Tetrys, a Patent-Free Network Coding Protocol - Update

http://tools.ietf.org/html/draft-detchart-nwcrg-tetrys-02

Jonathan Detchart, Jérôme Lacan, Emmanuel Lochin
ISAE
Vincent Roca
INRIA

IETF93
NWCRG
July 21th, 2015
Prague

Note Well

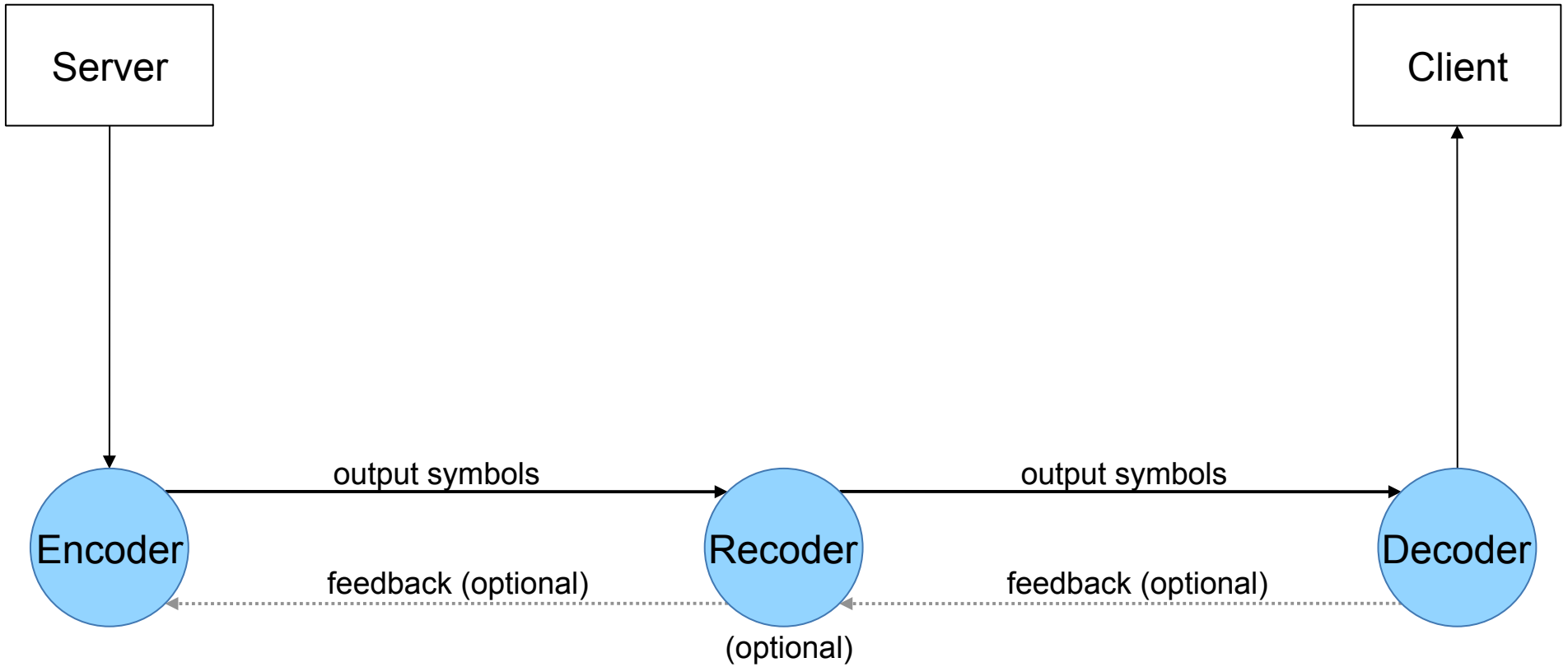
- We, the authors, didn't try to patent any of the material included in this presentation
- We, the authors, are not reasonably aware of patents on the subject that may be applied for by our employers
- If you believe some aspects of this presentation and its corresponding draft may infringe IPR you are aware of, then please fill an IPR disclosure and let us know

http://irtf.org/ipr

What's new?

- A deterministic method for recoding in the intermediate nodes without transmitting the coefficients
- Some fields in the encoding vectors have been reordered and a new one to signal the use of variable size symbols has been added
- A proposition for the multicast use case with Tetrys using an elastic encoding window

A simple use case



Recoding with Tetrys

use the received coded symbols to generate deterministically new coded symbols without decoding operation

Recoding without decoding nor sending explicitly the new coding vector

- In the intermediate nodes, we have (R_1,..., R_t), a set of t received source and coded symbols
  - these symbols have been produced from the set (S_1,..., S_k) of k source symbols

- Problem: compute a new coded symbol R'= $\sum_i S_i * a_i$, where the coding coefficients $(a_1, \ldots, a_k)$ are generated as usual, from:
  - the coded symbol ID of R'
  - a Coding Coefficient Generator

  ⇒ doing so avoids having to transmit the coding coefficients, only the new coded symbol ID is transmitted ☺

Recoding without decoding nor sending explicitly the new coding vector (con't)

- To that purpose, in the general case :

  – if we have: $(R_1, \ldots, R_t) = (S_1, \ldots, S_k) * G$, where G is of rank k, then extract a invertible k*k submatrix G' of G such that $(R_{\sigma(1)}, \ldots, R_{\sigma(k)}) = (S_1, \ldots, S_k) * G'$ try to compute $M=G'^{-1}$ such that :

  $$(S_1, \ldots, S_k) = (R_{\sigma(1)}, \ldots, R_{\sigma(k)}) * M$$

  ⇒ it usually succeeds, otherwise wait receiving a few more symbols

  – Then $\mathrm{R}' = \sum_i S_i * a_i = \sum_i R_{\sigma(i)} * v_i$ where $(v_1, \ldots, v_k) = (a_1, \ldots, a_k) * transpose(M)$

Recoding without decoding nor sending explicitly the new coding vector (con't)

- G is sometimes rather sparse (e.g., when receiving many source symbols and/or when the generator generates sparse linear combinations)

  - simplifies the matrix M computation

  - can allow iterative operations (like in LDPC decoding) to generate new coded symbols

- Computing M does not always succeed

  - e.g., if t < k or if the linear system is singular

  - then try to generate coded symbols from a subset of $(S_1, \ldots, S_k)$

  - else wait a few more symbols

  it rarely happens in practice given the way Tetrys works

Recoding complexity

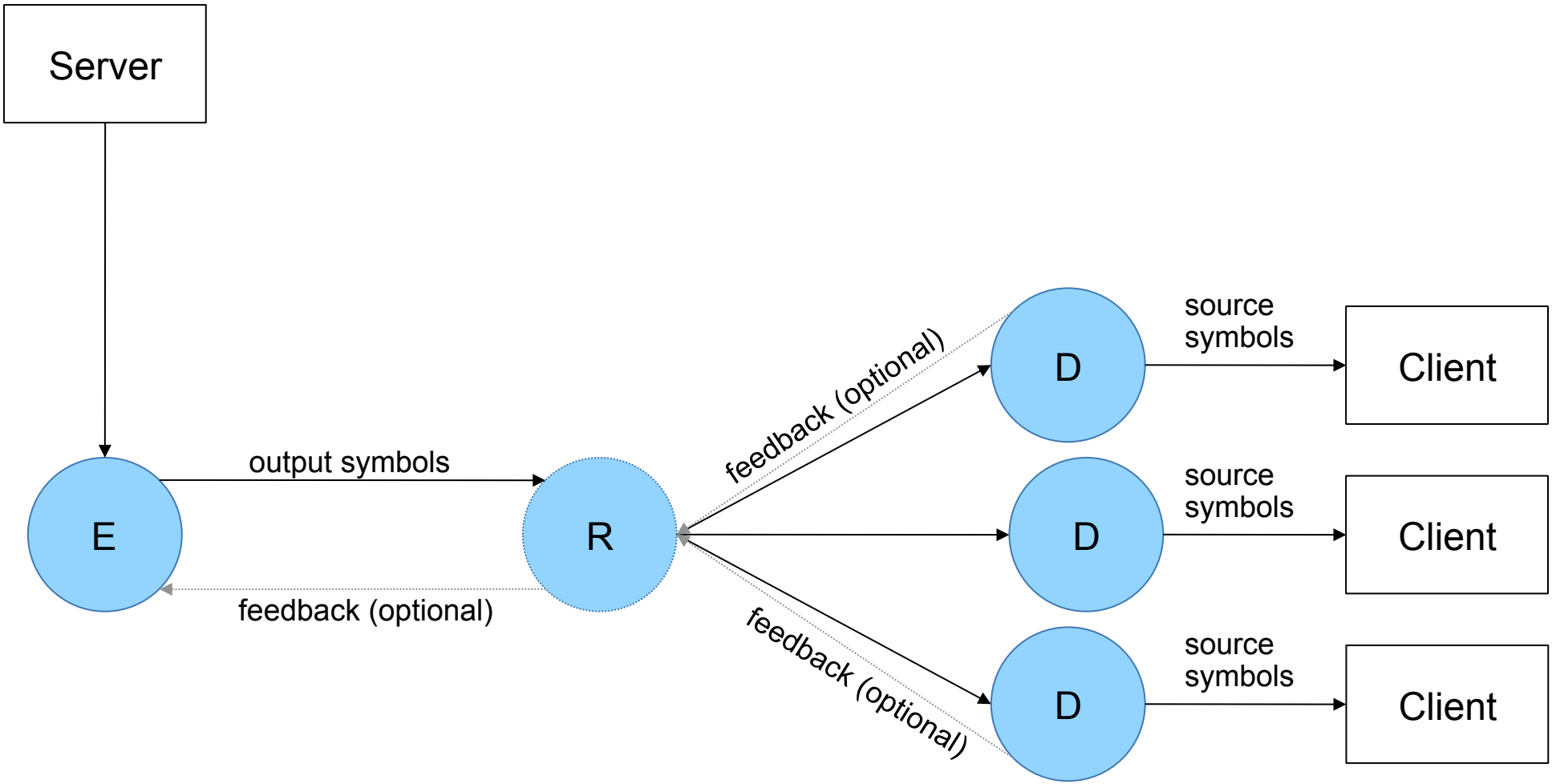
- To generate a new coded symbol from k received coded symbols:
  - Recoding in Tetrys:
    - *k*k matrix inversion:* k^3
    - Matrix-vector product: k^2
    - Linear combination of the received symbols: k*pkt_size
  - Recoding in RLNC:
    - Matrix-vector product: k^2
    - Linear combination of the received symbols: k*pkt_size

➔ In most situations, pkt_size >> k and therefore **linear combinations** of symbols **take more time** than matrix inversion

The Encoding Vectors

- Sent in the coded packets
- They allow to:
  - Signal the use of variable size source symbols
  - Carry the coefficients used to generate the coded symbol
  - Identify the source symbols used to generate the coded symbol
- Yet this information is optional
  - Depends on the coding scheme

Multicast Use Case



Multicast

- With feedback:
  - Need to define a strategy to remove a source symbol
    - When it is decoded or received by all the receivers
    - When it is decoded or received by a percentage of the receivers
- Without feedback:
  - The Elastic Encoding Window grows up to a limit, then the oldest source symbols are removed

    Simple but not fully reliable

Conclusion

- We proposed Tetrys, a *flexible* network coding *Protocol*
- Recoding is now possible at intermediate nodes
- Multicast is also possible using an Elastic Encoding Window or just a Sliding Window

- Tetrys is **independent** of the coding scheme!

Thank you !

Authors:
jonathan.detchart@isae.fr
jerome.lacan@isae.fr
emmanuel.lochin@isae.fr
vincent.roca@inria.fr

Draft:
http://tools.ietf.org/html/draft-detchart-nwcrg-tetrys-02