

Multi-path QUIC Extension and Experiments

draft-liu-multipath-quic

IETF 111 - 2021.7

Yanmei Liu - miaoji.lym@alibaba-inc.com

Yunfei Ma - yunfei.ma@alibaba-inc.com

Basic design for multi-path quic

- A minimally-scoped extension based on QUICv1.
- Bidirectional path
- Keep the packet header formats unchanged and use Connection IDs
- Congestion Control, RTT measurements and PMTU discovery are per-path
- Only three extension frames.

More details: <https://datatracker.ietf.org/doc/draft-liu-multipath-quic/>

Experimental Motivation & Methods

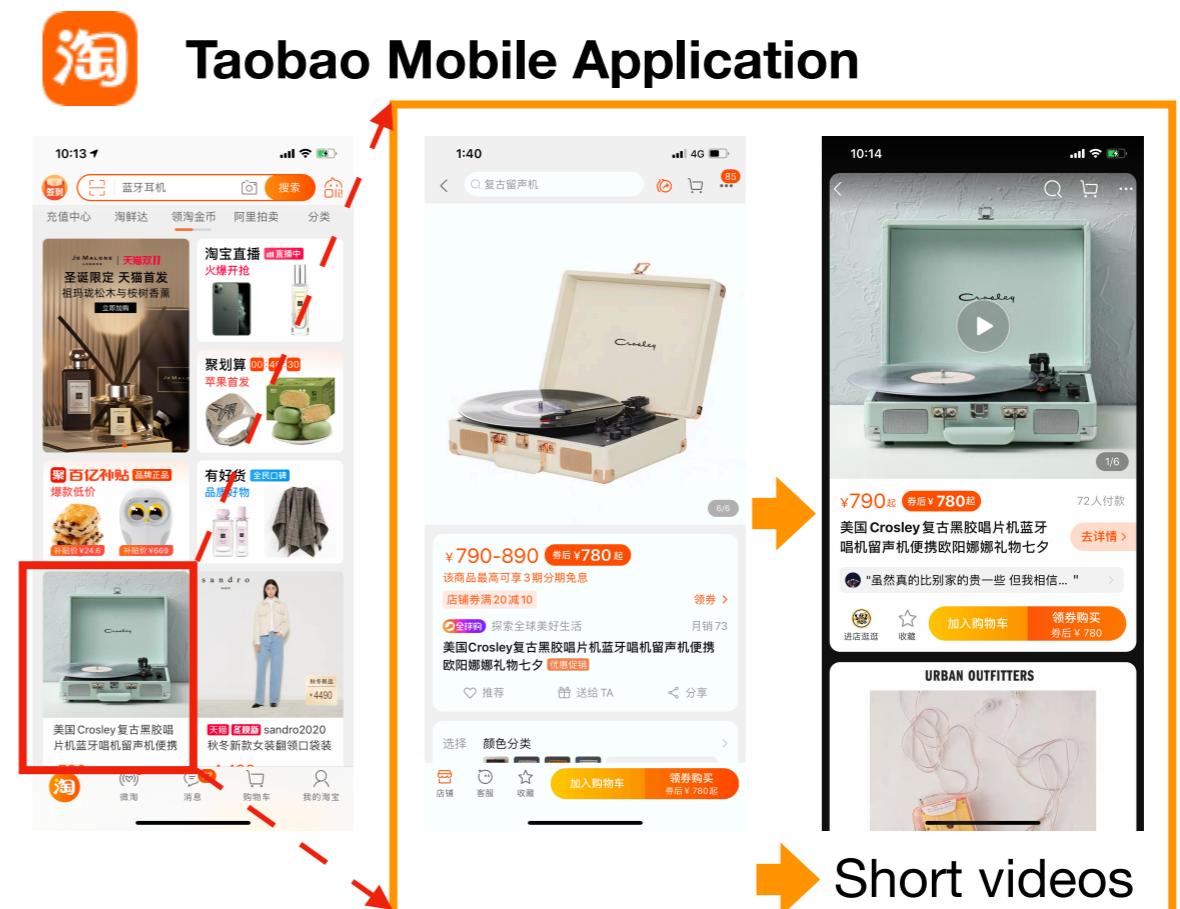
Motivation

- To verify if we can obtain real performance **gain** with the use of multi-path QUIC in short-form video streaming
- To better understand the challenges of using multi-paths for video applications.

Methods

- A/B Test with 100K participants who upgraded to test versions.
 - Two contrast groups running in parallel.
 - Multi-path users are zero-rated.

- Client-side: Taobao Mobile Android app with single/multi-path QUIC
- Server-side: Edge server for video service
- Both client & server use XQUIC as protocol implementation



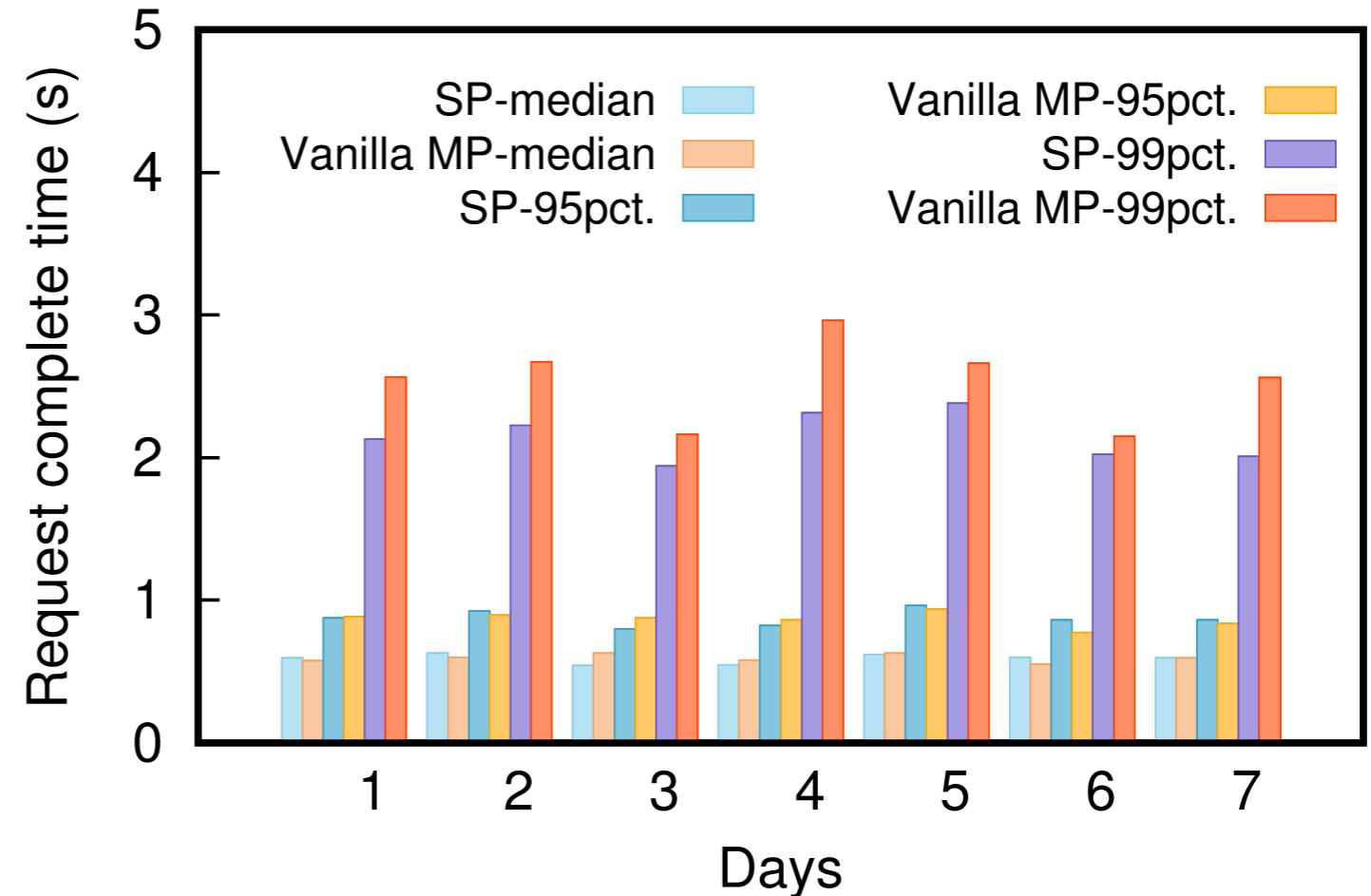
Protocol used in experiments

<https://datatracker.ietf.org/doc/draft-liu-multipath-quic/>

A/B test results of vanilla multi-path scheduling

Min-RTT Scheduler

Downloading 1MB video chunk



A sample from the experiment

	avg	p50	p90	p95	p99
sp-close_delay (us)	389668	294151	562843	825402	2027727
sp-srtt (p1) (us)	41133	30520	69598	96128	200317
mp-split-close_delay (us)	453946	339488	671178	902282	2009848
mp-split-srtt (p1) (us)	46377	32012	79025	112028	255380
mp-split-srtt (p2) (us)	133915	73228	221850	342091	922026

Multi-path head-of-line blocking

If one of the paths is not working, packets sent to that path will be lost. It will take time to correct these losses.
<https://huitema.wordpress.com/2021/01/26/implementing-multipath-in-quic/>

Use re-injection to overcome MP-HoL

Multi-path SHOULD achieve no worse performance than single path

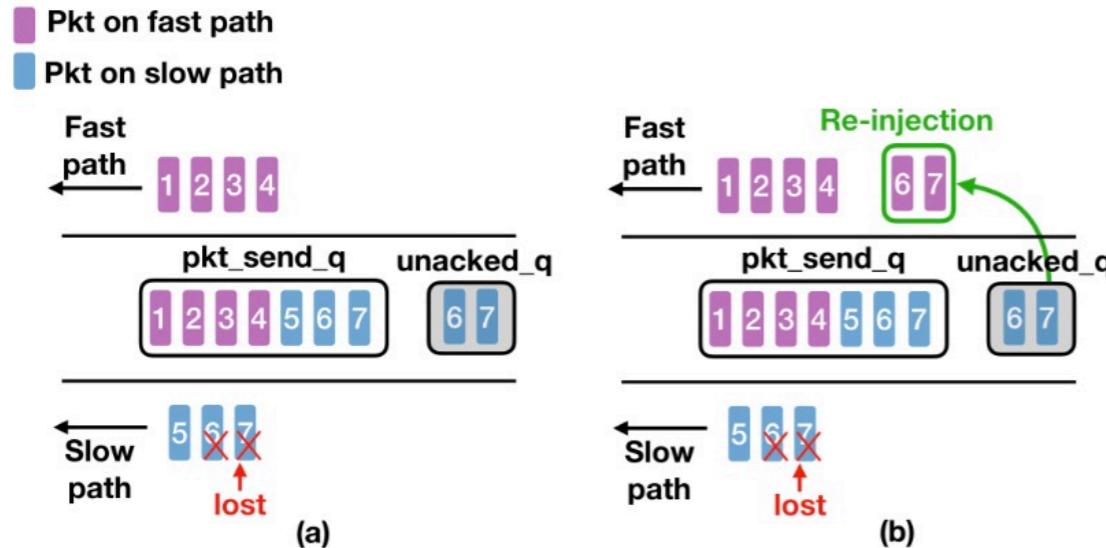


Figure 3: Use re-injection to overcome multi-path HoL blocking
 (a) Without re-injection, packets lost on the slow path would block the fast path. (b) With re-injection, lost packets on the slow path can be quickly recovered from the fast path.

Get better quality of service now, but more than 15% additional traffic costs?

- Still too expensive for users and video services

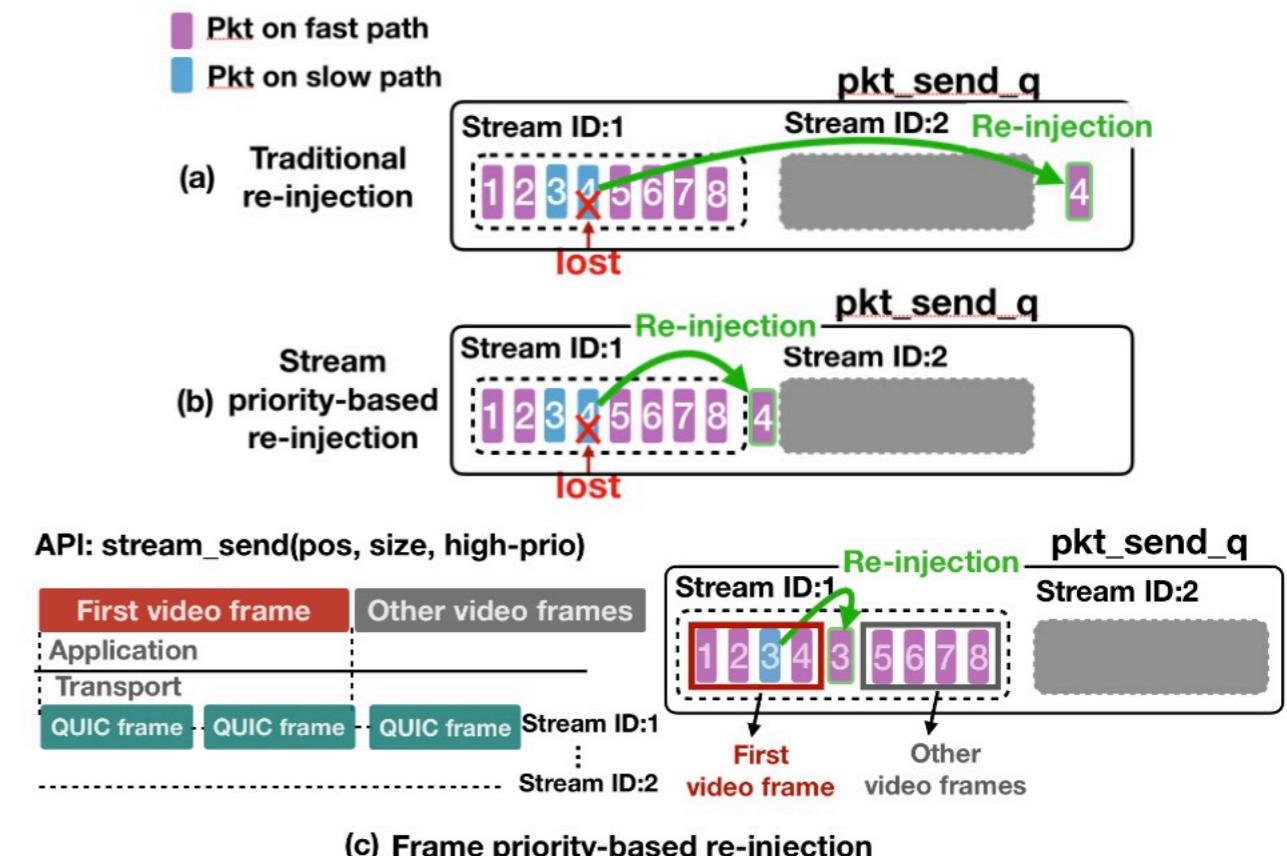
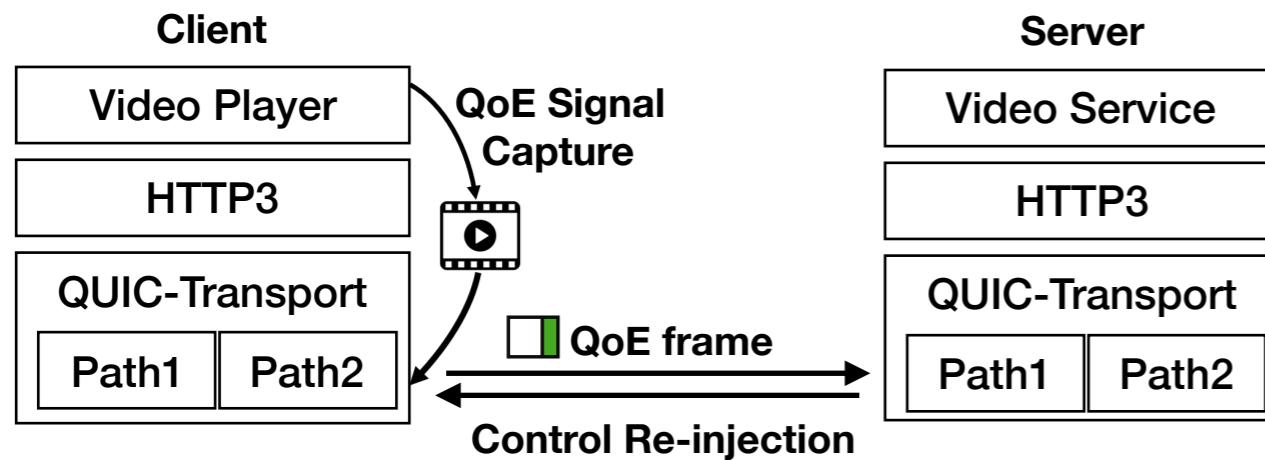
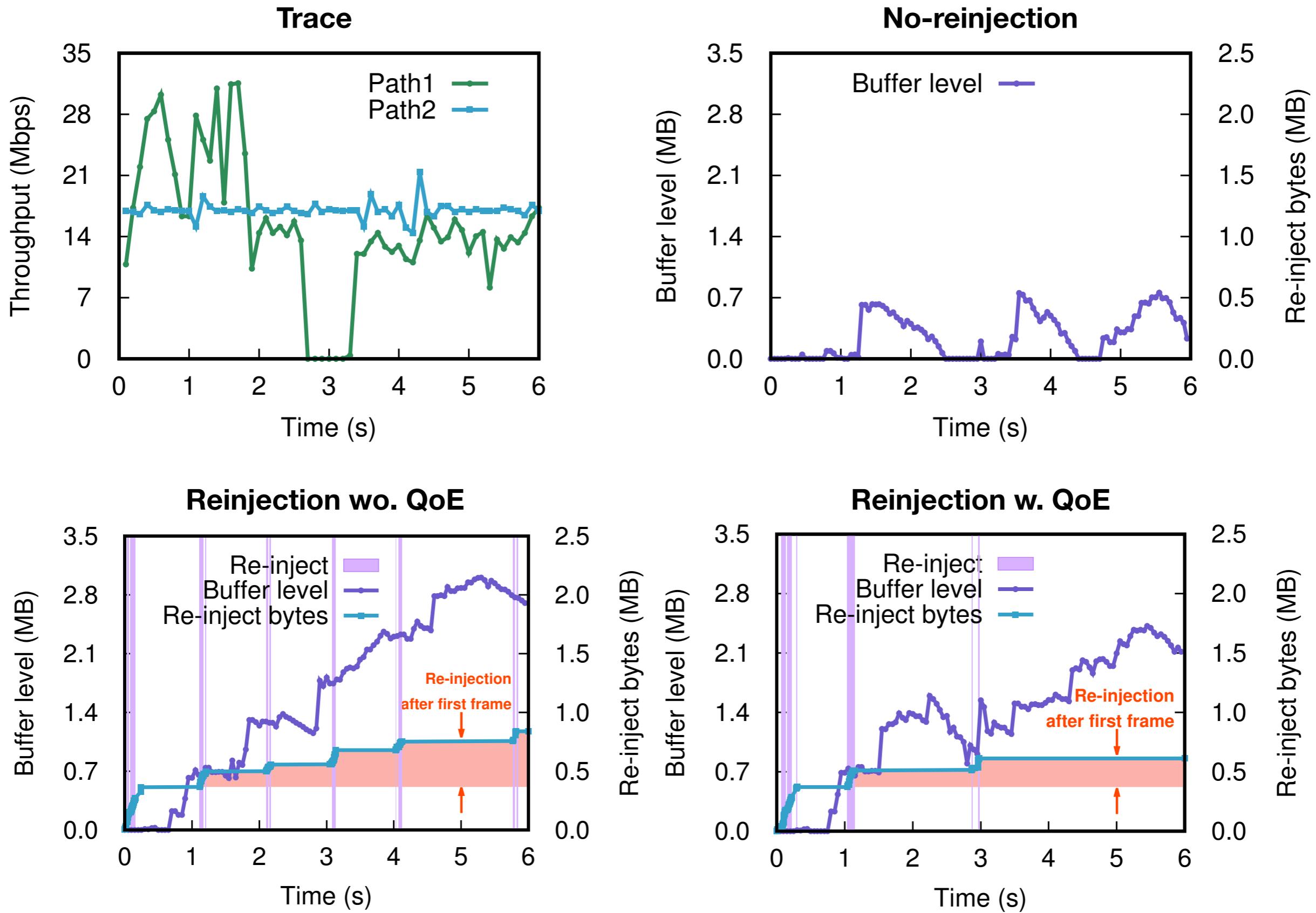


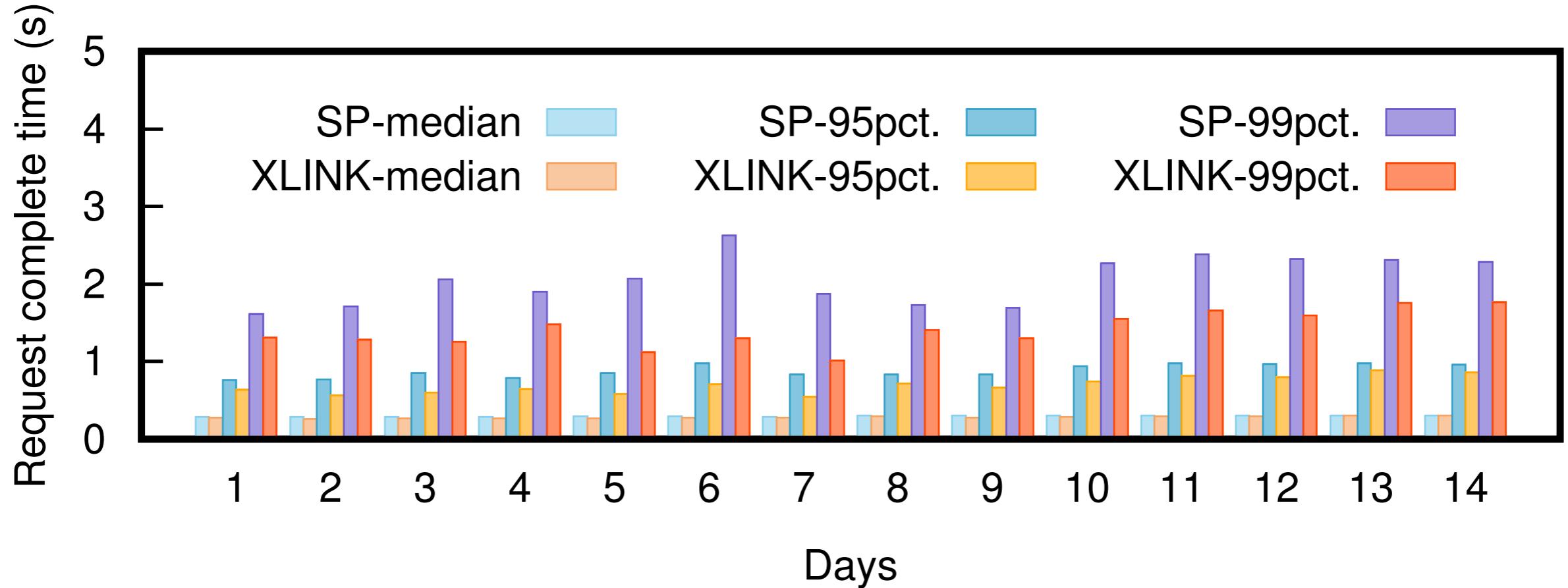
Figure 4: Different modes of re-injection: (a) Traditional (Appending) mode, (b) stream priority-based mode to address stream blocking and (c) video-frame priority-based mode to address video frame blocking.

- Use QoE feedback to control the aggressiveness of re-injection
- For VoD: QoE feedback contains info related to client's video buffer level

Scheduler with QoE feedbacks



QoE-driven scheduling A/B test



Improvement median: 2.3%-8.9%

Improvement 95th: 9.4%-34%

Improvement 99th: 19%-50%

Re-injection traffic overhead reduce from 15% to 2%

Summary

- Obtaining good performance with multi-path QUIC is not straightforward
- Need to overcome MP-HoL blocking
- Leverage QUIC to collaborate with application
- Use QoE feedback (balance cost and performance)
 - It's optional, don't be worry about binding application layer and scheduler on transport layer
 - QoE feedback and Scheduler algorithm may depend on application scenarios
 - QoE_Control_Signal frame is used for experiments
 - Maybe need an additional transport parameter for algorithm and QoE feedback format negotiation

For more details ...

<XLINK: QoE-Driven Multi-Path QUIC Transport in Large-scale Video Services>
SIGCOMM 2021 Conference (SIGCOMM '21), August 23–28, 2021

Multi-path Transport Questions for Discussion

Does multipath QUIC need additional signaling to mitigate path HOL-blocking?

- As it's hard to predict future network conditions, use timely QoE would help multi-path scheduling and re-injection get better results

What is very much in scope is what mechanisms does a multi-path quic design need to be deployable at scale?

- Simple and clear extension, easy to get deployed
- obtaining real gain on QoE