About FECFRAME version 2
Adding convolutional FEC codes support to the FEC Framework
NWCRG discussion

Vincent Roca, Inria, France
Ali Begen, Networked Media, Turkey
https://datatracker.ietf.org/doc/draft-roca-tsvwg-fecframev2/
November 2016, IETF97, Seoul
Note well

- we, authors of -02 version, didn’t try to patent any of the material included in this presentation/I-D
- we, authors of -02 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
Work undergone so far

- initial work at NWCRG
  - problem position I-D
  - discussed during IETF93

- activity moved to TSVWG
  - draft-roca-tsvwg-fecframev2
  - discussed during IETF 96 and IETF 97

- two sides
  - protocol support of convolutional FEC schemes
  - convolutional FEC schemes

- FECFRAME v2 is the common umbrella
  UNDER PROGRESS

- RLC
- SRLC under progress
- potentially any conv. FEC scheme
  REMAINS TBD
FECFRAME vs. FECFRAMEv2

- a follow-up of [RFC 6363] describing FECFRAME
  - RFC 6363, M. Watson, A. Begen, V. Roca, October 2011
- a shim layer for robust and scalable distribution of real-time flows
  - already part of 3GPP (e)MBMS standards
  - we start to have deployment experience
- FECFRAME relies on block FEC codes
  - single encoder and single decoder (no recoding)
FEC encoding for this block

src pkt src pkt src pkt src pkt src pkt src pkt

repair repair repair

recovered

erasure recovered after some delay

switch to convolutional codes

FEC encoding for this window

src pkt src pkt repair src pkt src pkt repair src pkt src pkt repair

recovered

FEC encoding for this window

...
block codes add latency to everybody, all the time

... this issue is solved with convolutional FEC codes
  - good reception conditions: near zero latency 😊
  - bad reception conditions latency: still significantly inferior

v2 adds convolutional code support
  - in a fully backward compatible way

FECFRAME vs. FECFRAMEv2... (3)
**Status of the work**

- **FECFRAMEv2 document relatively mature**
  - we do not expect major changes to I-D now

- we have a FECFRAME implementation experience
  - leverages on a FECFRAME implementation (Vincent) being commercialized (Expway), for which interop. tests have been conducted
  - FECFRAMEv2 implementation under progress (Vincent)

- we have a conv. code implementation experience
  - RLC over GF(2^8) codec implemented in a private version of OpenFEC.org
  - SRLC codec under progress
Status of the work... (2)

● we made progress in terms of block vs convolutional codes evaluation
  ○ block FEC codes are totally sub-optimal for real-time flows
  ○ true with small or larger block/encoding window sizes
  ○ motivates the need for FECFRAME v2

latency CDF with conv. codes

intermediate rx conditions

(a) R-S vs. RLC CDF when loss = 15%

(b) R-S vs. RLC CDF when loss = 25%

(c) R-S vs. RLC CDF when loss = 30%

latency CDF with block R-S codes

very bad rx conditions

V. Roca, B. Teibi, C. Burdinat, T. Tran, C. Thienot, "Block or Convolutional AL-FEC Codes? A Performance Comparison for Robust Low-Latency Communications", https://hal.inria.fr/hal-01395937, November 2016.
Close-up on RLC over GF($2^8$) signaling

- pretty simple signaling...
  - ... because there's no recoding
    - do not transport coefficients but only a key that along with a predefined PRNG enables to generate them
  - ... because we only consider encoding windows w/o gaps
    - carry starting source Encoding Symbol ID (ESI) + number symbols
  - makes header/trailer pretty **compact** :-)

- unlike block codes, we need to separate ESI spaces
  - one for source symbols
  - one for repair symbols (in this case more a key than an ESI)
Close-up on RLC over GF(2^8) signaling... (2)

/**
 * Source and Repair FEC Payload ID encoding formats, RLC over GF(2^8):
 * 
 *   1                   2                   3
 * 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 * | Source Encoding Symbol ID (ESI) |
 * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 * Source FEC Payload ID (trailer added to source symbols)
 *
 *   1                   2                   3
 * 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 * | Repair key | nb symbols in coding window |
 * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 * | ESI of first symbol in coding window |
 * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 * Repair FEC Payload ID (header added to repair symbols)
 */
Next steps

• TODO 1: finalize FECFRAME v2 implementation
  ● to be sure we didn't miss anything
  ● sender already okay, receiver will be done for IETF98

• TODO 2: propose RLC convolutional FEC Scheme
  ● all the convolutional FEC code complexity is here!
    ● specify all code details
    ● specify all signaling aspects
    ● identified by a IANA registered FEC Encoding ID