

FEC and Network Coding for dummies

vincent.roca@inria.fr

NWCRG Nov. 5th, 2018, IETF 103, Bangkok

Motivation

- **comment made during IETF102 NWCRG**

“99% of people using TCP don't know how it works but think the opposite. We need a "network coding for dummies" document. It's really important to have people think they understand how NC works for them to adopt the technology.”

- **what are the most basic yet essential messages to make people believe they understand?**
- **keep it small**
 - ✓ **it's not a tutorial**

Idea 1-

- **“We focus on networks where a packet either arrives or is lost”**
 - we're not at PHY-layer, we are above in the protocol stack and potential bit errors have either been fixed or the packet dropped

Idea 2-

- “Encoding consists in **adding redundancy** (i.e., repair packets) to the flow
- decoding consists in **using redundancy** (i.e., repair packets) to recover from packet losses”

Idea 3-

- **“Math is not an obstacle to understand FEC and NC”**
 - it's essentially a matter of **linear combination and linear system resolution** (e.g., via basic Gaussian elimination)
 - details (e.g., computations in a certain Finite Field) can be complex, but mastering them is not required

Idea 4-

- **“There are roughly two categories of FEC codes: block codes and sliding window codes”**
 - block: segment the packet flow into blocks and apply FEC encoding per block, independently
 - sliding window: an encoding window slides progressively over the packet flow, the encoder computes a linear combination of packets in this encoding window

Idea 5-

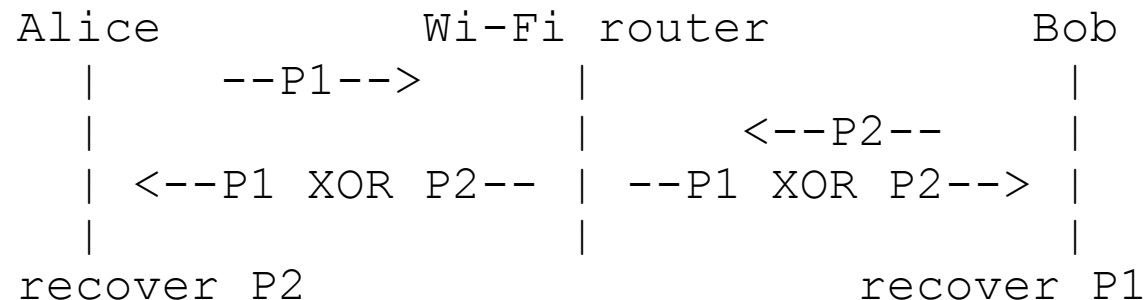
- **“Block FEC codes are great for bulk, non real-time traffic, sliding window FEC codes are great for real-time traffic”**
 - ... because splitting the application flow into blocks delays the moment when repair packets can be generated!

Idea 6-

- “Some codes are restricted to a **single encoder** (e.g., sender) and **single decoder** (e.g., receiver)”
 - usually called FEC
- “Other codes can be used within **intermediate nodes** (i.e., multiple encoders)”
 - usually called Network Coding (NC)

Idea 7-

- **“With NC, network equipments can perform FEC encoding to improve network usage”**
 - trivial example where a network equipment could reduce traffic (it sends a single “P1 XOR P2” packet instead of sending both P1 and P2)



(Figure from Zverev Mihail, Ikerlan)

Idea 8-

- **“One can use FEC and NC in a congestion friendly manner”**
 - only stupid persons will further overload a congested network with even more redundant traffic in the hope it may help!

Using it...

- already in github's README.md
 - rg-materials repo
 - swif-codec repo
 - ... and wherever it makes sense
- does it help? Feedback is welcome

The screenshot shows the GitHub repository page for 'irtf-nwcrgrg-materials'. The repository has 48 commits, 1 branch, 0 releases, and 2 contributors. The latest commit is by 'vincent-grenoble' (upload revision of slides) 7 hours ago. The commit history includes:

Commit	Message	Time
ietf102-2018-07	adding ietf102 slides	3 days ago
ietf103-2018-11	upload revision of slides	7 hours ago
interim-2018-09	interim minutes update	a month ago
CONTRIBUTING.md	added ietf103	2 months ago
README.md	Update README.md	2 days ago

The README.md file content is visible below the commit history:

rg-materials

Repository for the NWCRG meeting materials (agenda, presentations, minutes) used during IETF meetings and Interim meetings. Related Internet-drafts are accessible from the [NWCRG datatracker web site](#).

Note that, in addition to this repository:

- meeting material is always uploaded at [\[https://datatracker.ietf.org/meeting/materials/\]](https://datatracker.ietf.org/meeting/materials/) prior to each IETF meetings;
- past meeting material for all IETF/IRTF groups is archived at [\[https://www.ietf.org/how/meetings/past/\]](https://www.ietf.org/how/meetings/past/);
- past meeting material related to NWCRG is easily accessible at [\[https://datatracker.ietf.org/rg/nwcrg/meetings/\]](https://datatracker.ietf.org/rg/nwcrg/meetings/).

Eight simple ideas to start understanding FEC and Network Coding

Idea 1-

"We focus on networks where a packet either arrives or is lost"

- we're not at PHY-layer, we are above in the protocol stack and potential bit errors have either been fixed or the packet dropped

Idea 2-