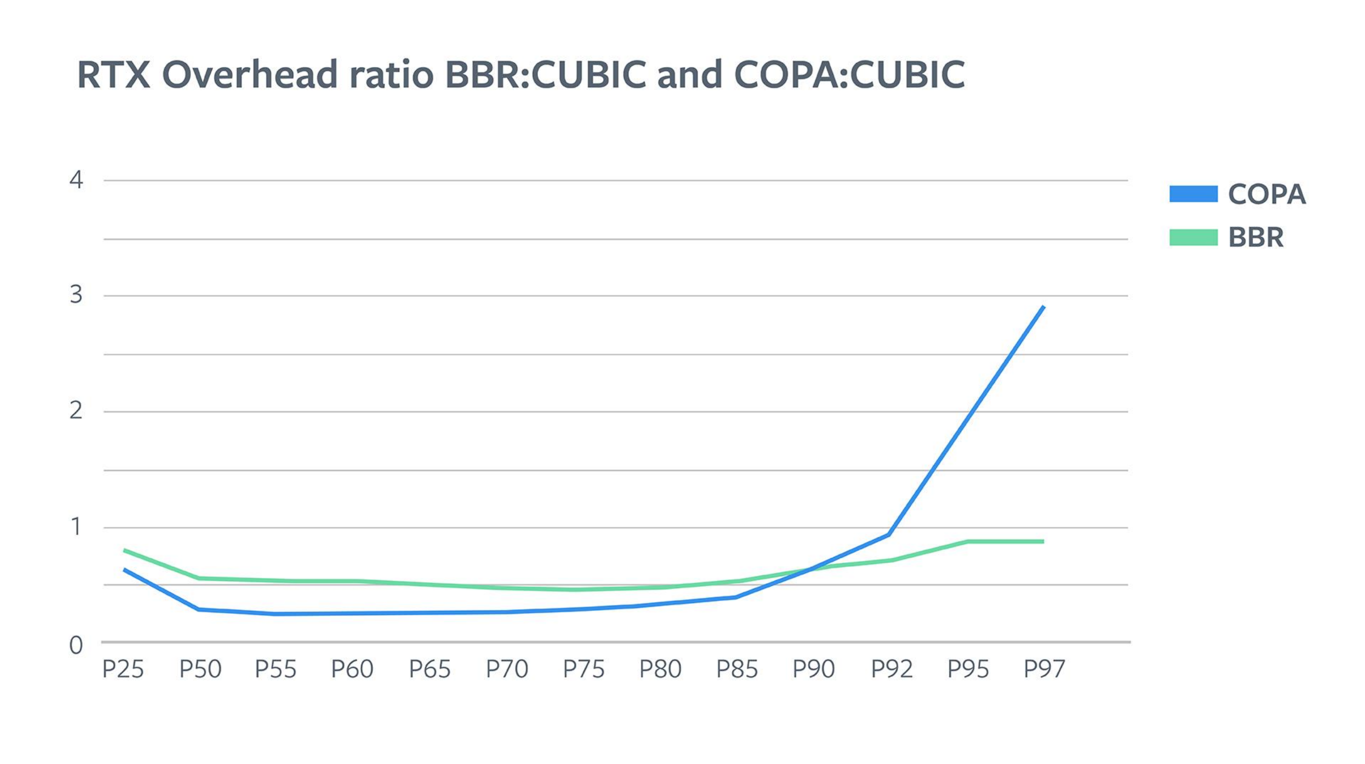


- Trend was similar to App RTT measurements.
- BBR reduced transport RTT most for below P50. But the difference is smaller.
- COPA reductions were highest for the worst connections. P90: BBR 8.8%, COPA 38%.
- Shows that COPA not only reduced App RTT, but also RTT for the network.

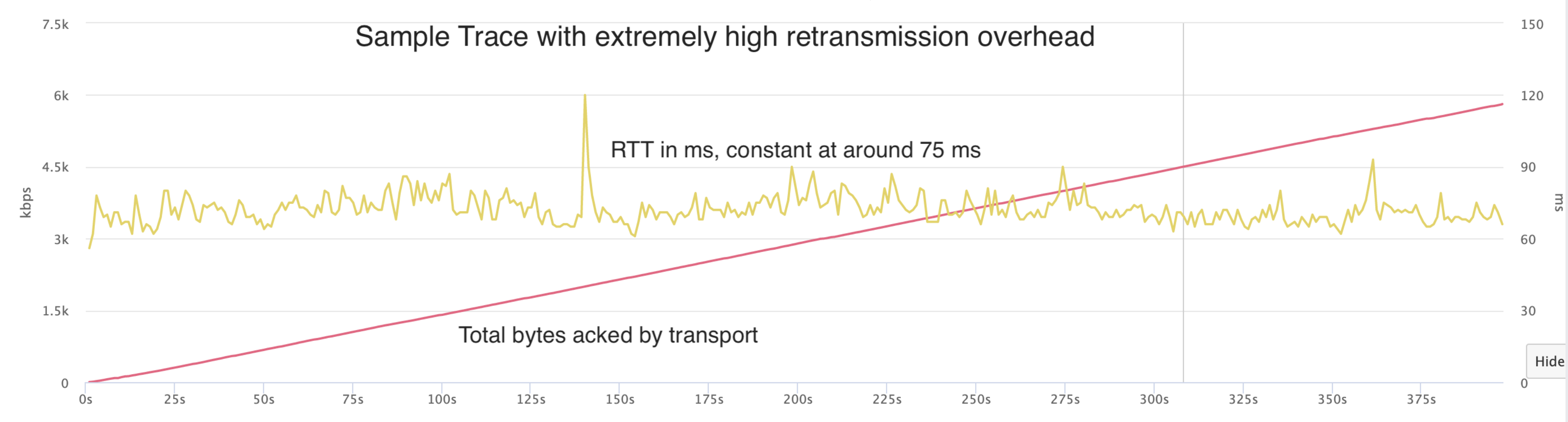
RTX Overhead

RTX Overhead: Total bytes re-transmitted by transport / total bytes acked by transport, during the broadcast.

- COPA RTX overhead is lower for 90% of broadcasts, indicating where some of the application wins might have come from.
- Grows rapidly for the last 10%. But surprising that we did not see a corresponding degradation in application metrics

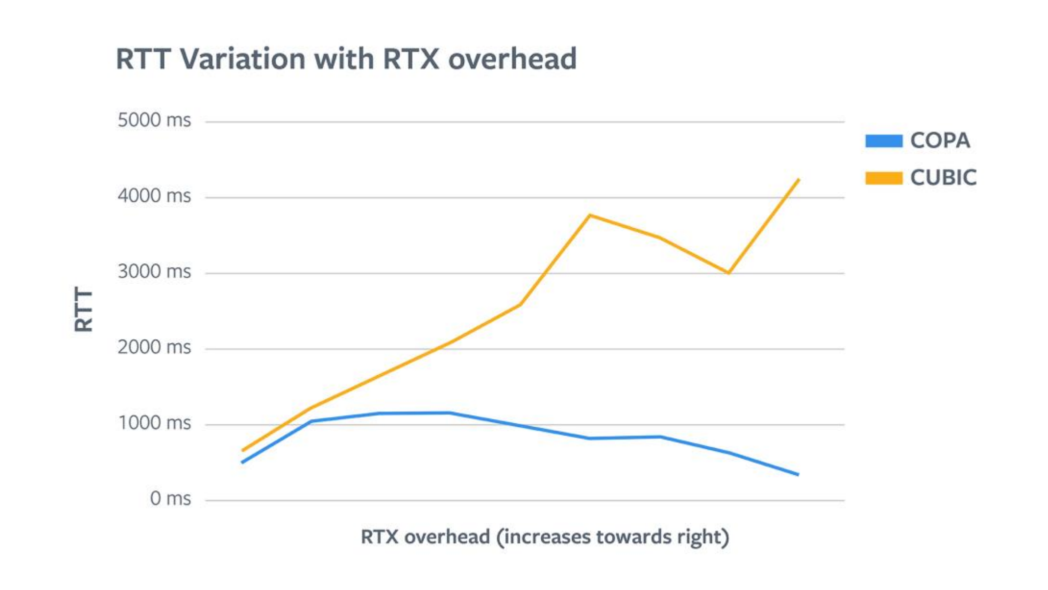


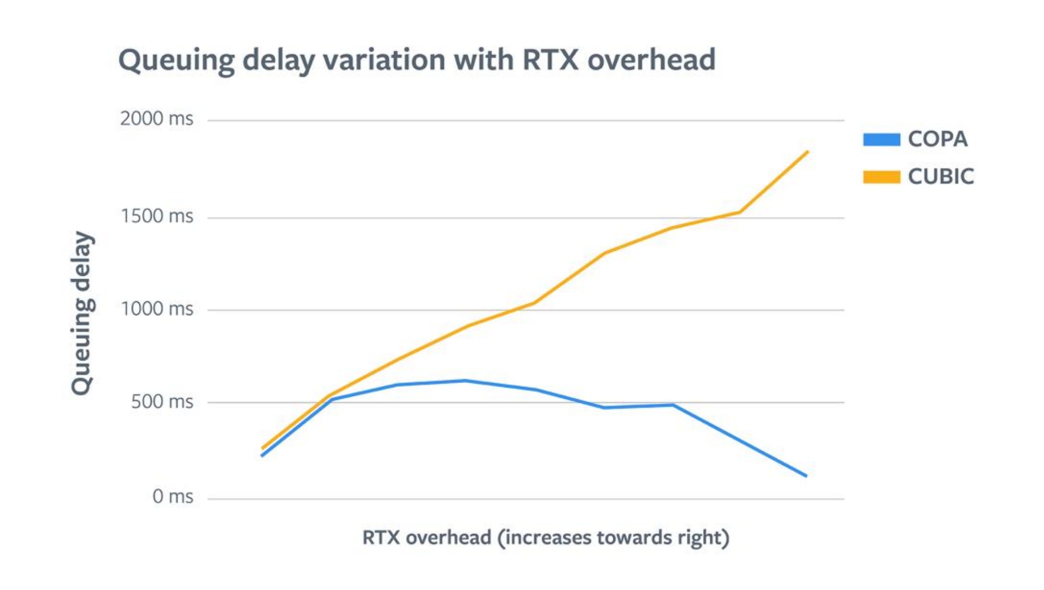
Debugging the tail cases



- Sampling individual traces with a high loss rate showed characteristics like network policed flows.
    - Consistent throughput
    - Low RTT and low RTT variation
- RTX overhead varied greatly by ASN

RTX Overhead and RTT/Queuing delay correlation





- For CUBIC, RTX overhead and RTT/Queuing delays are correlated.
- For COPA, worst RTX overhead broadcasts have low RTT and Queuing delays

RTX Overhead investigation summary

- Strong indication that High RTX for last 10% of users in COPA are due to Network policing.
  - It's possible that there are other reasons too, like short buffers.
- Improvements needed in COPA to reduce loss rates.
  - Competitive mode could help
  - Add a heuristic to change cwnd based on loss, e.g. multiplicative decrease based on target loss.
  - Explicit network policer detection similar to BBR v1

Conclusion and Future work

- Aggregated results show that COPA provided better quality and lower latencies in our tests for mobile broadcasts as compared to CUBIC and BBR with QUIC.
    - With BBR undergoing several changes, results may differ in the future.
- Better understanding for the reasons behind the improvements.
    - Lower RTX overhead?
    - Better target rate estimation and faster convergence?
    - Something else?
- Test on the other extreme of latency vs quality tradeoff: test for ultra-low latency case and compare with alternatives like GCC (Google Congestion Control).
- Test for other use cases such as video playback traffic.

Thank you

- COPA paper: https://www.usenix.org/conference/nsdi18/presentation/arun
- Source code https://github.com/facebookincubator/mvfst/tree/master/quic/congestion_control
- Post with more details https://engineering.fb.com/video-engineering/copa/
- Email: ngarg@fb.com

Thanks to Yang Chi, Subodh Iyenger, Roberto Peon, Kirill Pugin, Marek Latuskiewicz, Udeepta Bordoloi and several others from Video Infra & Traffic Protocols team at Facebook.