Forward Erasure Correction for QUIC loss recovery

François Michel
Olivier Bonaventure

Imperfect links

Packet losses can happen for at least 2 reasons

- Congestion on the routers/CPE
- Medium imperfections

Besides throughput, those losses can impact the app's responsiveness

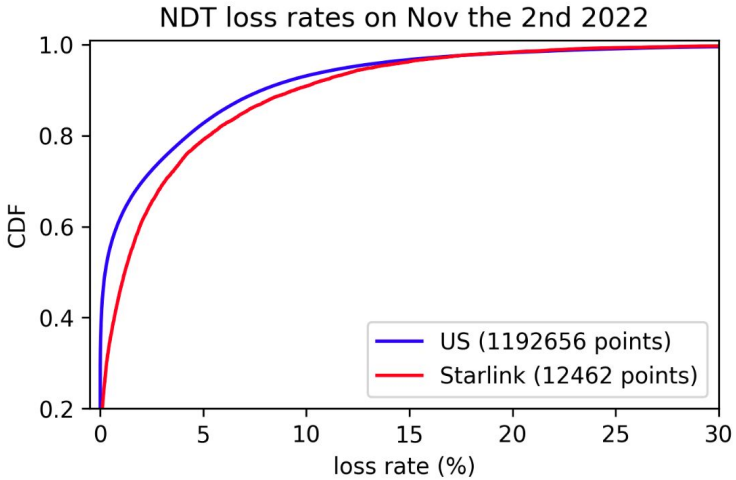
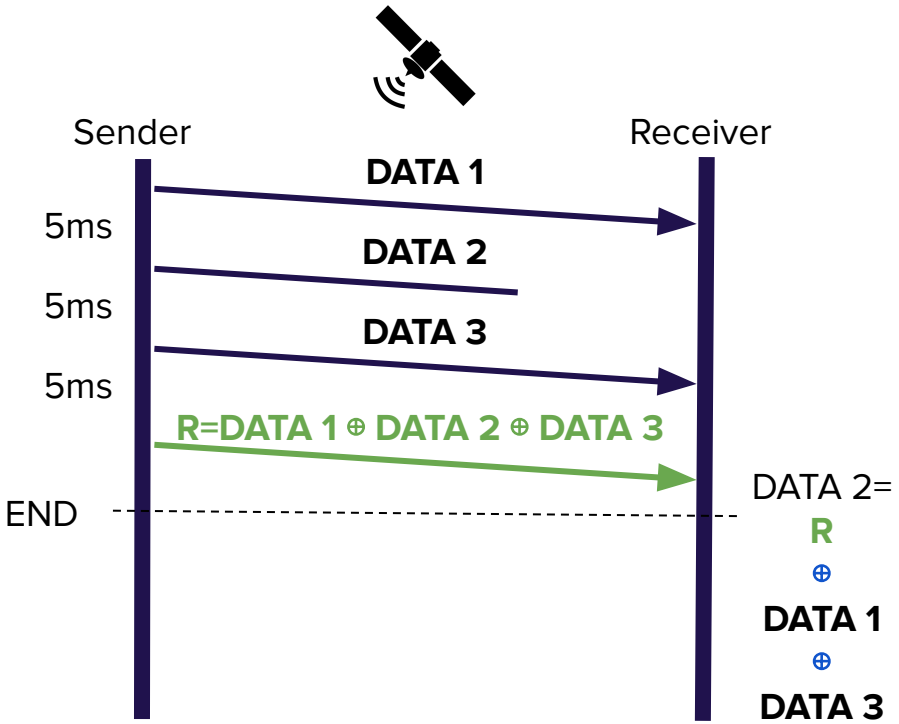


Table 2: QUIC packet loss ratios

H3 ↓	H3 ↑	Messages ↓	Messages ↑
1.56%	1.96%	0.40%	0.45%

A first look at starlink performance, IMC2022

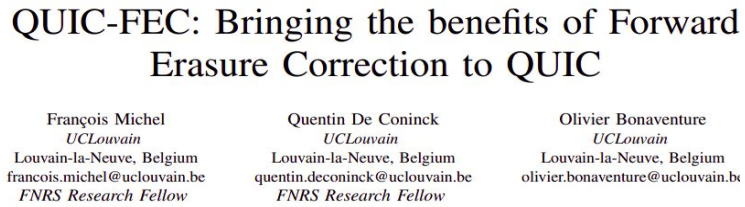
What we can do: Forward Erasure Correction



QUIC + FEC

QUIC's design make it easy to design and experiment with FEC

- Explore several erasure correcting codes
- Several use-cases (H3, MoQ?)

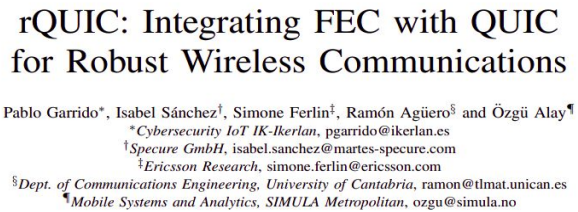


QUIC-FEC: Bringing the benefits of Forward Erasure Correction to QUIC

François Michel
UCLouvain
Louvain-la-Neuve, Belgium
francois.michel@uclouvain.be
FNRS Research Fellow

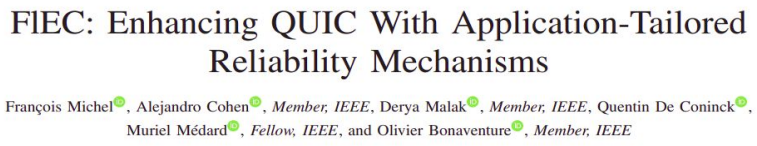
Quentin De Coninck
UCLouvain
Louvain-la-Neuve, Belgium
quentin.deconinck@uclouvain.be
FNRS Research Fellow

Olivier Bonaventure
UCLouvain
Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.b



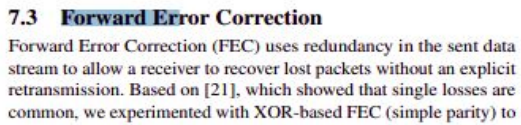
rQUIC: Integrating FEC with QUIC for Robust Wireless Communications

Pablo Garrido*, Isabel Sánchez†, Simone Ferlin‡, Ramón Agüero§ and Özgü Alay¶
*Cybersecurity IoT IK-Ikerlan, pgarrido@ikerlan.es
†Specure GmbH, isabel.sanchez@martes-specure.com
‡Ericsson Research, simone.ferlin@ericsson.com
§Dept. of Communications Engineering, University of Cantabria, ramon@tlmat.unican.es
¶Mobile Systems and Analytics, SIMULA Metropolitan, ozgu@simula.no



FlEC: Enhancing QUIC With Application-Tailored Reliability Mechanisms

François Michel, Alejandro Cohen, *Member, IEEE*, Derya Malak, *Member, IEEE*, Quentin De Coninck, Muriel Médard, *Fellow, IEEE*, and Olivier Bonaventure, *Member, IEEE*



7.3 Forward Error Correction

Forward Error Correction (FEC) uses redundancy in the sent data stream to allow a receiver to recover lost packets without an explicit retransmission. Based on [21], which showed that single losses are common, we experimented with XOR-based FEC (simple parity) to

QUIC + FEC on Starlink

- Simple implementation using Cloudflare's quiche
- H3 50kB uploads over Starlink
- Repair frames sent at the end of the upload, within what is allowed by the cwin

	Med(DCT)	Med(DCT) FEC	pct95(DCT)	pct95(DCT) FEC
All transfers	248 ms	247 ms	382 ms	373 ms
Transfers with losses	287 ms	277 ms	395 ms	386 ms
Transfers with losses in the 10 last pkts	311 ms	279 ms	560 ms	447 ms

No harm in median among all transfers
Noticeable improvement when losses happen

draft-michel-quic-fec

A design to experiment with different implementations

- Distinguishes packet acks from symbol acks (c.f. RFC9265)
- Negociates a coding window to avoid over-complicated packet recoveries
- Compatible with different error correcting codes
  - ID frame common to every ECC
  - REPAIR frame specific to the ECC used
  - Negotiated through transport parameters

FEC and IPR

Several erasure correcting codes are under active patents

- Raptor codes (until 2030): https://datatracker.ietf.org/ipr/2554/
- RLC (until 2034)

Some of them, such as Reed Solomon & LDPC are not under patents

- Interesting but not fountain codes

A good candidate: draft-irtf-nwcrg-tetrys ?

- Fountain-code
- No IPR disclosure
- Described as “patent-free”

Tetrys, a Patent Free, On-the-fly Network Coding Protocol

Next steps: let’s experiment together !

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

- Discuss the draft on the mailing list
- Implementing draft-michel-quic-fec
- Using Tetrys-like ECC as encoder/decoder for experiments
  - We have a rust lib for the encoder/decoder with bindings for C/C++.
    - Send us an e-mail to get it and experiment with us: francois.michel@uclouvain.be
- Please, do it with us, not on your own ! :-)

backup slide

netem badly imitates real losses

Wi-Fi & latency

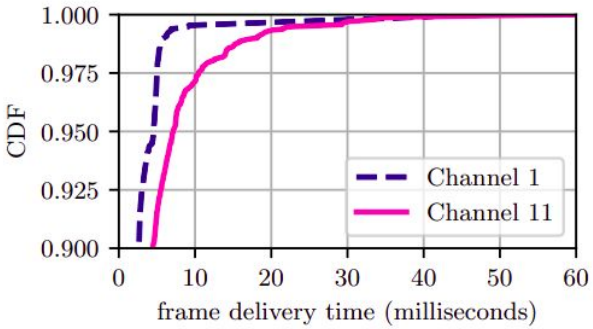
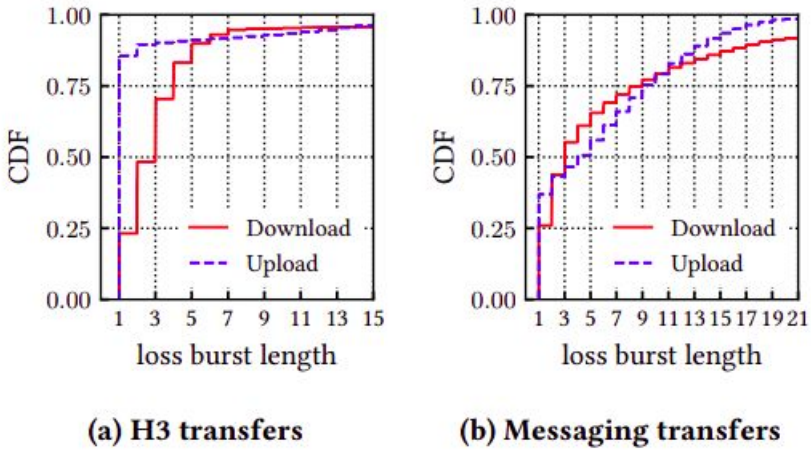


Figure 1: *Frame delivery time CDF for 802.11n on channels 1 and 11.*

Starlink & loss bursts



(a) H3 transfers **(b) Messaging transfers**