

***FECFRAME – extension***  
***Adding sliding window codes***  
***support to FECFRAME***

**Vincent Roca, Inria, France**

**Ali Begen, Networked Media, Turkey**

<https://tools.ietf.org/html/draft-ietf-tsvwg-fecframe-ext/>

**July 2017, IETF99, Prague**

## **Note well for FECFRAME-ext + RLC I-Ds**

- **we, authors, didn't try to patent** any of the material included in this presentation/I-D
- **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

# Main changes since IETF'98

- now a WG-Item document
  - <https://datatracker.ietf.org/doc/draft-ietf-tsvwg-fecframe-ext/>
  - file name does not refer to a new protocol version any more
- references to “convolutional codes” have been replaced by “sliding window codes”
  - “convolutional” is too much associated to PHY layer codes
  - “sliding window” illustrates well the code behavior
- filled “Section 3. Architecture Overview”
  - introduces FECFRAME architecture, terminology, and key concepts
  - goal is to help a newcomer to understand FECFRAME

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

**Vincent Roca, Inria, France**

<https://datatracker.ietf.org/doc/draft-ietf-tsvwg-rlc-fec-scheme/>

July 2017, IETF99, Prague

# Main changes since IETF'98

- now a WG-Item document
  - <https://datatracker.ietf.org/doc/draft-ietf-tsvwg-rlc-fec-scheme/>

---

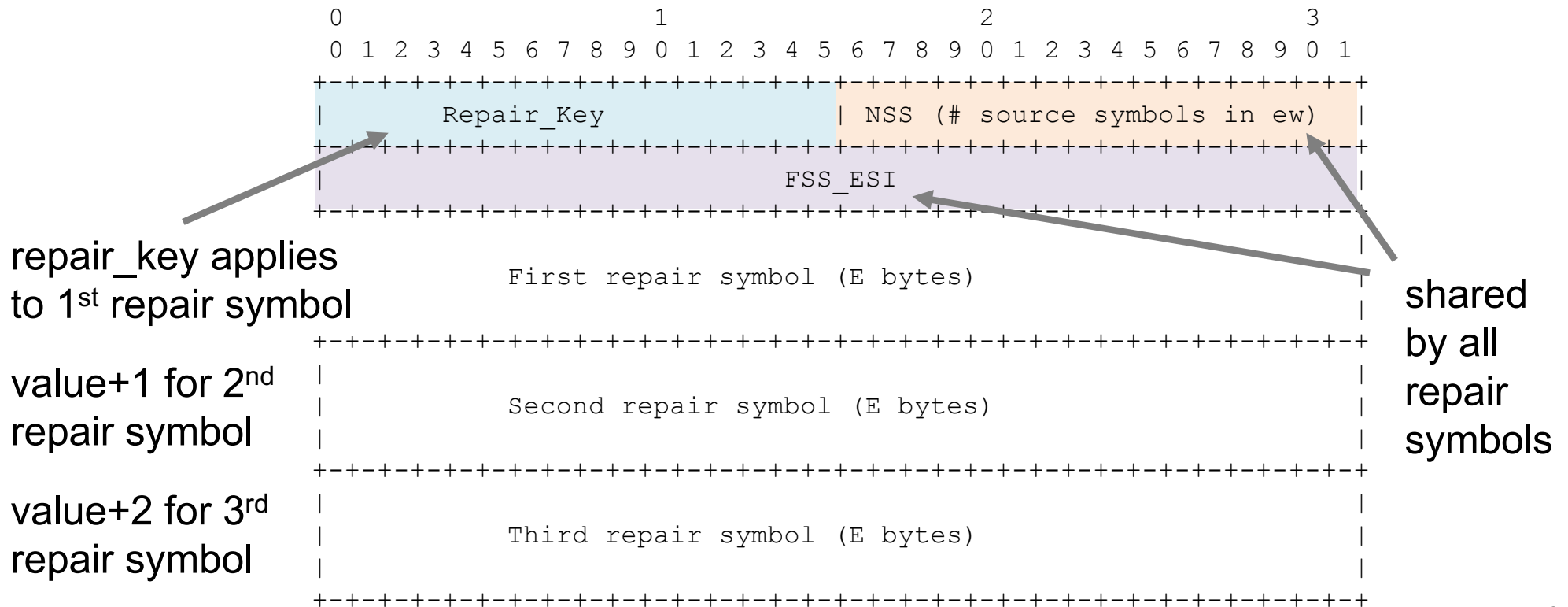
Private feedbacks from Marie-José Montpetit triggered some of the following updates (thanks)

---

- references to “convolutional codes” have been replaced by “sliding window codes”
  - same reasons
- kept the RLC acronym but changed the expanded name to add “sliding window”

# Main changes since IETF'98 (2)

- support multiple repair symbols per FEC Repair Pkt
  - the **main** technical change
  - useful when the symbol size **E** is small compared to the PMTU
  - avoids extra overhead (uses a single RLC IP/UDP/RLC header)
  - # repair symbols determined by comparing UDP payload length and symbol size (**E**)



## ***Main changes since IETF'98 (3)***

- filled Section 5. “FEC Code Specification”
  - describes encoding and decoding operations
  - sections 3, 4 and 5 are sufficient to have interoperable implementations of this FEC Scheme ☺
- detailed sections 7 “Security Considerations” and 8 “Operations and Management Considerations”

# Next steps for both I-Ds

- FECFRAME-extension: almost done
  - proof-reading
- Sliding Window RLC: a few things...
  - proof-reading
  - Open question: adding support of sparse versions of RLC?
    - to date we only considers high-density encoding vectors...
    - ... what about adding support to sparse matrices for **larger** encoding windows as a **compromise** between correction performance and complexity?

$$\text{repair}_1 = \alpha_1 * \text{src}_1 + \alpha_2 * \text{src}_2 + \alpha_3 * \text{src}_3 + \alpha_4 * \text{src}_4 + \alpha_5 * \text{src}_5 + \alpha_6 * \text{src}_6$$

density 1

↓ adding?

$$\text{repair}_1 = \alpha_1 * \text{src}_1 + 0 + 0 + \alpha_4 * \text{src}_4 + \alpha_5 * \text{src}_5 + 0$$

density 1/2