FECFRAME version 2
Adding convolutional FEC codes support to the FEC Framework

Vincent Roca, Inria, France

Ali Begen, Networked Media, Turkey

https://datatracker.ietf.org/doc/draft-roca-tsvwg-fecframev2/

July 2016, IETF96, Berlin

Note well

- **we, authors of -01 version, didn't try to patent** any of the material included in this presentation/I-D
- **we, authors of -01 version, are not reasonably aware** of patents on the subject that may be applied for by our employer
- if you believe some aspects may infringe IPR you are aware of, then fill in an IPR disclosure and please, let us know

FECFRAME [RFC 6363]

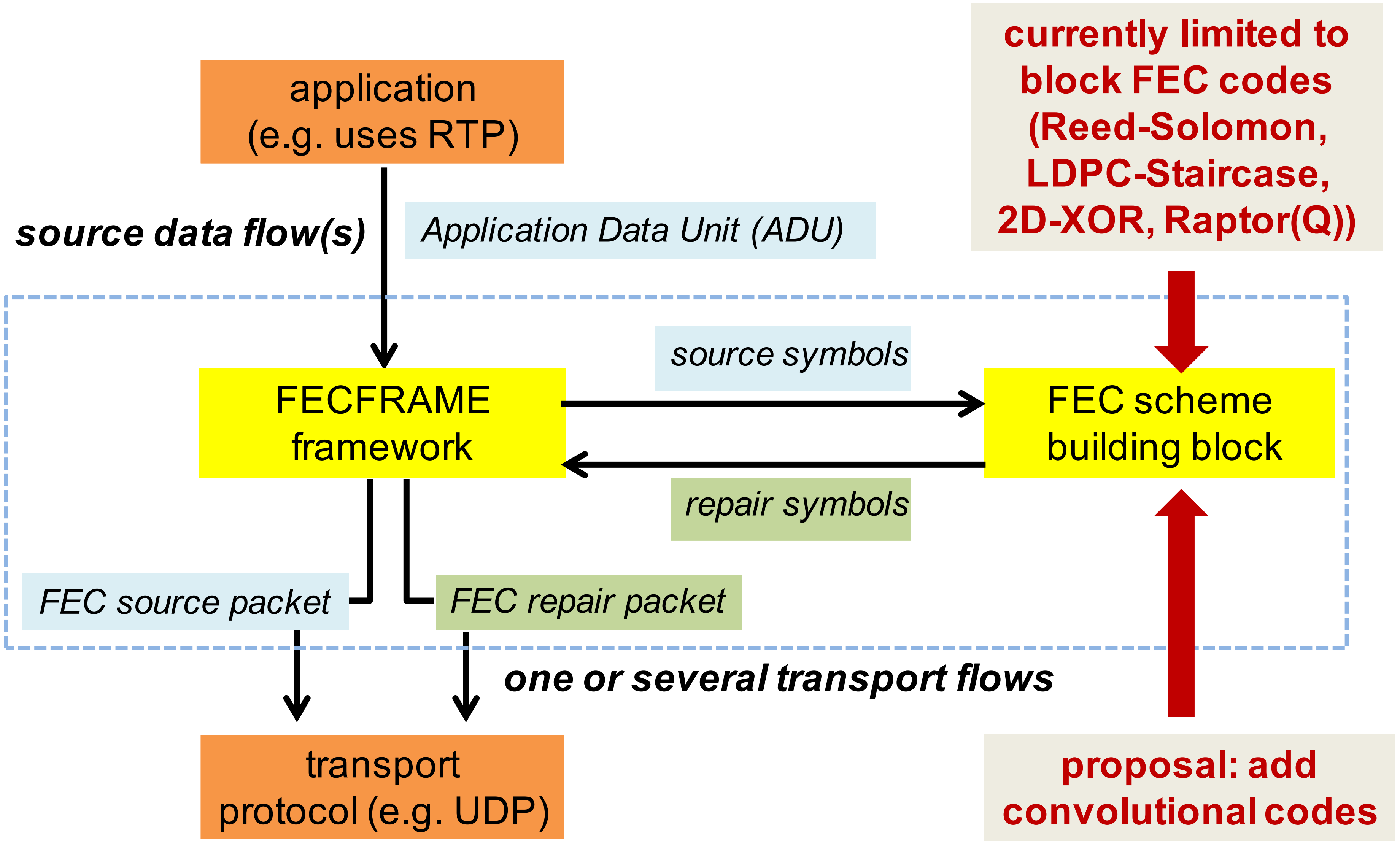
- a follow-up of the "Forward Error Correction (FEC) Framework", A.K.A. FECFRAME
  - RFC 6363, M. Watson, A. Begen, V. Roca, October 2011
    - **produced by the FECFRAME IETF WG**
    - **goal of FECFRAME is to add AL-FEC protection to real-time unicast or multicast flows in a flexible way**
  - already part of **3GPP (e)MBMS** standards

FECFRAME target use-case example

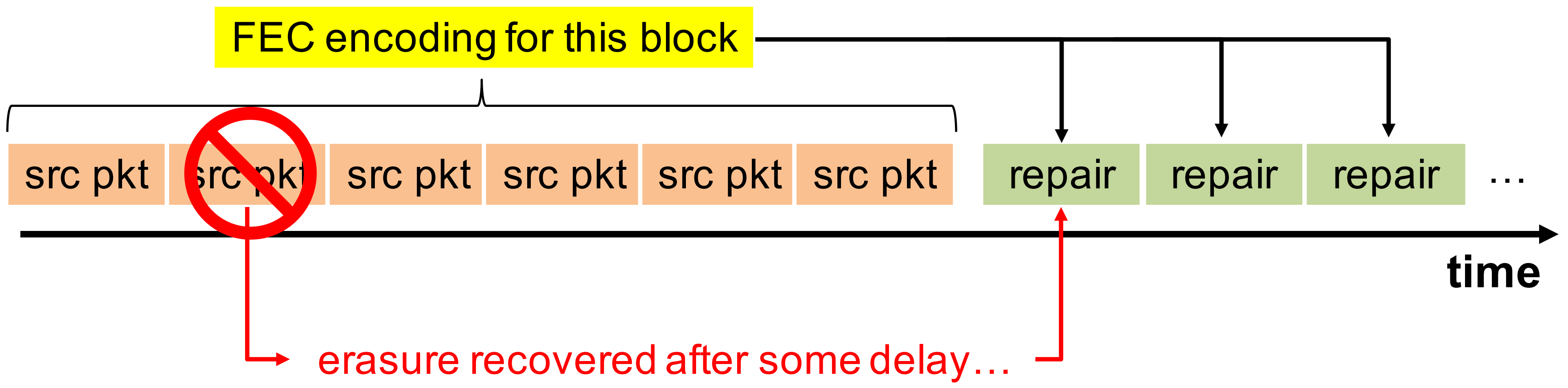
- 3GPP Multimedia Broadcast/Multicast Service (MBMS) are perfect for scalable delivery
  - **everybody's interested by the same content at the same time at the same place**
  - **FLUTE/ALC ⇒ files (largely deployed)**
  - **FECFRAME ⇒ streaming (deployment should begin soon)**
  - **end-to-end latency DOES matter!**

Architecture

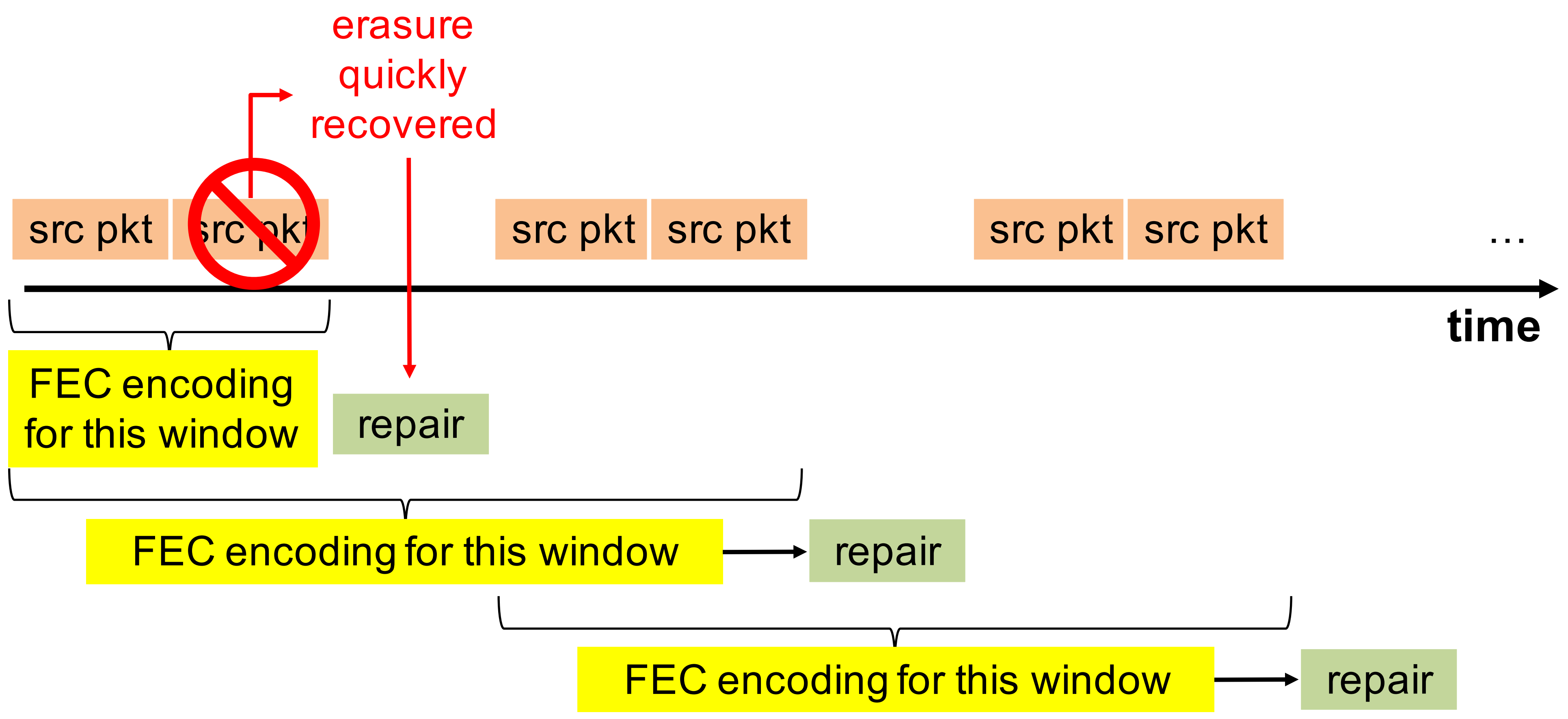
- a **shim layer** to add reliability to real-time flows in a flexible way


currently limited to block FEC codes (Reed-Solomon, LDPC-Staircase, 2D-XOR, Raptor(Q))
application (e.g. uses RTP)
source data flow(s)
Application Data Unit (ADU)
FECFRAME framework
source symbols
FEC scheme building block
repair symbols
FEC source packet
FEC repair packet
one or several transport flows
transport protocol (e.g. UDP)
proposal: add convolutional codes

Block FEC codes… (1)

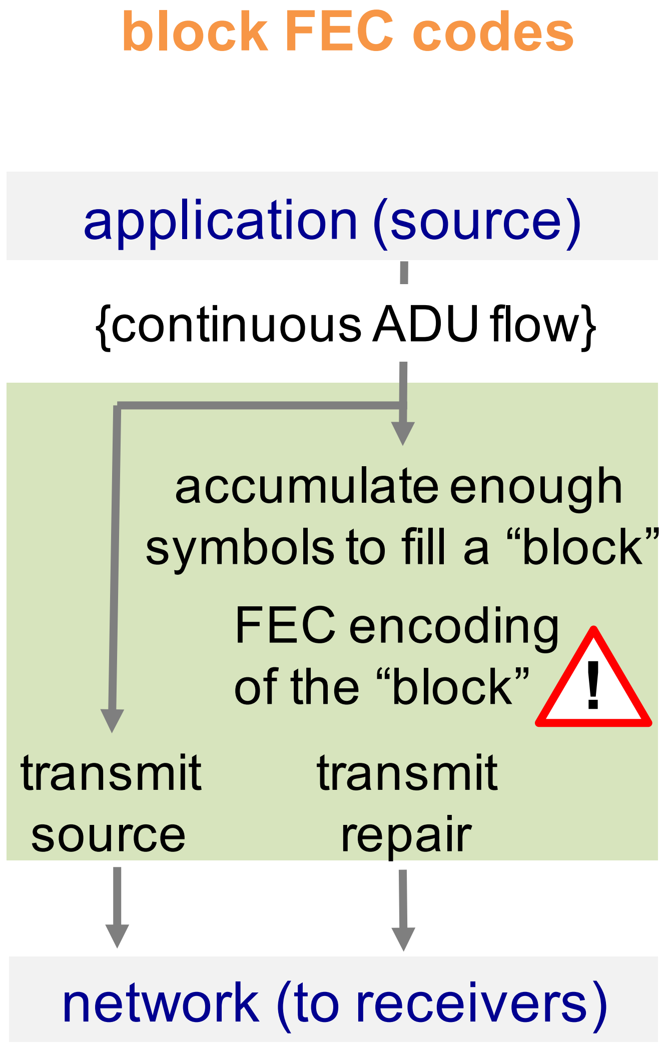


…versus Convolutional FEC codes (2)


erasure quickly recovered
src pkt
src pkt
src pkt
src pkt
src pkt
src pkt
…
time
FEC encoding for this window
repair
FEC encoding for this window
repair
FEC encoding for this window
repair

Why updating FECFRAME? (1)

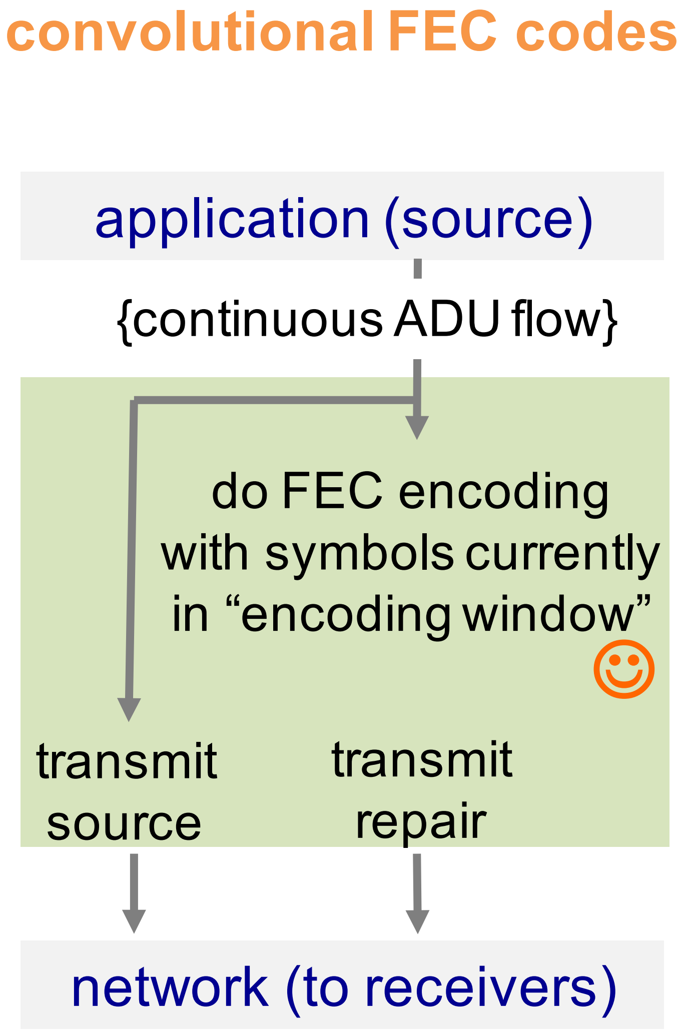
- block FEC codes **add latency to everybody**
  - no matter your reception conditions
  - due to FEC blocks
  - find a balance between added latency and robustness!



NB: we only consider FEC-related latencies here

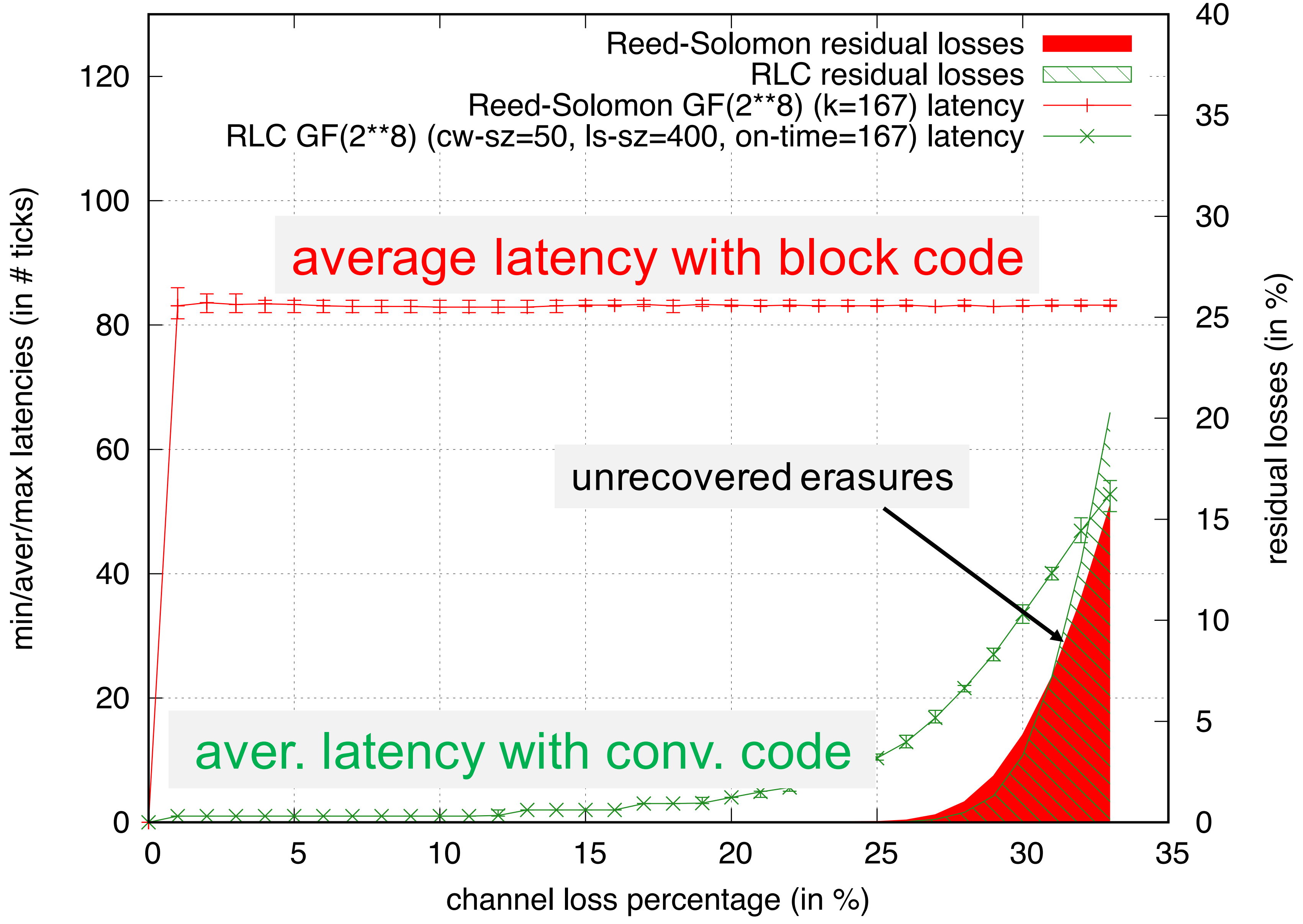
Why updating FECFRAME? (2)

- issue **solved** with convolutional FEC codes
  - **good reception conditions:** **near zero latency** ☺
  - **bad reception conditions:** some latency but unless very close to decoding limits, latency is still **significantly inferior** to that of block codes



Yes it's working ☺

- simulations, CR=2/3 (decoding limit PLR=33.3%)



Updating [RFC6363]

- no fundamental issue
  - no change to existing mechanisms
  - **it's incremental, not disruptive!**
- it DOES NOT break legacy receivers
  - legacy receivers see an unsupported FEC Scheme in the SDP description and ignore the source + repair flows
  - by sending both FECFRAMEv1 and v2 source + repair flows, all the terminals will be satisfied
- it is called version 2…
  - ...but there is no version number in FECFRAME and FEC Schemes

Running code is almost here…

- (non-public) FECFRAME implementation available
  - I did it (Vincent)
  - interoperability tests successful
  - commercialized by http://expway.com

- FECFRAMEv2 implementation in progress…
  - hopefuly ready for IETF 97
  - will rely on our (non-public) convolutional FEC codec already available

What else?

- problem position I-D exists
    - in NetWork Coding Research Group for historical reasons

      https://datatracker.ietf.org/doc/draft-roca-nwcrg-fecframev2-problem-position/

- TODO: propose an equivalent to [RFC5052]…
    - explain how to design FEC Schemes for conv. codes

- TODO: propose convolutional FEC Schemes in the future
    - e.g., for RLC-like codes (very simple)… and others