

# **FEC and NC performance evaluation**

**vincent.roca@inria.fr**

NWCRG Sep. 25<sup>th</sup>, 2018 Interim meeting

# Several dimensions

## 1. Performance metrics

- what?

## 2. Evaluation methodology

- how?

## 3. Communication channels considered

- which use-cases?

## 4. Tools

- a hackathon's project?

# Previous presentations

- **Already mentioned in NWCRG as an interesting topic + a few hints**
  - ✓ “Performance and Feature Comparison of Erasure Correcting Coding Software Libraries”, Steinwurf
    - <https://www.ietf.org/proceedings/89/slides/slides-89-nwcrg-4.pdf>
- **Also, for a concrete use-case:**
  - ✓ “FECFRAME–extension: Adding convolutional FEC codes support to the FEC Framework”, Vincent Roca et al. (slides 13-23)
    - <https://datatracker.ietf.org/meeting/98/materials/slides-98-tsvwg-sessb-63-fecframe-drafts-00>

# Topic 1: Performance metrics

- **usual metrics:**
  - erasure correction performance
    - ✓ average overhead, decoding failure probability WRT number packets received, ...
  - codec oriented
    - ✓ encoding and decoding speed, required memory, ...
- **but also:**
  - added FEC-related latency to achieve a certain quality (residual packet erasures below a threshold, e.g.,  $10^{-3}$ )

# Topic 1: Performance metrics (2)

- **time is difficult to catch...**
  - e.g., to evaluate FEC-related latency on the whole path
    - ✓ requires to be reproducible
    - ✓ easier with a CBR source flow
- **we proposed a methodology in**
  - “Less Latency and Better Protection with AL-FEC Sliding Window Codes: a Robust Multimedia CBR Broadcast Case Study”, <https://hal.inria.fr/hal-01571609v1/en/>
  - (no time to summarize it here, see article)

## Topic 2: Evaluation methodology

- **3 main approaches**

- theoretical analyzes

*important but not addressed thereafter*

- simulated end-to-end transmission

**+ fully controlled and reproducible**

**- accuracy needs to be checked**

- real-world experiments

**+ maximum accuracy (if done correctly)**

**- complex, partial control, less reproducible**

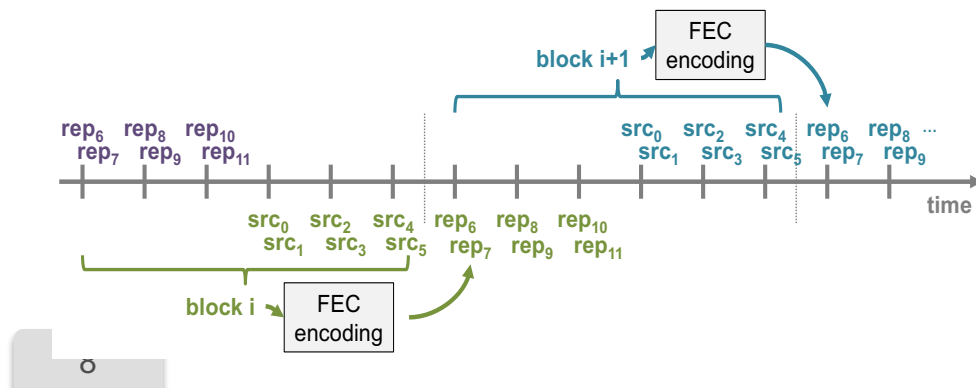
## Topic 2: Evaluation methodology (2)

- **example of simulated transmission system: “eperftool”**
  - (cf. <http://openfec.org>)
- *a single process for everything*
- *no true transmission, it's simulated*
- *true FEC encoding, true FEC decoding, with a real codec*
- *packet losses are simulated (various loss models, possibility to use true loss traces)*

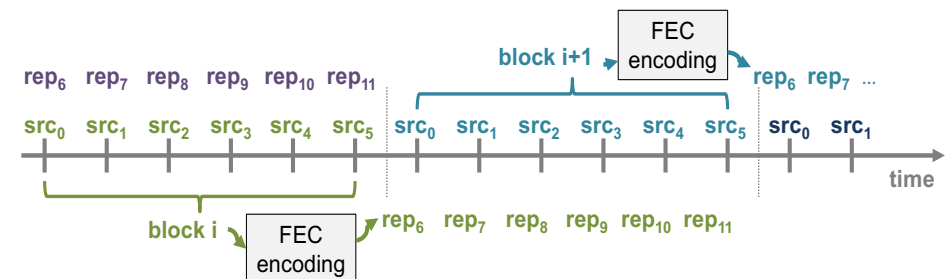
## Topic 2: Evaluation methodology (3)

- **beware of codec design choices**
  - how source and repair packets are transmitted (tx order) will impact results significantly...
  - example: block FEC code, CBR output

**single output leaky bucket: sending repair packets takes time and delays source packets**



**two output leaky buckets: repair and source packets transmitted in parallel**





# Topic 3: Communication channels considered

- we need channel models
- we've been using 3GPP SA4 official mobility traces

- which packets are lost?
- not universal (obtained under precise circumstances) but useful
- freely available

(see <https://hal.inria.fr/hal-01571609v1/en/>)

○vehicle passenger ⇒ losses are "evenly" spread

4 different average loss rates (1%, 5%, 10%, 20%)



○pedestrian ⇒ loss bursts

4 different average loss rates (1%, 5%, 10%, 20%)



- do we have something else?

## Topic 4: Tools

- **develop a performance evaluation environment**
  - in a hackathon's project?
  - ... once the codec is available

# A side topic: define key codec parameters

- **what** are the parameters?
- **how to derive** the parameters?
  - depending on the flow features (real-time or not)
- **possible answer: “sliding window RLC FEC Scheme” I-D**
  - defines relationships between tolerable latency budget, encoding window size, decoding window size, linear system size
    - [see: https://tools.ietf.org/html/draft-ietf-tsvwg-rlc-fec-scheme](https://tools.ietf.org/html/draft-ietf-tsvwg-rlc-fec-scheme)

|               |  |           |
|---------------|--|-----------|
| <u>3.</u>     | Procedures . . . . .                     | <u>7</u>  |
| <u>3.1.</u>   | Possible Parameter Derivations . . . . . | <u>7</u>  |
| <u>3.1.1.</u> | Case of a CBR Real-Time Flow . . . . .   | <u>8</u>  |
| <u>3.1.2.</u> | Other Types of Real-Time Flow . . . . .  | <u>10</u> |
| <u>3.1.3.</u> | Case of a Non Real-Time Flow . . . . .   | <u>11</u> |

## Wrap up

- in addition to theoretical analyzes (not discussed here)...

- essential and non trivial

- **outcomes:**

- a new I-D?
- a new hackathon project?
- another I-D on parameters derivation

**sure!**

**stay focused: codec is #1 prio**

**probably**

- **we need to share experience/tools/channels/good practices/...**
  - who's interested?