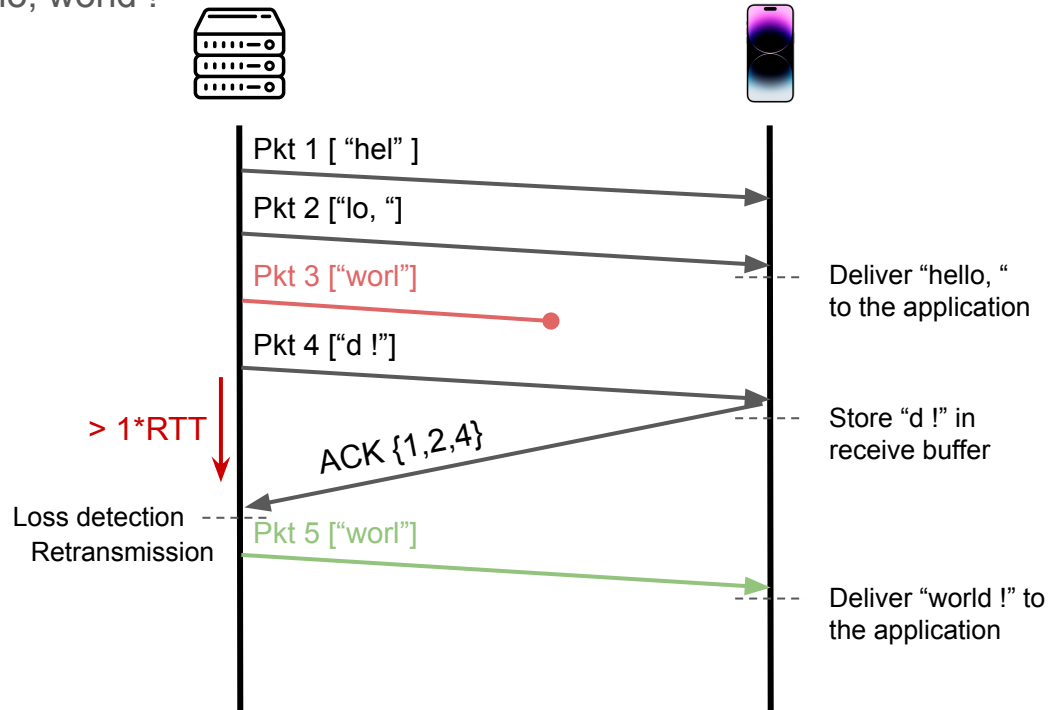# QUIC+FEC

Some results for low latency video streaming

François Michel
Olivier Bonaventure
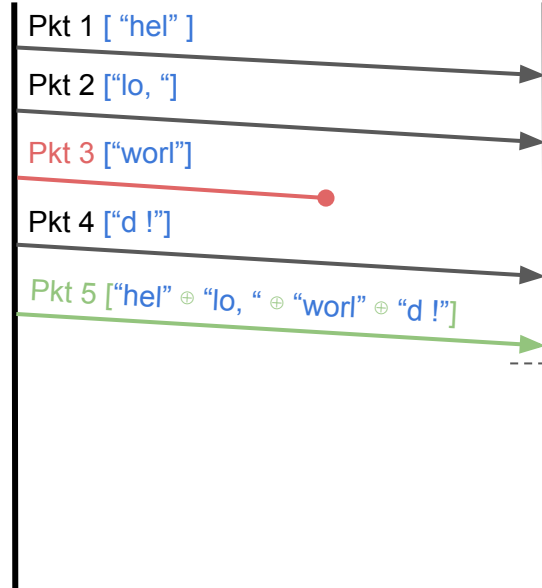
UCLouvain

Loss recovery in classical transport protocols (SR-ARQ)

Sending "hello, world !"

Pkt 1 ["hel"]

Pkt 2 ["lo, "]

Pkt 3 ["worl"]

Deliver "hello, "
to the application

Pkt 4 ["d !"]

> 1*RTT

ACK {1,2,4}

Store "d !" in
receive buffer

Loss detection
Retransmission

Pkt 5 ["worl"]

Deliver "world !" to
the application

2

Forward Erasure Correction in the transport

Sending "hello, world !"

Pkt 1 ["hel"]

Pkt 2 ["lo, "]

Pkt 3 ["worl"]

Pkt 4 ["d !"]

Pkt 5 ["hel" ⊕ "lo, " ⊕ "worl" ⊕ "d !"]

Pkt 1, 2, 3, 4 contain *source symbols*
Pkt 5 contains a *repair symbol*

Reconstruct "worl"
Deliver "hello, world !"
to the application

draft-michel-quic-fec-01

Forward Erasure Correction for QUIC loss recovery

Abstract

This documents lays down the QUIC protocol design considerations needed for QUIC to apply Forward Erasure Correction on the data sent through the network.

QUIRL: implementing draft-michel-quic-fec-01

Based on Cloudflare's `quiche` implementation.

- `quiche` is a production-ready implementation
- it is deployed on Cloudflare's edge servers
- used by the DNS resolver on recent Android versions
- can be integrated with `curl` for HTTP/3 queries

Using QUIRL for FFmpeg/GStreamer

- Every **RTP packet** is placed into a dedicated **QUIC stream**
  - large RTP packets cannot fit in DATAGRAM frames
- Repair symbols are sent regularly to protect one or more video frames
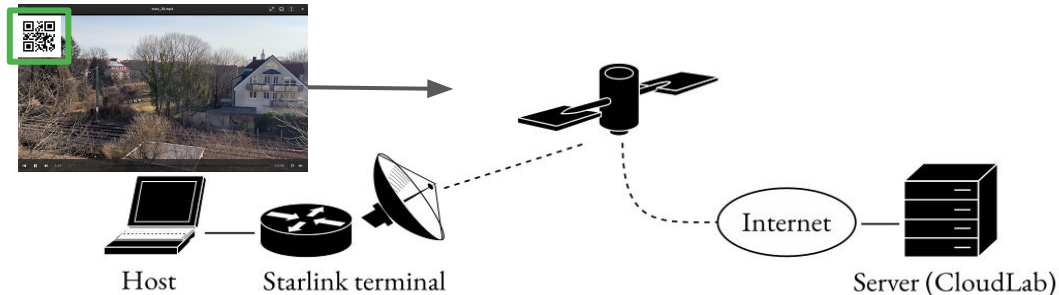- We want to minimize frames **lateness** to improve video fidelity (SSIM)
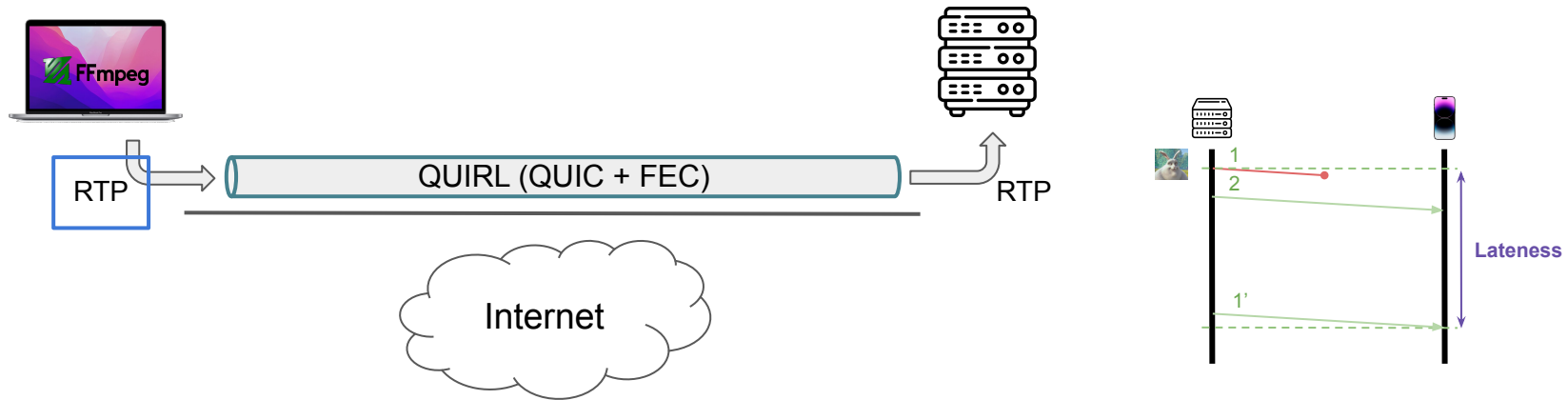
Replaying drone videos over Starlink

1000 experiments performed from a laptop in Belgium to a Cloudlab server (US)
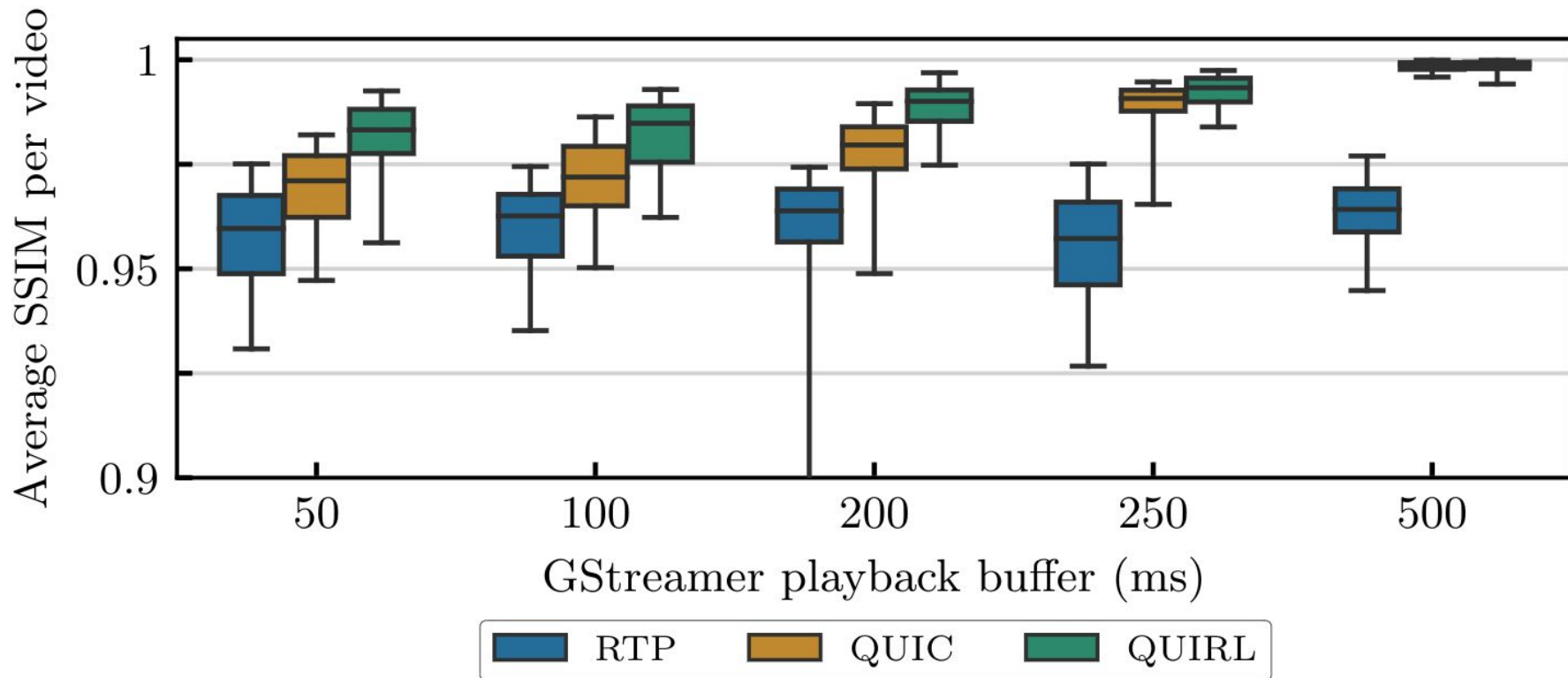
- Using 5 different playback buffer values

Real-time replay of drone videos from Baltaci *et al* (IMC '22). [1]

Used to match frames between original and playback

[1] Baltaci, Aygün, et al. "Analyzing real-time video delivery over cellular networks for remote piloting aerial vehicles." *IMC* 2022.
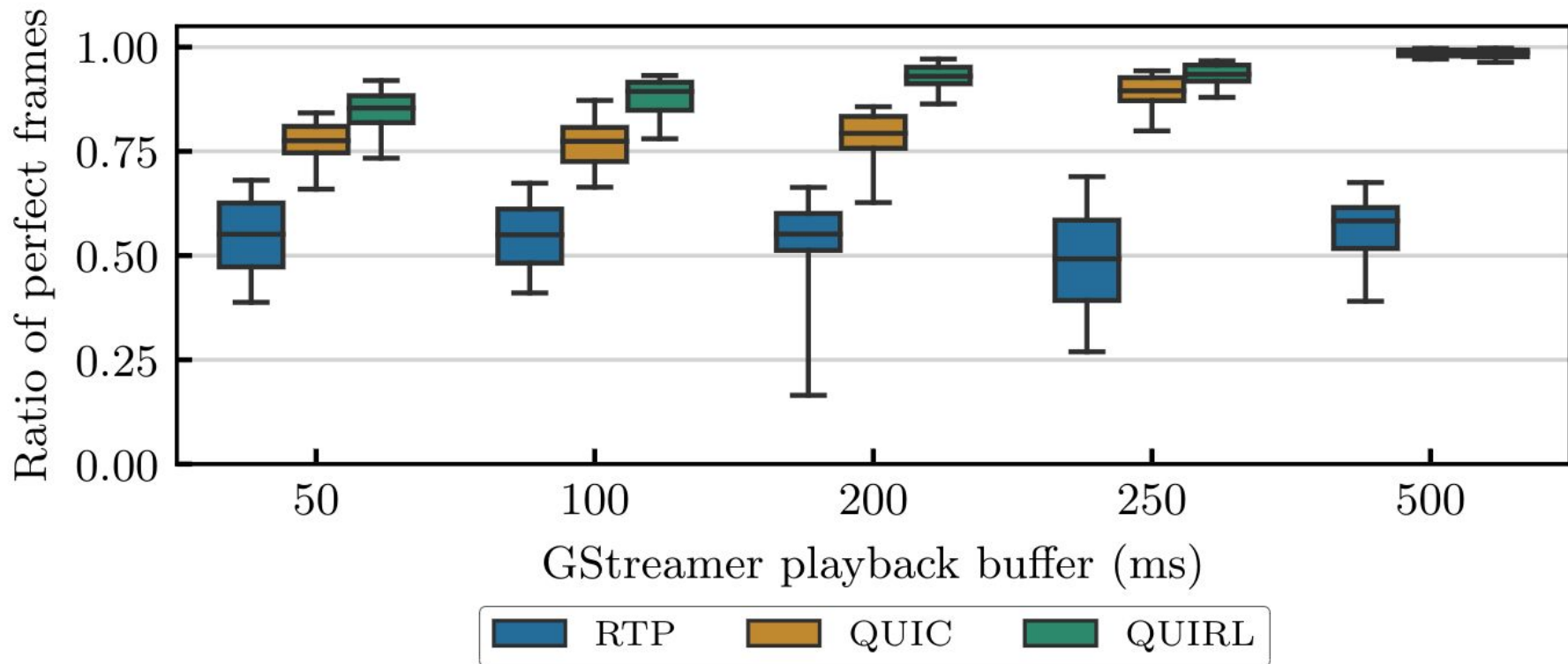
Average SSIM per video over Starlink

Ratio of perfect frames (SSIM=1) over Starlink

Summary

All our work and code will soon be open source. More details can be found in:

- My thesis: https://ncs.uclouvain.be/phd/2023/10/12/michel-phd.html
- The QUIRL paper (soon)

If your use-cases may benefit from QUIC-FEC, here's how we could start :

- Discuss on slack and the mailing list
- Send us an e-mail to collaborate: francois.michel@uclouvain.be
- Implementing draft-michel-quic-fec
- Please, do it with us, not on your own ! :-)