FECFRAME – extension
Adding sliding window codes support

Sliding Window Random Linear Codes (RLC) FEC Scheme
...for FECFRAME - extended

Vincent Roca, Inria, France


March 2018, IETF101, London
Note well for FECFRAME-ext + RLC l-Ds

- we, authors, didn’t try to patent any of the material included in these l-Ds
- we, authors, 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
Quick reminder

- goal of FECFRAME (RFC 6363) is to make real-time unicast or multicast flows robust to packet losses
  - already part of 3GPP MBMS
  - deployment in progress

- FECFRAME is limited to block codes…
  - e.g., Reed-Solomon, LDPC, Raptor(Q)

- 1st I-D adds support to sliding window codes
- 2nd I-D defines the RLC sliding window codes
Why?

- RLC performs **always better** than any block code with time-constrained flows
  - reduced FEC added latency
  - improved robustness for real-time flows

→ RLC achieves the desired target quality with significantly less repair traffic

references:


- our WiMob’2017 article:  [https://hal.inria.fr/hal-01571609v1/en/](https://hal.inria.fr/hal-01571609v1/en/)

- our research report (2016):  [https://hal.inria.fr/hal-01395937/en/](https://hal.inria.fr/hal-01395937/en/)
Last open question @ IETF 100

-01 adds a density parameter to reduce complexity
  - balance to find between complexity/erasure correction

- Q: does it really reduce complexity?

- we further optimized our codec and tested
  - clear benefits at the encoder
    - e.g., density $\frac{1}{2} \rightarrow$ increases speed by $\sim 80\%$
  - clear benefits at the decoder
    - e.g., density $\frac{1}{2} \rightarrow$ increases speed by up to $\sim 50\%$

- conclusion: yes, it can help with CPU constrained devices or with larger coding windows
Running code

● (non-public) FECFRAME implementation available
  ○ I did it
  ○ compliant to 3GPP MBMS and successful interop. tests

● (non-public) FECFRAME-extended implementation
  ○ I did it too

● (non-public) RLC implementation
  ○ Belkacem (I-D co-author) did it
Next step

● both I-Ds are ready for TSVWG WGLC
  ○ I’ll ask NWCRG for complementary reviews

● FECFRAME-ext I-D:
  ○ 20 pages long (all included)
    ○ includes 6 pages ½ of introduction/definitions/architecture
    ○ includes 4 big figures easy to read
    ○ 1 page non normative annex that can be skipped ;-)

● RLC I-D:
  ○ 27 pages long (all included)
    ○ no complex math, promised ;-)
    ○ it’s next generation end-to-end FEC code that will soon be used throughout Internet when latency does matter (not just in FECFRAME) ;-)