QUIC+FEC

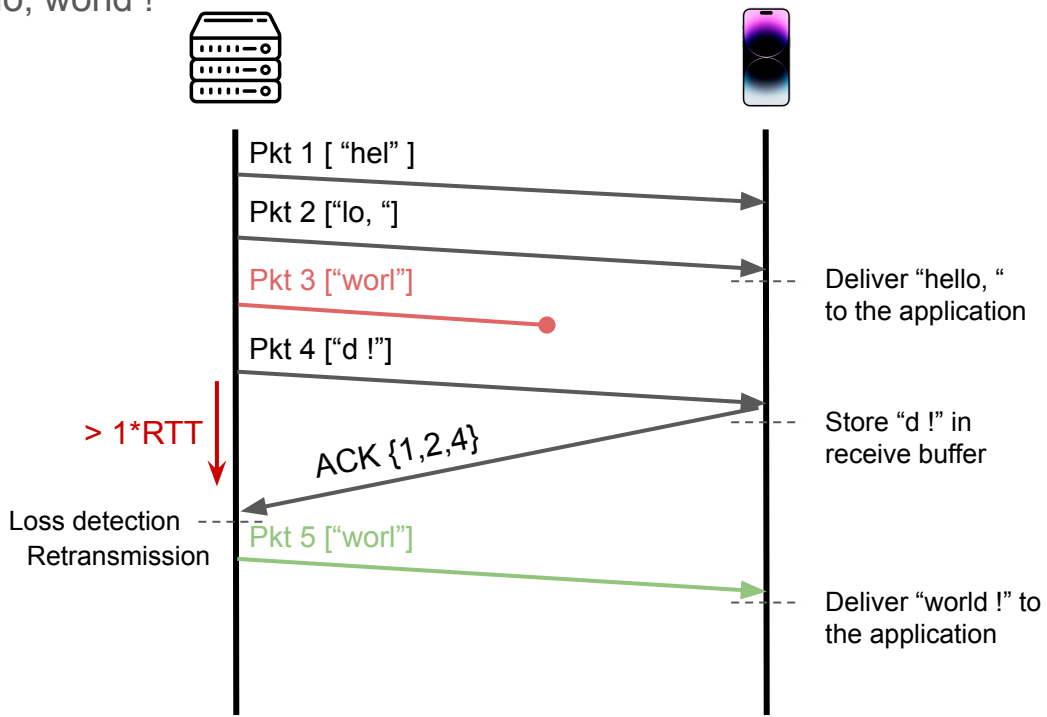
Some results for low latency video streaming

François Michel
Olivier Bonaventure

UCLouvain

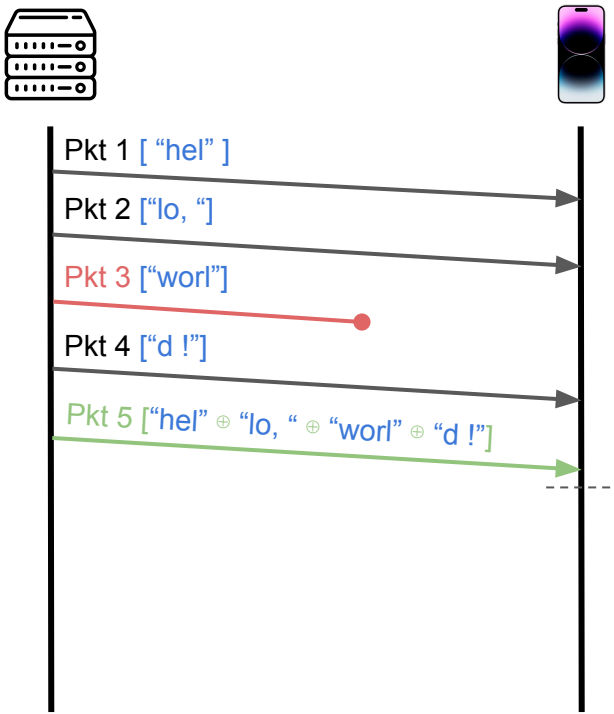
Loss recovery in classical transport protocols (SR-ARQ)

Sending “hello, world !”



Forward Erasure Correction in the transport

Sending “hello, world !”

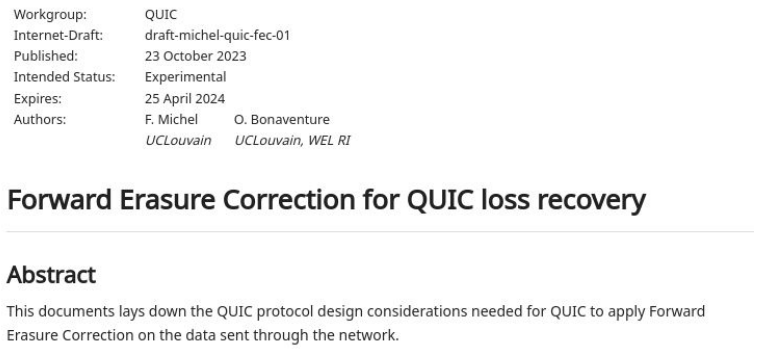


Pkt 1, 2, 3, 4 contain *source symbols*
Pkt 5 contains a *repair symbol*

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

Based on Cloudflare's `quiche` implementation.

- the patch for adding FEC is around 1500 lines (the rest sits in external crates)
- can be integrated with `curl` for HTTP/3 queries
- can probably be integrated with MOQT as well



Workgroup: QUIC
Internet-Draft: draft-michel-quic-fec-01
Published: 23 October 2023
Intended Status: Experimental
Expires: 25 April 2024
Authors: F. Michel (*UCLouvain*) O. Bonaventure (*UCLouvain, WEL RI*)

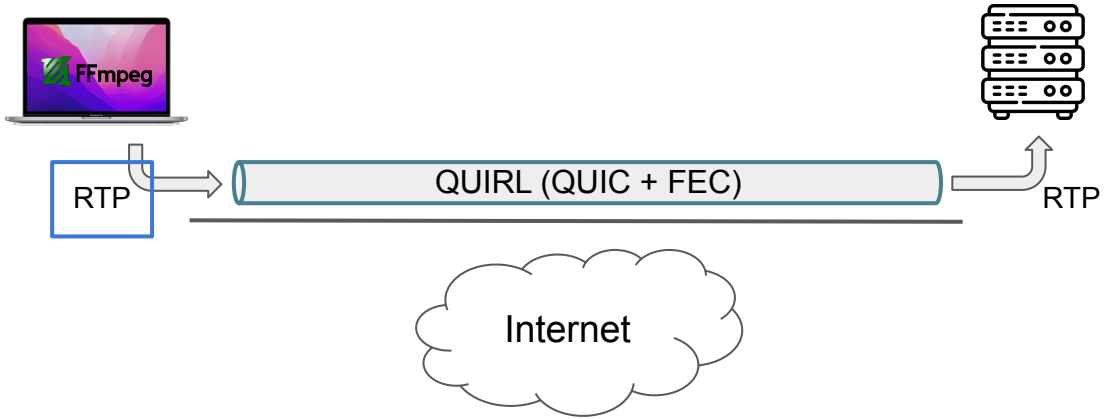
Forward Erasure Correction for QUIC loss recovery

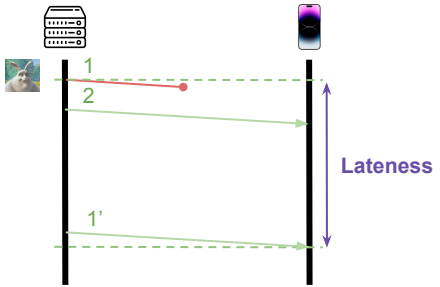
Abstract

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

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)





Sending drone videos over Starlink

Analyzing Real-time Video Delivery over Cellular Networks for Remote Piloting Aerial Vehicles

Aygün Baltaci
ayguen.baltaci@tum.de
Technical University of Munich & Airbus
Germany

Hendrik Cech
hendrik.cech@tum.de
Technical University of Munich
Germany

Nitinder Mohan
mohan@in.tum.de
Technical University of Munich
Germany

Fabien Geyer
fabien.geyer@airbus.com
Airbus
Germany

Vaibhav Bajpai
bajpai@cispa.de
CISPA Helmholtz Center for Information Security
Germany

Jörg Ott
ott@in.tum.de
Technical University of Germany

Dominic Schupke
dominic.schupke@airbus.com
Airbus
Germany

ABSTRACT

Emerging Remote Piloting (RP) operations of electrified Unmanned Aerial Vehicles (UAVs) demand low-latency and high-quality video delivery to conduct safe operations in the low-altitude airspace. Although cellular networks are one of the prominent candidates to provide connectivity for such operations, their ground-centric nature limits their capabilities in achieving seamless and reliable aerial connectivity. In this paper, we study the feasibility of supporting RP operations with low latency and high-quality video delivery over commercial cellular networks. By setting up an adaptive bitrate video transmission pipeline with the Google Congestion Control (GCC) and Self-Clocked Rate Adaptation for Multimedia (SCReAM) Congestion Control (CC) algorithms, we analyze the video delivery performance for the RP application requirements and compare the performance of GCC and SCReAM against constant bitrate video delivery. Our results show that low-latency video delivery with < 300 ms playback latency between full-HD and 4K resolution can be maintained up to about 95% of the time in the air. and meeting the service requirements using cellular networks. We release our collected traces and the video transmission pipeline as open-source to facilitate research in this field.

CCS CONCEPTS

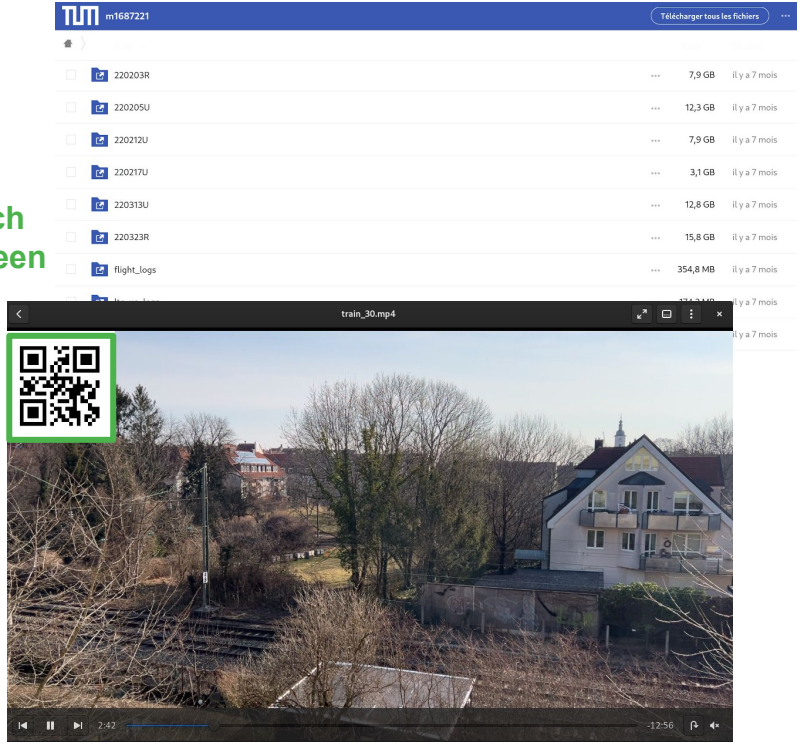
• **Networks** → *Network performance evaluation*; **Network performance analysis**.

KEYWORDS

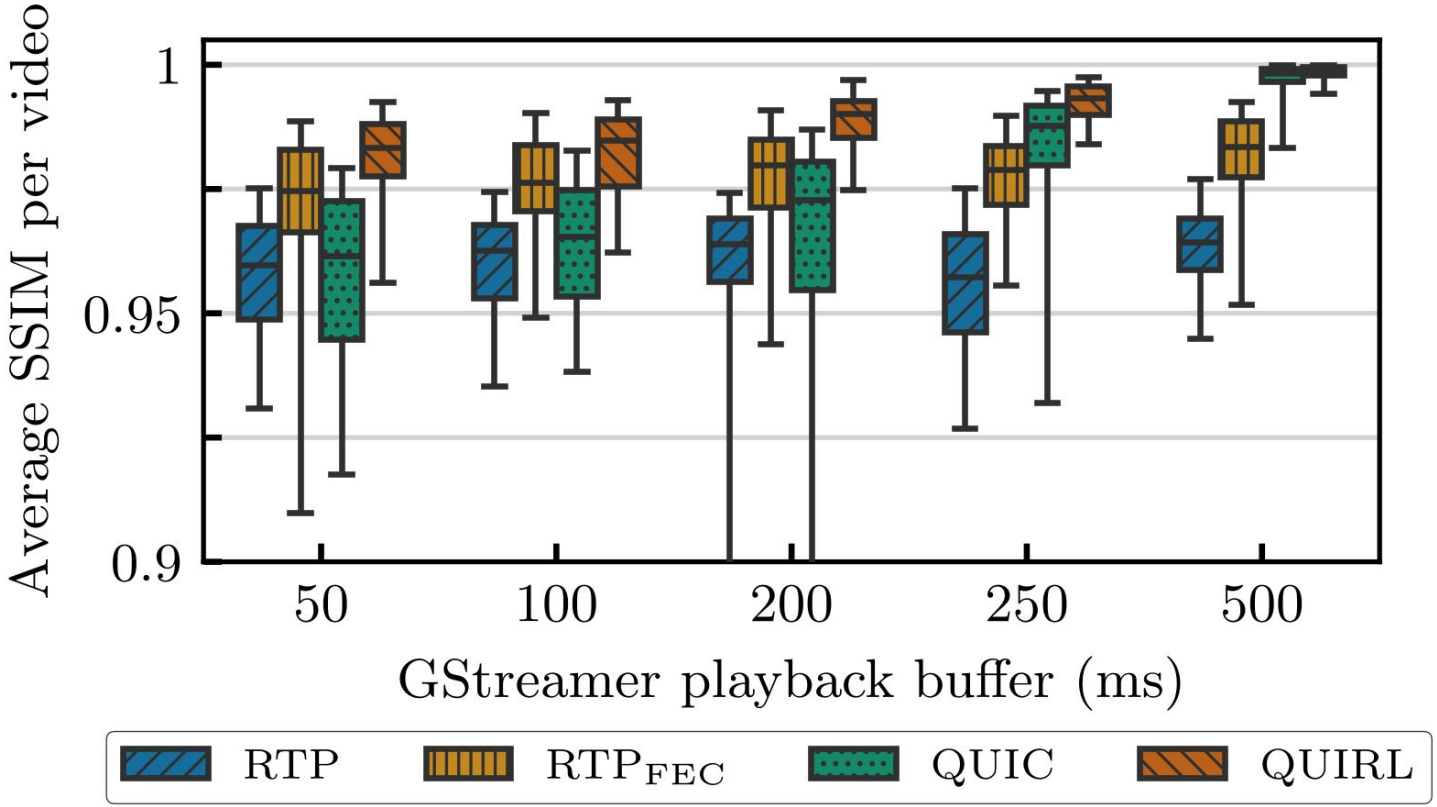
Adaptive streaming; real-time video; cellular networks; LTE; UDP; RTP; UAV; drone; eVTOL; flying taxi; UAM; AAM

ACM Reference Format:
Aygün Baltaci, Hendrik Cech, Nitinder Mohan, Fabien Geyer, Vaibhav Bajpai, Jörg Ott, and Dominic Schupke. 2022. Analyzing Real-time Video Delivery over Cellular Networks for Remote Piloting Aerial Vehicles. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3517745.3561465

Used to match frames between original and playback



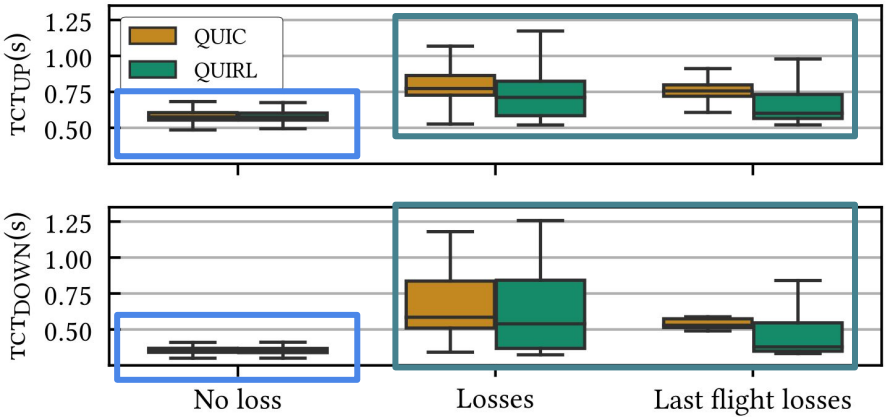
Average SSIM per video over Starlink



If time: Improving `curl`'s Transfer Completion Time (TCT)

We can send repair symbols during quiescence periods, when the cwin allows it.

50kB transfers using `curl` with QUIRL

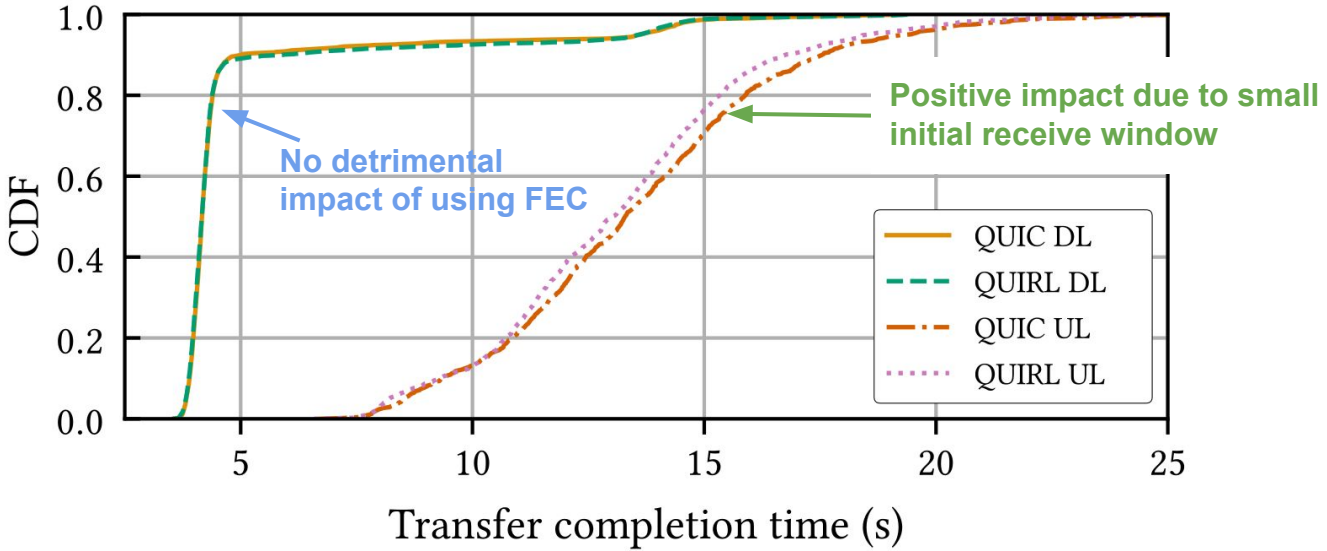


No impact when no loss occurs
Lower completion time upon losses, especially in the last flight

If time: Improving `curl`'s Transfer Completion Time (TCT)

We can send repair symbols during quiescence periods, when the cwin allows it.

10MB file transfers using `curl` with QUIRL



Summary

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

- My thesis: https://ncs.uclouvain.be/assets/pdf/michel-phd.pdf
- The QUIRL paper (soon)

If your use-cases may benefit from QUIC-FEC,

- Discuss on slack and the mailing list
- Send us an e-mail to collaborate: francois.michel@uclouvain.be
- Implement draft-michel-quic-fec