



RLC and AL-FEC @ IETF: when codes meet transport protocols and practical aspects

Vincent Roca

► To cite this version:

Vincent Roca. RLC and AL-FEC @ IETF: when codes meet transport protocols and practical aspects. Angeles Vazquez-Castro. Algebraic approaches to storage and network coding - COST IC1104, Feb 2014, Barcelone, Spain. hal-00948895

HAL Id: hal-00948895

<https://inria.hal.science/hal-00948895>

Submitted on 18 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RLC and AL-FEC @ IETF: when codes meet transport protocols and practical aspects

Vincent Roca, Inria, France

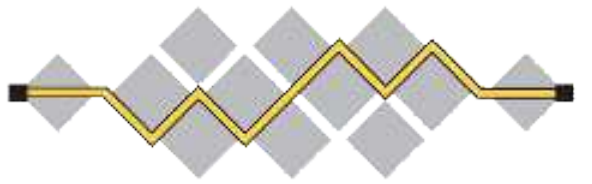
Algebraic approaches to storage and network coding

COST IC1104 meeting

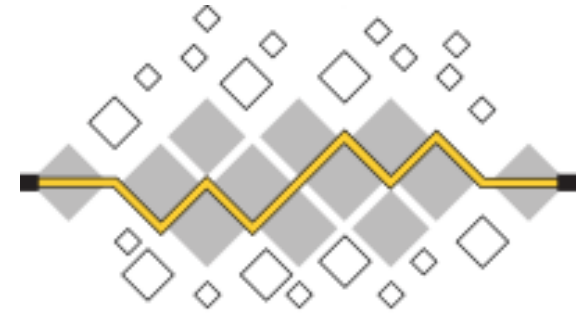
Feb. 7th, 2014, Barcelona

Goals

- focus on **AL-FEC** (all kinds, RLC included) codes for the **erasure channel** and their use in IETF standards
- I'll discuss
 - the key **IETF/IRTF Working Groups**
 - RMT, FECFRAME, DTNRG, NWCRG
 - how is **AL-FEC standardization** addressed at IETF?
 - on the importance of signaling
 - focus on **NetWork Coding (NWCRG)** IRTF activities
 - Tetrys on-the-fly coding
 - structured RLC



I E T F[®]



I R T F

A quick survey of related IETF/ IRTF working groups

About IETF

- Internet Engineering Task Force <http://ietf.org/>
 - the place where Internet technology is standardized
 - TCP/IP and much more...
 - historically focusing on protocols, but now embraces FEC
 - they play a major role in recent communication systems!
 - open to everybody
 - no fee, open discussion lists
 - open specifications (Internet-Drafts and RFCs)
 - “Internet” is the target use-case
 - later IETF’s technology is often instantiated by other SDOs (3GPP, DVB, OMA, ISDB, ...)

IETF motto: “we believe in rough consensus and working code”

About IETF (con't)

- about IPR (Intellectual Property Rights, i.e. patents)
 - any IETF/IRTF contributor has to disclose any IPR he/she is “reasonably aware of”
 - you may be one of the inventors...
 - it may be one of your colleagues...
 - or anybody else if there is a good reason for you to be aware of the existence of the patent
 - IETF/IRTF takes no position WRT IPR's validity or scope
 - it only provides a registry: <https://datatracker.ietf.org/ipr/>
 - but WGS are authorized to take it into consideration
 - to know more: <https://www.ietf.org/ipr/policy.html>

About IETF... (cont')

- two working groups primarily concerned by AL-FEC
 - **RMT** (reliable multicast transport) WG <http://tools.ietf.org/wg/rmt/>
(1999 - 2013)
 - **AL-FEC and protocols for reliable object distribution to multiple receivers simultaneously**
 - **standardized AL-FEC:**
 - Raptor, RaptorQ, LDPC-Staircase, Reed-Solomon
 - No-code (the most useful one ☺)
 - **AL-FEC and FLUTE/ALC are now widely deployed...**

*Example: ISDB-Tmm (NOTTV), Japan
Relies on FLUTE/ALC and LDPC-Staircase*



Sports program received on a mobile phone

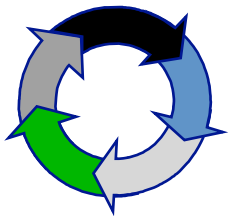


News program on a smart phone

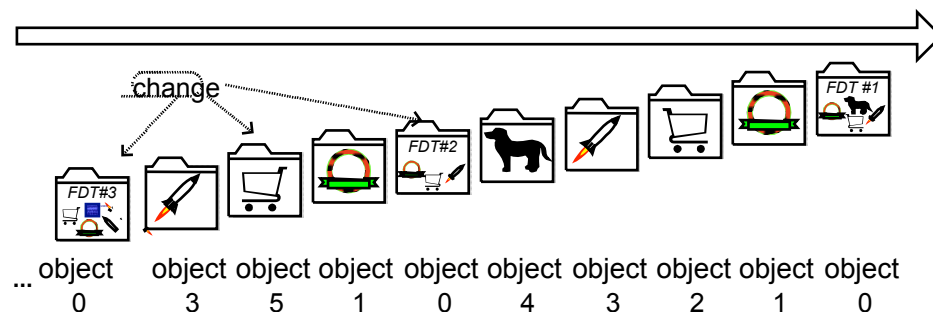
About IETF... (cont')

○ example: FLUTE/ALC

- unidirectional communication (no feedback)
- massively scalable, from 0 to billions of receivers
- transmits files (ALC) and metadata (FLUTE)
- reliability achieved thanks to:
 - the use of AL-FEC
 - the carousel approach (several tx loops)



carrousel (dynamic)
FLUTE/ALC + AL-FEC



• selects object 4



About IETF... (cont')

- **FECFRAME** (FEC Framework) WG <http://tools.ietf.org/wg/fecframe/>
(2006 - 2013)
 - **AL-FEC and protocols for streaming applications**
 - typically RTP/UDP flows
 - **standardized AL-FEC:**
 - Raptor, RaptorQ, LDPC-Staircase, Reed-Solomon, 1D/2D parity
 - **limited deployments so far... but things may evolve**

About IRTF

You said IETF or IRTF?

- Internet Research Task Force

<http://irtf.org/>

- complements the IETF

- focuses on research more than engineering aspects

- it can be the first step before launching an IETF WG

- IETF: 125 WGs in 8 areas

- IRTF: only 9 WGs (!)

About IRTF... (cont')

- two working groups primarily concerned by AL-FEC

- NWCRG (NetWork Coding) RG

<http://irtf.org/nwcrg>

(brand new WG, launched on Nov. 2013)

- RLC and protocols for **network coding**

- more to come...

- DTNRG (Delay Tolerant Networks) RG

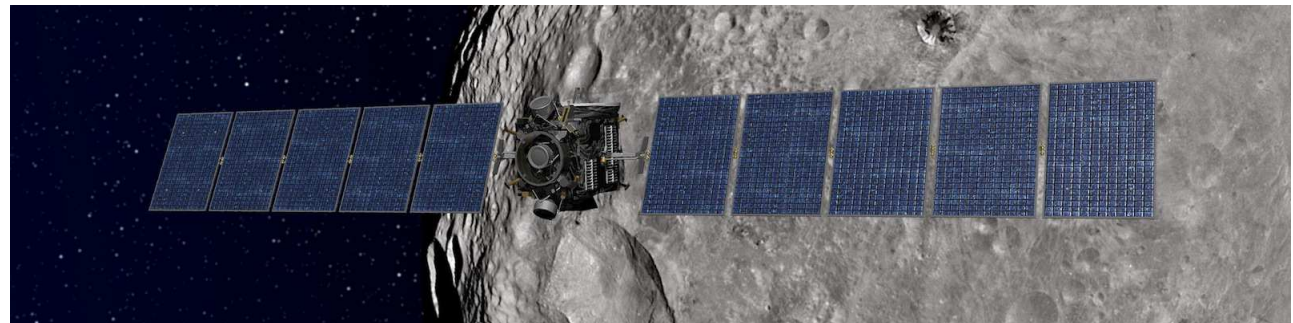
<http://irtf.org/dtnrg>

(2002 – now)

- RLC and protocols for **Delay Tolerant Networks (DTN)**

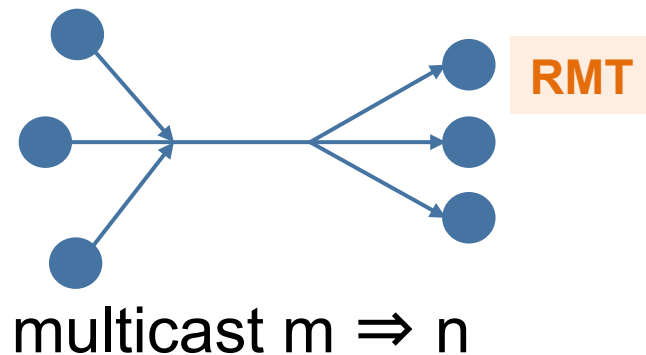
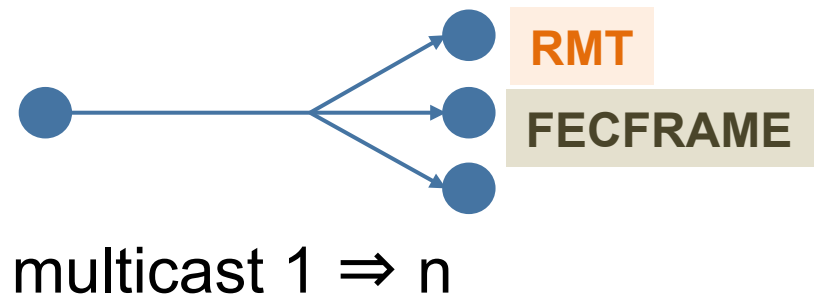
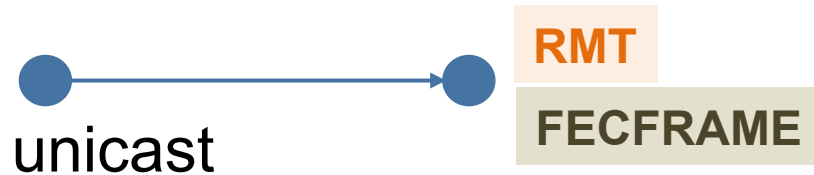
- ex: space communications

- includes an RLC proposal for improved robustness

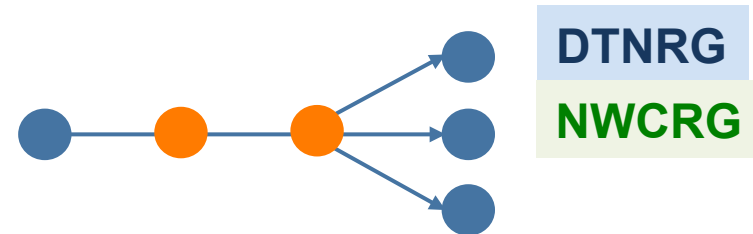


Different transm. models according to WG

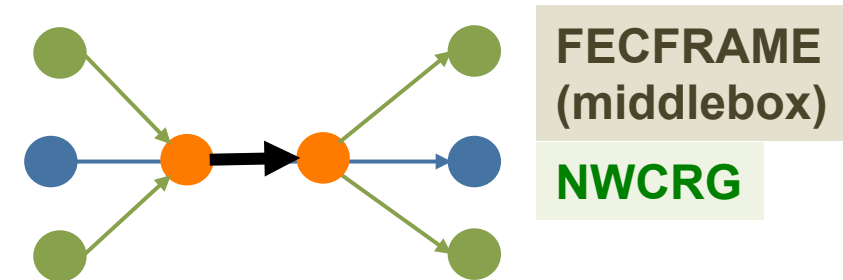
end-to-end



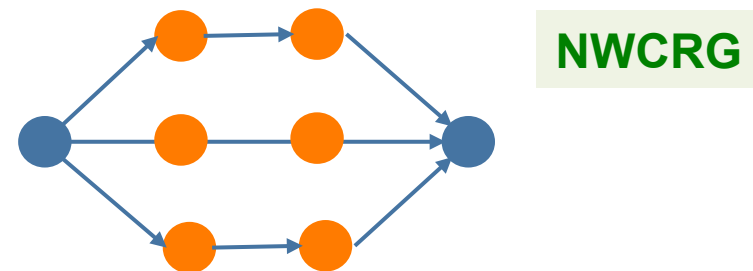
in-network encoding



uni/multicast, intra-flow enc.



uni/multicast, inter-flow enc.



multi-path transfer

How is FEC standardization addressed in IETF?

Situation

- focus on the **erasure channel** only

- we're at IETF

- we observe losses, not transmission errors

- causes: router congestion, bad reception conditions (wireless), intermittent connectivity, etc.

- focus on “**higher**” layers

- we're at IETF

- they are called **Application-Level FEC**, but they are found:

- within the **application**
 - within the **transport** layer (e.g. between RTP/UDP for streaming, in FLUTE/ALC for filecasting)
 - within the **MAC** layer (e.g., in DVB-H/MPE-FEC, or in DVB-SH/MPE-IFEC)
 - NB: not sure for routing layer: AL-FEC maybe not? NC maybe?

- a direct consequence: everything is done in **software**

Situation... (cont')

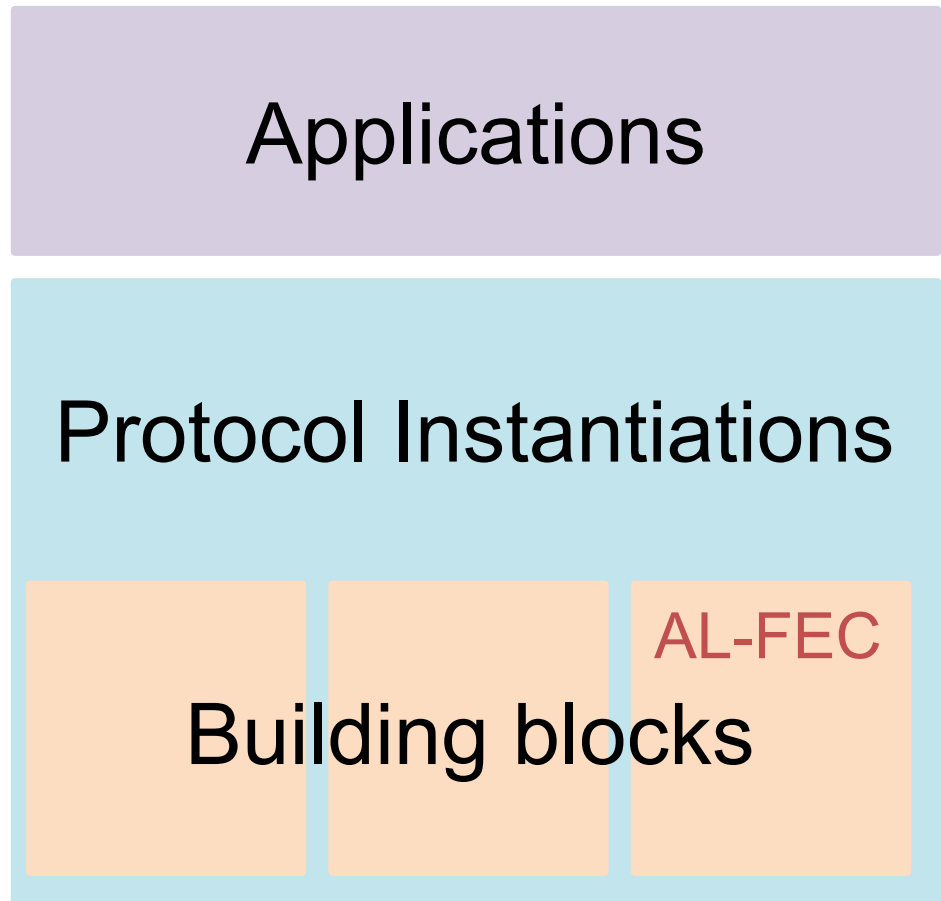
- must accommodate wide variety of **different needs**
 - from **small blocks** ($k=10s$) to **large blocks** ($k=10,000s$)
 - e.g. filecasting as in FLUTE/ALC requires large k values
 - encoding a very large object with Reed-Solomon over $GF(2^8)$ is quickly limited by the “coupon collector problem”
 - small rate codes are sometimes useful, but **$CR \geq 2/3$ is sufficient** most of the time, even in “digital fountain” applications like FLUTE/ALC
 - code parameters ($n; k$) are determined **dynamically**
 - **code creation time is critical**
 - e.g., no time to apply complex code optimization technics
 - Vandermonde matrix creation for Reed-Solomon is penalizing

Situation... (cont')

- AL-FEC is a small component of a much larger system
 - should be standardized independently
 - should be reusable across different protocols

Principle 1: “divide to conquer”

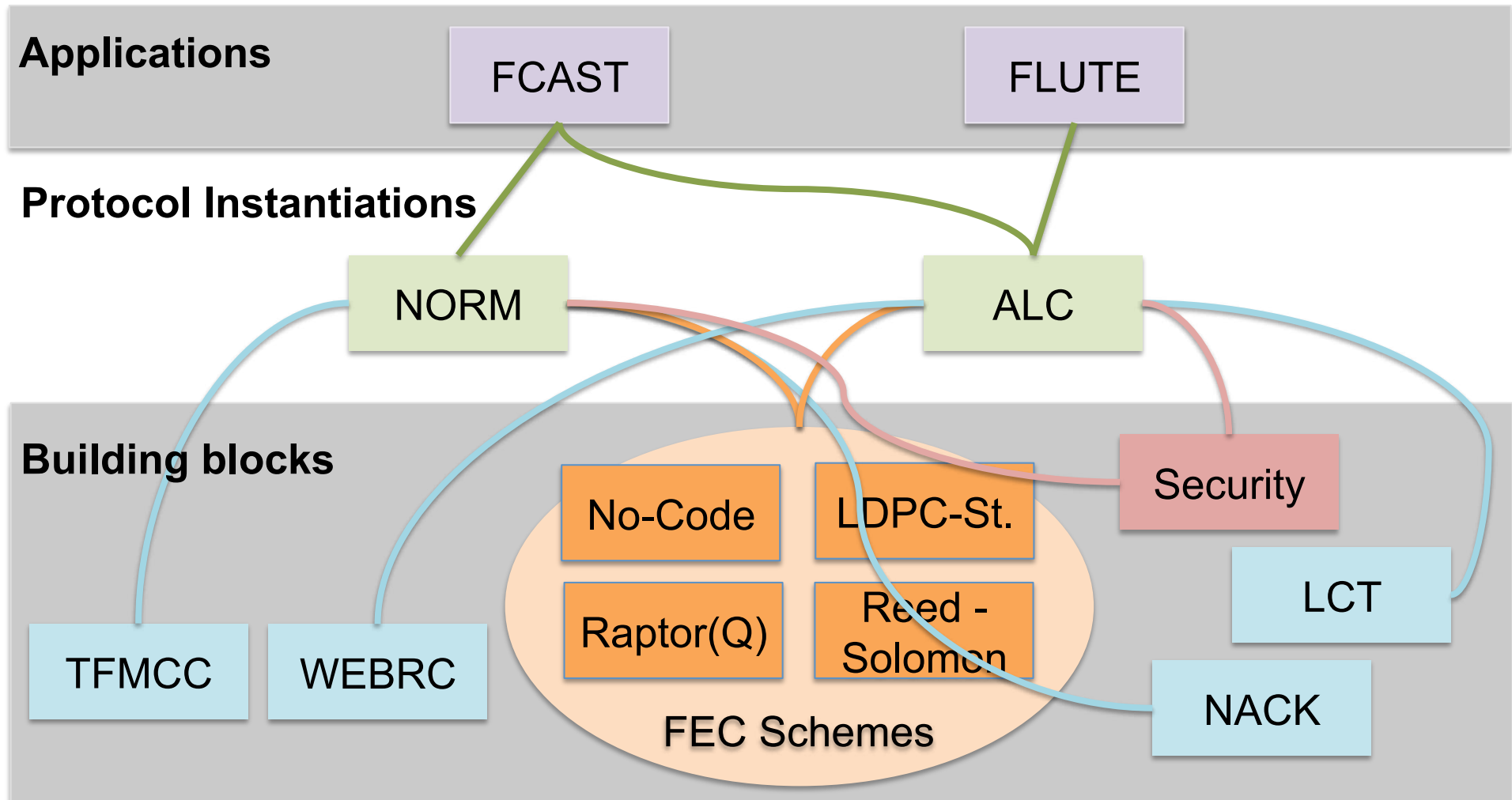
- define applications on top of protocols
 - if meaningful
 - offers specialization
- assemble BBs and create protocol inst.
 - protocol = { building blocks, specialized if needed }
 - working solution
- “building block” (BB) approach
 - focused and reusable components



Principle 1: “divide to conquer”... (cont’)

- ex. RMT

<http://www.ietf.org/proceedings/88/slides/slides-88-nwcr-1.pdf>



Principle 2: “specify FEC Schemes”

- the case of **fully-specified** FEC schemes
(the general case...)

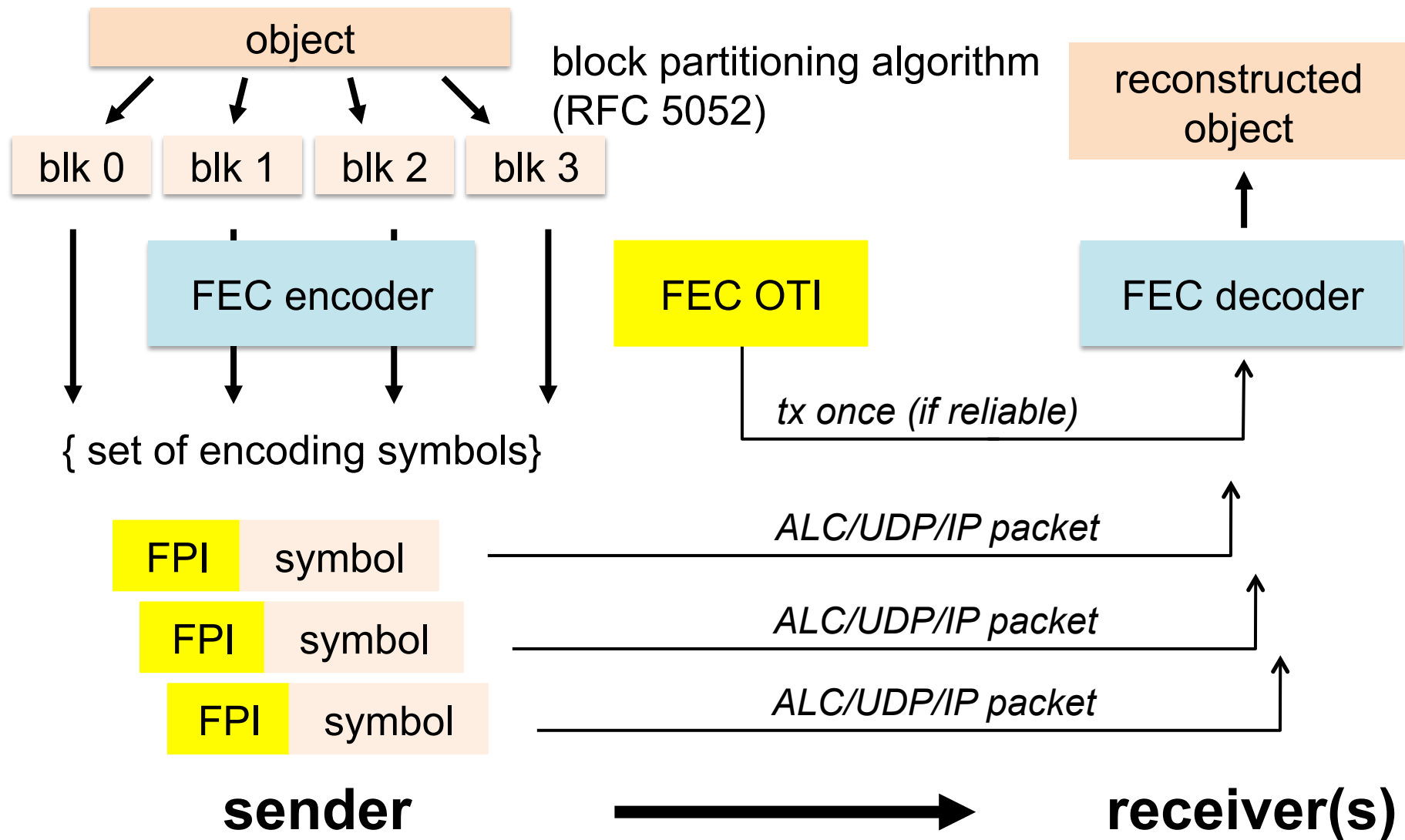
FEC Scheme
=
{identifier + code specifications + signaling }

- each scheme is uniquely **identified** (IANA registry)
 - FEC Encoding ID ex. Reed-Solomon is 5 for RMT
- all the **code details** are specified non ambiguously
 - interoperability is a MUST
- signaling enables encoder/decoder **synchronization**, for a given object transfer

Principle 2: “specify FEC Schemes” (cont’)

- more details on signaling...
 - some information is **sent once (reliably) per object transfer**
 - **FEC Object Transmission Information (FEC OTI)**
 - **info. for the object**
 - object length, parameters to partition it into blocks (if too large), symbol size
 - **info. for the FEC codec**
 - internal parameters, code rate (if needed)
 - some information is **sent in each packet**
 - **FEC Payload ID (FPI)**
 - **which symbol(s) does the packet contain?**
 - e.g. LDPC-Staircase ([RFC 5170](#))

Principle 2: “specify FEC Schemes” (cont’)



Principle 2: “specify FEC Schemes” (cont’)

- to know more about FEC Building Blocks
 - RMT \Rightarrow RFC 5052 <http://tools.ietf.org/html/rfc5052>
 - FECFRAME \Rightarrow RFC 6363 <http://tools.ietf.org/html/rfc6363>
 - in any case, signaling is essential to FEC

Focus on some NetWork Coding IRTF RG activities (nwcrgr)

A key aspect: what type of FEC code?

- NC use-cases ask for more flexibility than RMT/
FECFRAME do
 - re-encoding of coded packets, distributed encoding, sliding window encoding, etc.
 - blocks are really an issue in that case
- but not necessarily all potential use-cases
 - end-to-end NC still makes sense!
 - see Tetrys and Structured RLC

What type of FEC code... (cont')

- block encoding

RMT

FECFRAME

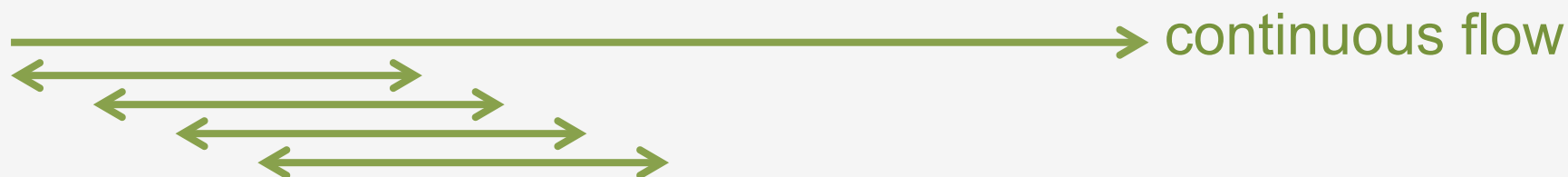
- fixed size, static encoding window



- sliding window encoding

NWCRG

- a fixed size encoding window slides over source symbols



- elastic window encoding

NWCRG

- a variable size encoding window slides over source symbols





convolutional codes

NWCRCG activities (non exhaustive list)

- see IRTF meeting proceedings

- IETF86 <http://www.ietf.org/proceedings/86/nwcrg.html>
- IETF87 <http://www.ietf.org/proceedings/87/nwcrg.html>
- IETF88 <http://www.ietf.org/proceedings/88/nwcrg.html>

- some contributions are on RLC codes

- various results on RLC codes and their application
- structured RLC codes  informatiques mathématiques
- distributed Reed-Solomon encoding
- Kodo RLC library (comment: beware of license)
- *not yet available, but in progress... Addition of RLC support in our <http://openfec.org> library*  informatiques mathématiques

NWCRG activities... (cont')

- others are on transport protocols for NC
 - coded TCP
 - improve TCP goodput by sending encoded symbols
 - TCP-IR (instant recovery)
 - a simple and pragmatic approach to solve some of the problems
 - Tetrys
 - DragonCast



Focus 1: Tetrys “on-the-fly” encoding protocol (representative of a broader class of solutions)

<http://www.ietf.org/proceedings/86/slides/slides-86-nwcrg-1.pdf>

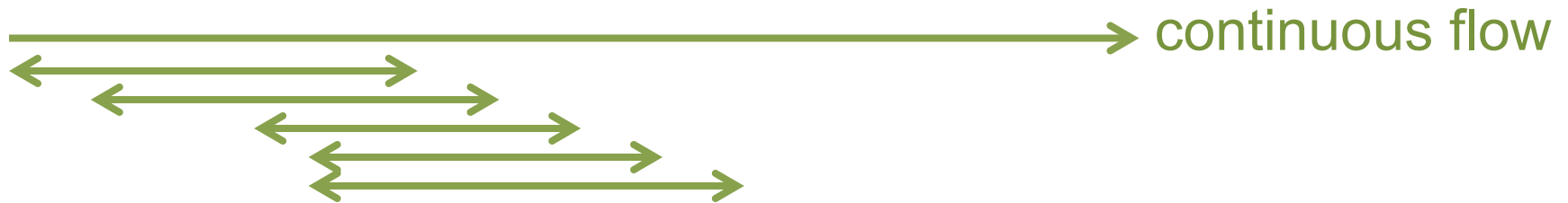
<http://websites.isae.fr/tetrys/>

P.U. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, V. Roca,
“On-the-fly erasure coding for real-time video applications”,
IEEE Transactions on Multimedia, Vol 13, Issue 4, August 2011.

Tetrys principles

- one technique, several ways to apply it

- **elastic encoding window** approach



- **unicast transmission with acknowledgments**

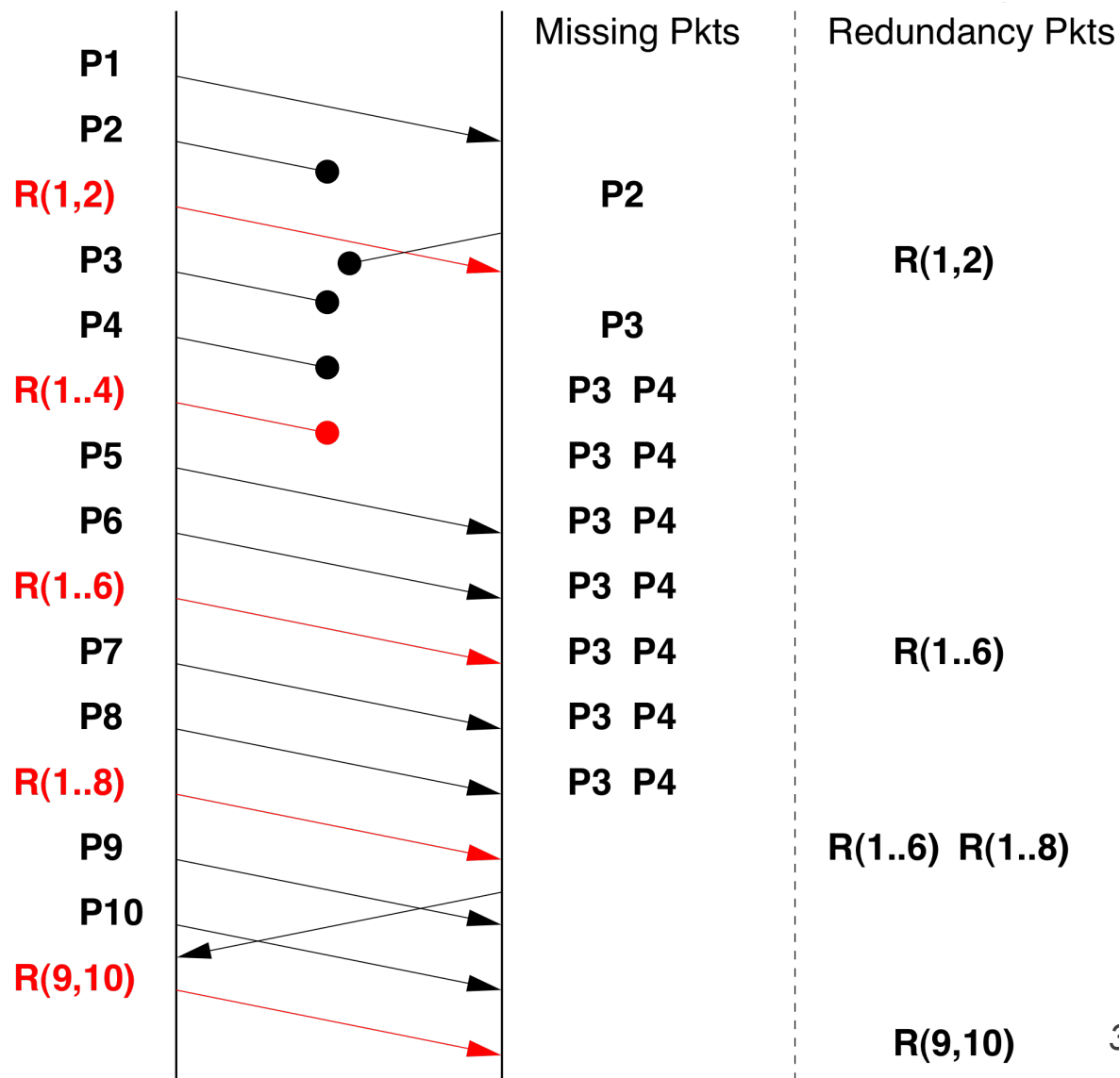
⇒ **encoding window contains any non-acknowledged packet**

- **key parameter: code rate**

Tetrys principles... (cont')

- **example:** ACKs enable sender to adjust encoding window

code rate 2/3



Tetrys principles... (cont')

- Tetrys can be turned to a sliding window (particular case)



○ Useful in case of unicast or multicast flows, without any acknowledgment

- Tetrys can be used in multi-path environments too

Tran-Thai, Tuan and Lochin, Emmanuel and Lacan, Jérôme
Online multipath convolutional coding for real-time transmission.
19th Int. Packet Video Workshop, May 2012.

Focus 2: Structured RLC codes: why? what for? how?

Motivations

- RLC are naturally random
 - it's easy, efficient, and flexible
- but there are incentives to have “structured” codes
 - **sparse codes** are **faster** to encode/decode
 - an order of magnitude difference, because:
 - fewer XOR and/or FF symbol operations
 - fast Iterative (IT) decoding works better
 - certain **structures** are extremely **efficient**
 - e.g., LDPC-Staircase [RFC5170] [WiMob13]
 - e.g., irregular LDPC codes perform the best with IT decoding

[WiMob13] V. Roca, M. Cunche, C. Thienot, J. Detchart, J. Lacan, **“RS + LDPC-Staircase Codes for the Erasure Channel: Standards, Usage and Performance”**, IEEE 9th Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2013.
<http://hal.inria.fr/hal-00850118/en/>

Goals of this work

- design codes that:
 - can be used as **sliding/elastic encoding window** (convolutional) and **block** codes
 - there are use-cases for each approach
 - can be used with encoding window/block sizes in **1-10,000s symbols** range
 - depends on the use-case
 - can be used as **small-rate** codes
 - can generate a large number of repair symbols
 - even if it's rarely useful

Goals of this work... (cont')

- have excellent erasure recovery **performance**
 - often a **complexity versus performance** tradeoff
 - it's good to be able to adjust it on a use-case basis
- enable **fast** encoding and decoding
 - sender and/or receiver can be an embedded device
- enable **compact and robust signaling**
 - transmitting the full encoding vector does not scale
 - prefer a function + index to identify the symbols/coefficients
 - can be a PRNG + seed
 - the function is known to both ends and the key is carried in the packet header

Goals of this work... (cont')

- focus **only** on use-cases that require **end-to-end encoding**
 - “end” means either “host” or “middlebox”, it’s the same
 - because it simplifies signaling
 - intermediate node re-encoding requires carrying the full encoding vectors which does not scale!
 - sure, it’s a subset of NWCRG candidate use-cases
 - but it’s well suited to Tetrisand also to FLUTE/ALC and FECFRAME

Idea 1: mix binary and non-binary

- mix binary and non binary
 - most equations are **sparse** and coefficients **binary**
 - a limited number of columns are **heavy** with non-binary coefficients (e.g., on $\text{GF}(2^8)$)
- there are good reasons for that:
 - sparseness is a key for high encoding/decoding speeds
 - density/non binary are good for recovery performances
 - gathering dense coefficients in columns (i.e. to certain symbols) is a key for high speed decoding [WiMob13]

[WiMob13] V. Roca, M. Cunche, C. Thienot, J. Detchart, J. Lacan, “**RS + LDPC-Staircase Codes for the Erasure Channel: Standards, Usage and Performance**”, IEEE 9th Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2013.
<http://hal.inria.fr/hal-00850118/en/>

Idea 1: mix bin and non-bin... (cont')

- block code example

- (sparse + non-bin. columns) only

$$H = \begin{bmatrix}
 \begin{array}{c|c|c}
 \text{source symbols} & & \text{repair symbols} \\
 s_0 & s_1 & \dots & s_{19} & s_{20} & \dots & s_{39} & r_0 & r_1 & r_2 & \dots
 \end{array} \\
 \hline
 \begin{array}{ccccccc|c}
 1 & 0 & \dots & 1 & \mathbf{29} & 0 & 0 & \dots & 1 & \mathbf{77} & 1 \\
 0 & 1 & \dots & 1 & \mathbf{62} & 1 & 0 & \dots & 0 & \mathbf{18} & 1 \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 \text{sparse} & & & & & \text{sparse} & & & & & \vdots \\
 \text{binary} & & & & & \text{binary} & & & & & 1 \\
 \text{part} & & & & & \text{part} & & & & &
 \end{array}
 \end{bmatrix}
 \rightarrow r_0 = s_0 + \dots s_{18} + \mathbf{29} * s_{19} + \dots s_{38} + \mathbf{77} * s_{39}$$

dense non-binary columns over $GF(2^8)$

Idea 2: add a structure

- technique 2: add a structure to the right part of H
 - we know that a “staircase” (A.K.A. double diagonal) is highly beneficial...

$$\mathbf{H} = \left[\begin{array}{ccccccccc|cccc}
 s_0 & s_1 & & & & & & & & s_{k-1} & r_0 & r_1 & \dots & r_{n-k+1} \\
 0 & 1 & 0 & 0 & 1 & \dots & \dots & \dots & \dots & 0 & 0 & 1 & & \\
 1 & 0 & 0 & 1 & 0 & \dots & \dots & \dots & \dots & 1 & 1 & 0 & & \\
 0 & 0 & 1 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & & \\
 & & \vdots & & & & & & & \vdots & & & & \\
 & & & & & & & & & & & & & \\
 0 & 1 & 0 & 1 & 0 & \dots & \dots & \dots & \dots & 0 & 1 & 0 & & \\
 & & & & & & & & & & & & &
 \end{array} \right]$$

- ... but when used in convolutional mode, **signaling** turns out to be **prohibitively complex**
 - believe me ;-)

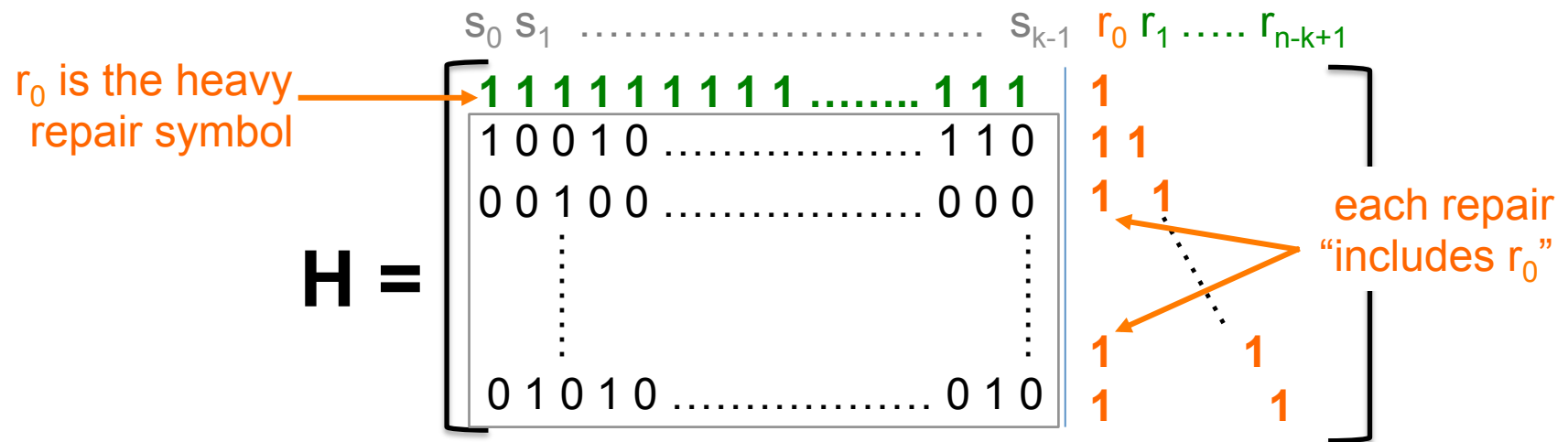
Idea 2: add a structure... (cont')

○ so we add a **single heavy row** and make **all repair symbols depend on it**

○ it's now quite simple, even when used in convolutional mode

- several sums will be transmitted (e.g., periodically), and it is sufficient to identify the last symbol of the sum in the signaling header

○ it's efficient (see later), at the price of extra XOR operations



○ NB: other ways to define heavy rows are feasible (e.g., with random coefficients over $GF(2^8)$...

Let's put ideas 1 and 2 together

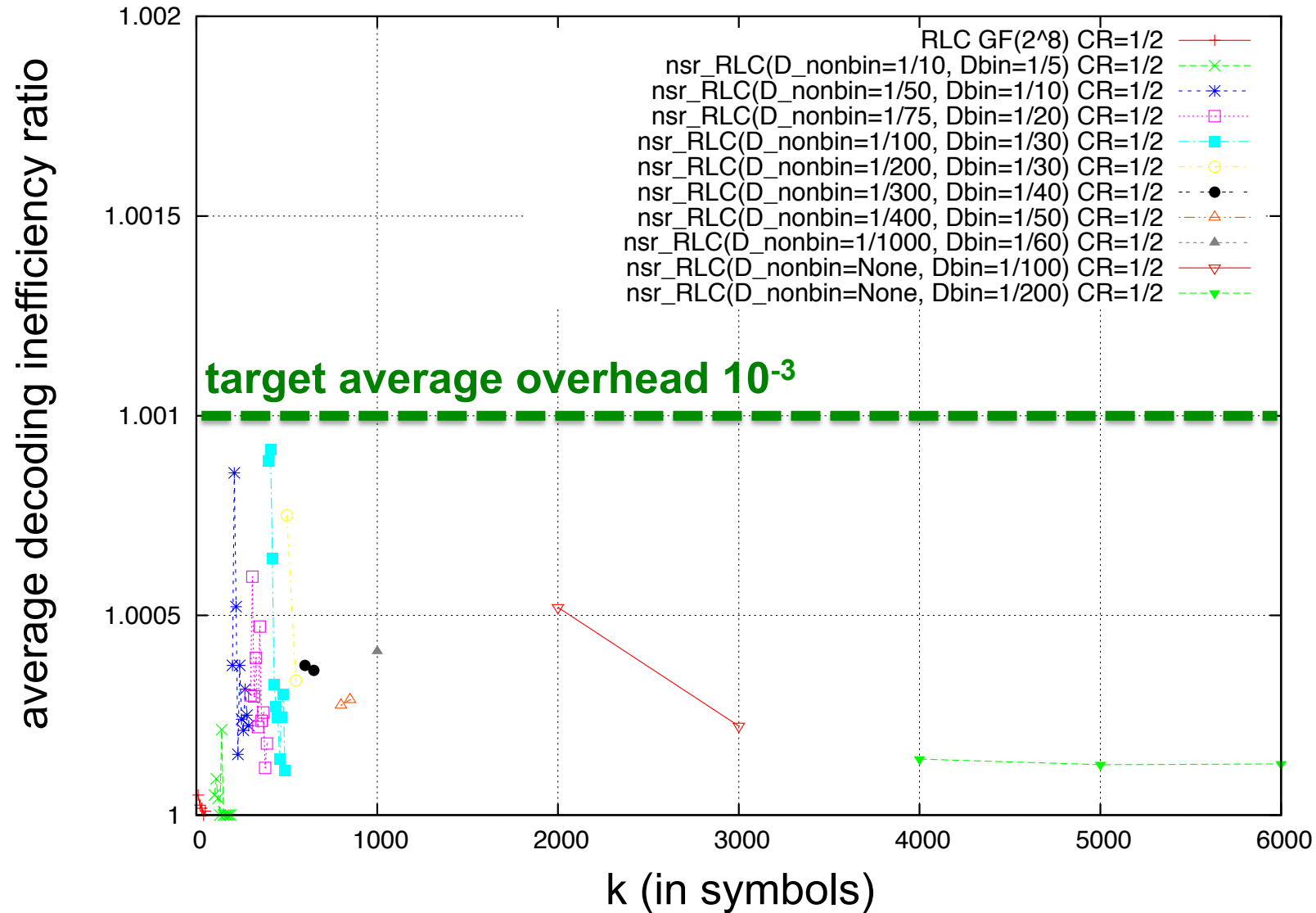
● 3 key parameters

- **k**: source block or current encoding window size
- **D_nonbin**: controls number of heavy non-binary columns
 - $D_nonbin = nb_non-binary_coeffs / k$
- **D_bin**: controls the density of the sparse sub-matrices
 - $D_bin = nb_1_coeffs / total_nb_coeffs_in_binary_submatrix$
- **{D_nonbin, D_bin}** depend on k and target max. overhead

$$H = \left[\begin{array}{cccccccccccc|cccccccc} s_0 & s_1 & & & & & & & & & & & & & & & & s_{k-1} & r_0 & r_1 & & & & & & & & & r_{n-k+1} \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & & & & & 1 & 1 & 1 & & & & & & & & & & \\ 1 & 0 & \dots & 1 & \mathbf{29} & 0 & 0 & \dots & 1 & \mathbf{77} & 0 & 1 & & & & & & & 1 & 1 & & & & & & & & & & & \\ 0 & 1 & \dots & 1 & \mathbf{62} & 1 & 0 & \dots & 0 & \mathbf{18} & 1 & 0 & & & & & & & 1 & & & & & & & & & & & & \\ \text{sparse} & & & & & & & & & & \text{sparse} & \\ \text{binary} & & & & & & & & & & \text{binary} & \\ \text{part} & & & & & & & & & & \text{part} & \\ \hline & \end{array} \right]$$

dense non-binary columns

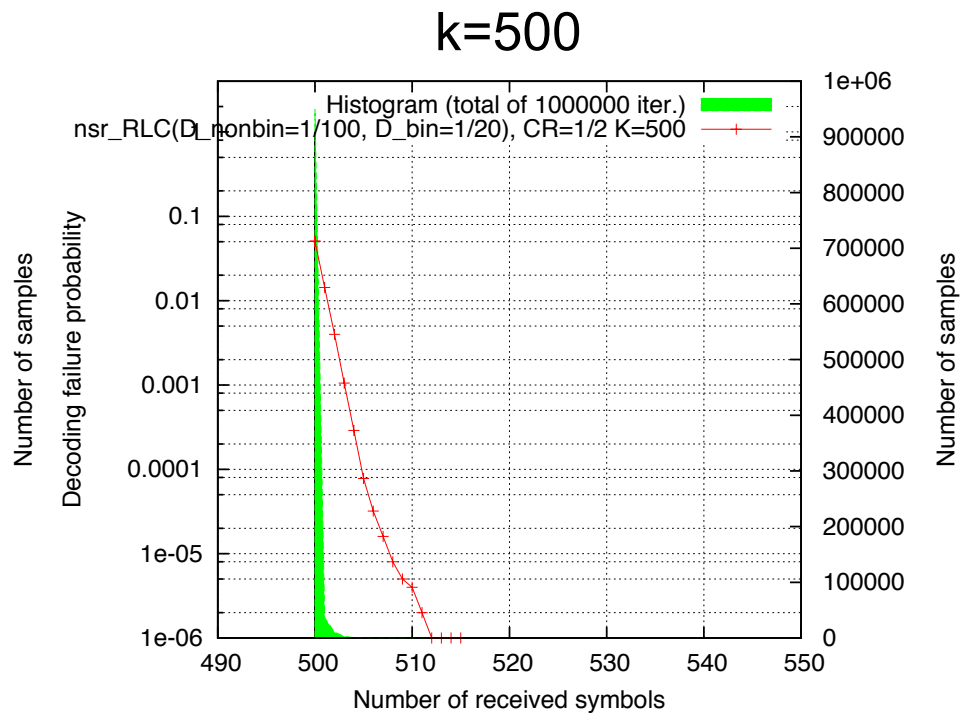
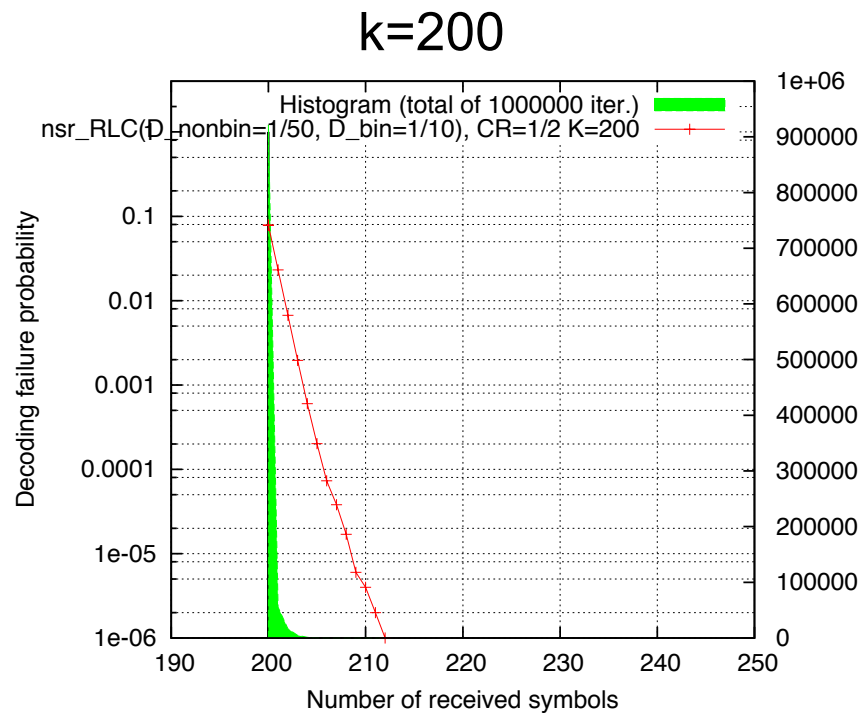
Preliminary results



NB: results are presented here as the concatenation of small curves...
In practice it will be a single curve for a single code...

Two close-ups

- decoding failure probability curves for $k=200, 500$
 - **no visible error floor** at 10^{-6} failure probability, which is excellent 😊



This is work under progress...

- many key questions remain

- what are the **performances** when used in sliding or elastic encoding window?
 - e.g. with Tetrys
- how **fast** is it?
 - e.g., compared to our optimized LDPC-Staircase/RS codecs
- how does it **scale** with k ?
 - e.g., compared to our optimized LDPC-Staircase codec
- define **signaling** aspects
 - FEC Payload ID (in each packet sent)
 - FEC Object Transmission Information (per object/session)

- want to know more?

<http://www.ietf.org/proceedings/88/slides/slides-88-nwcrg-2.pdf>

Conclusions

Conclusions

- there are plenty of opportunities to contribute to IETF/IRTF
 - it's open and academic-friendly
 - compared to other SDOs
 - the directions taken by the NWCRG group depend on individuals
 - there's no a-priori forbidden topics
 - if it fits within the “Internet” (in its broader meaning), it's okay
 - but it **MUST NOT break the Internet**
 - do not create flows/protocols that do not react in front of congestion signals
 - “TCP friendliness” (to some extent) is unavoidable