Netzwerkkodierung in Theorie und Praxis

Praktische Anwendungen der Netzwerkkodierung

Professor Dr.-Ing. Dr. h.c. Frank H.P. Fitzek

M.Sc. Juan Cabrera

Deutsche Telekom Chair of Communication Networks (ComNets)

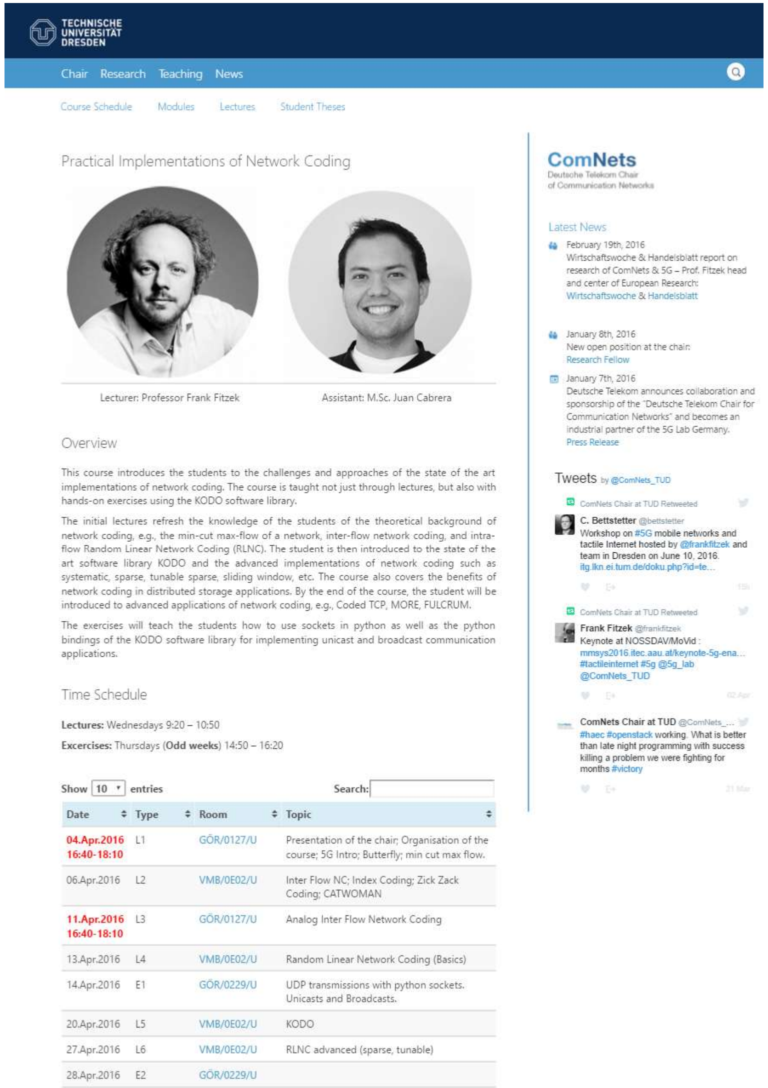
Netzwerkkodierungstheorie

Professor Dr.-Ing. Eduard Jorswieck

Dipl.-Ing. Johannes Richter

Theoretische Nachrichtentechnik

Lecture / Exercise Dates - tinyurl.com/zooafld



Chair Research Teaching News

Course Schedule Modules Lectures Student Theses

Practical Implementations of Network Coding

Lecturer: Professor Frank Fitzek

Assistant: M.Sc. Juan Cabrera

Overview

This course introduces the students to the challenges and approaches of the state of the art implementations of network coding. The course is taught not just through lectures, but also with hands-on exercises using the KODO software library.

The initial lectures refresh the knowledge of the students of the theoretical background of network coding, e.g., the min-cut max-flow of a network, inter-flow network coding, and intra-flow Random Linear Network Coding (RLNC). The student is then introduced to the state of the art software library KODO and the advanced implementations of network coding such as systematic, sparse, tunable sparse, sliding window, etc. The course also covers the benefits of network coding in distributed storage applications. By the end of the course, the student will be introduced to advanced applications of network coding, e.g., Coded TCP, MORE, FULCRUM.

The exercises will teach the students how to use sockets in python as well as the python bindings of the KODO software library for implementing unicast and broadcast communication applications.

Time Schedule

Lectures: Wednesdays 9:20 – 10:50

Excercises: Thursdays (**Odd weeks**) 14:50 – 16:20

Show 10 entries Search:

Date	Type	Room	Topic
04.Apr.2016 16:40-18:10	L1	GÖR/0127/U	Presentation of the chair; Organisation of the course; 5G Intro; Butterfly; min cut max flow.
06.Apr.2016	L2	VMB/0E02/U	Inter Flow NC; Index Coding; Zick Zack Coding; CATWOMAN
11.Apr.2016 16:40-18:10	L3	GÖR/0127/U	Analog Inter Flow Network Coding
13.Apr.2016	L4	VMB/0E02/U	Random Linear Network Coding (Basics)
14.Apr.2016	E1	GÖR/0229/U	UDP transmissions with python sockets. Unicasts and Broadcasts.
20.Apr.2016	L5	VMB/0E02/U	KODO
27.Apr.2016	L6	VMB/0E02/U	RLNC advanced (sparse, tunable)
28.Apr.2016	E2	GÖR/0229/U	

ComNets
Deutsche Telekom Chair of Communication Networks

Latest News

February 19th, 2016
Wirtschaftswoche & Handelsblatt report on research of ComNets & 5G – Prof. Fitzek head and center of European Research: Wirtschaftswoche & Handelsblatt

January 8th, 2016
New open position at the chair: Research Fellow

January 7th, 2016
Deutsche Telekom announces collaboration and sponsorship of the "Deutsche Telekom Chair for Communication Networks" and becomes an industrial partner of the 5G Lab Germany. Press Release

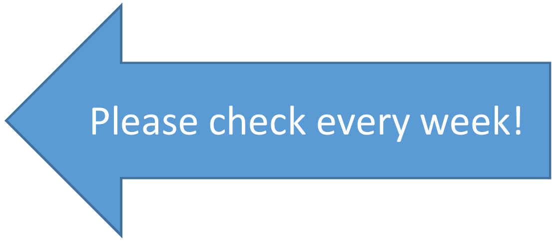
Tweets by @ComNets_TUD

ComNets Chair at TUD Retweeted
C. Bettstetter @bettstetter
Workshop on #5G mobile networks and tactile Internet hosted by @frankfitzek and team in Dresden on June 10, 2016. itg.lkn.ei.tum.de/doku.php?id=te...

ComNets Chair at TUD Retweeted
Frank Fitzek @frankfitzek
Keynote at NOSSDAV/MoVid : mmsys2016.itec.aau.at/keynote-5g-ena... #tactileinternet #5g @5g_lab @ComNets_TUD

ComNets Chair at TUD @ComNets_...
#haec #openstack working. What is better than late night programming with success killing a problem we were fighting for months #victory

- Here all information for the lecture and the exercise can be found.
- Slides
- Links
  - Steinwurf
  - Python
  - KODOMARK (google play)



TECHNISCHE UNIVERSITÄT DRESDEN

A Practical Guide to RLNC Libraries

Where is network coding located?

Application Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

http://www.youtube.com/watch?v=OnqGO7AWwxc

Where is network coding located?

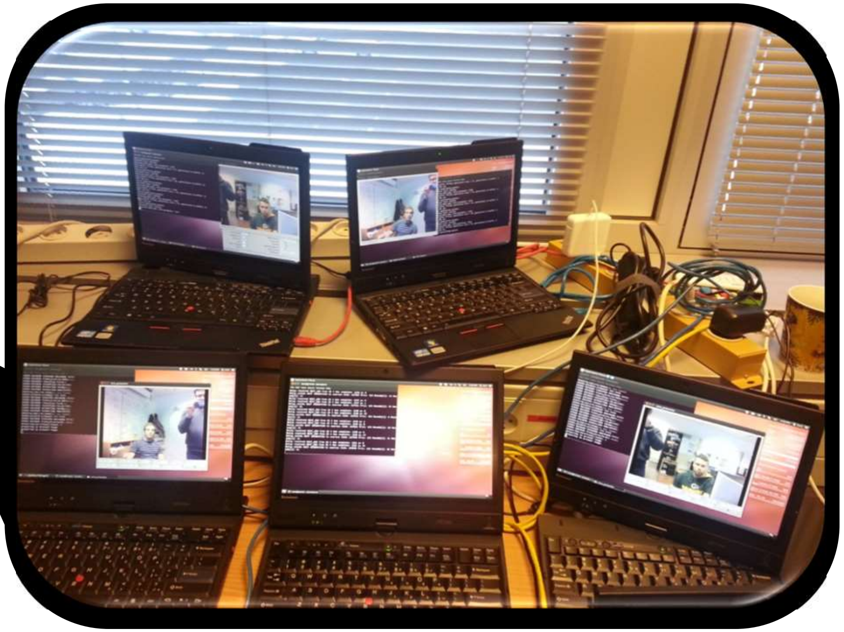
Application Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer



Where is network coding located?

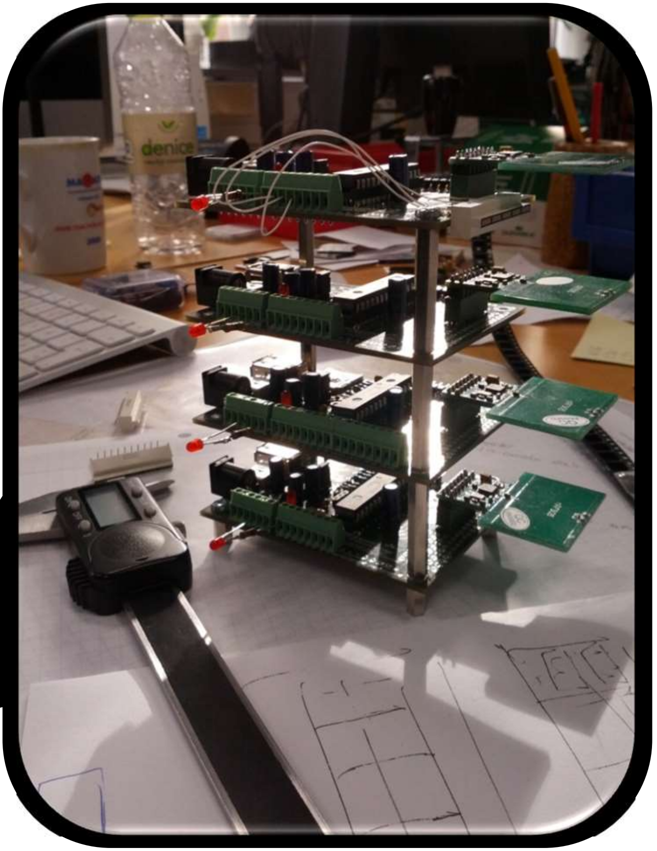
Application Layer

Transport Layer

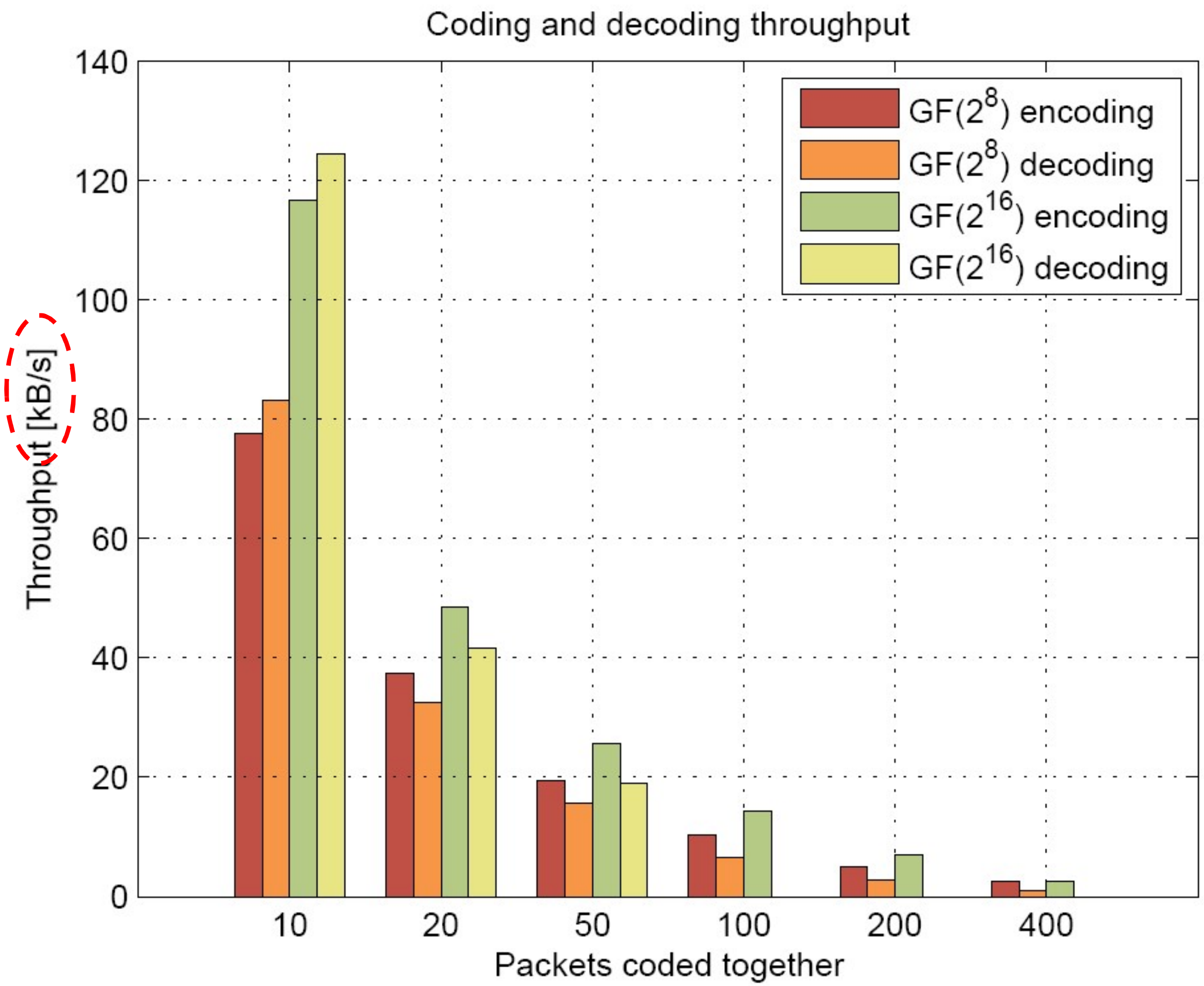
Network Layer

Data Link Layer

Physical Layer



S60 Implementation RLNC (2007)



Pre-allocated memory, generated the encoding vectors, so that we only had the raw encoding

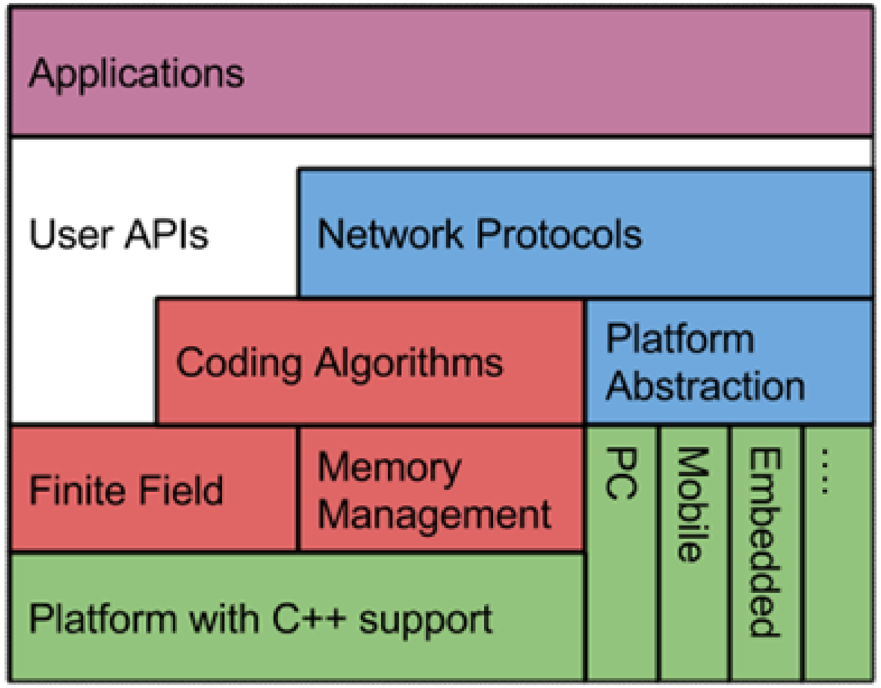
Key Technologies to Speed Up

- New software design
- Right choice of G and F
  - Binary case results in low complexity
- Hardware implementation
  - Dedicated hardware (OPENGL, SIMD)
  - Multi core / Many core (HAEC)
  - Kernel
- Sparse coding & Systematic coding
- Optimal Prime Fields (OPF), e.g., 232-5

TECHNISCHE UNIVERSITÄT DRESDEN

Software Library

Kodo: Software Implementation



- Software library for Network Coding
- Software library for Network Protocols
- Fully-tested Software
- Build System for several platforms
- In use by Customers

KODO: Shortens TTM

- Implemented Features
  - Recoding
  - Systematic coding
  - On-the-fly coding
  - Partial decoding
  - Real-time adjustable density
  - Symbol pruning
  - File encoder
  - Zero copy API
  - Object pooling
  - Hardware optimized
  - Variable symbol length

- Platforms

- Continuous Integration
  - build on every commit
  - buildbot.steinwurf.dk

Commercial Library Benchmarking

- Jerasure 1.2 by James Plank
- Jerasure 2.0 by James Plank
- OpenFEC by INRIA
- ISA-L by INTEL
- KODO by Steinwurf

- The intention is to make a fair comparison among them and start collaborative research on this topic!

TECHNISCHE UNIVERSITÄT DRESDEN

Feature List

Library Capabilities	Kodo	Jerasure 1.2	Jerasure 2.0	ISA-L	Open FEC
Reed-Solomon Codes Supported	X	X	X	X	X
Network Coding Supported	X				
Updated with Novel Code Structures	X				(X)
Continuous Testing and Support	X				
Continuous Optimization of Algorithms	X				
Automatic Adaptation to CPU Features	X				
OS Support	iOS			FreeBSD	
Compiler Support	GCC, Clang, MS VS	?	GCC	GCC	?
Date of Last Release	1/2014	8/2008 12/2011[x]	1/2014	11/2013	4/2012
Hardware Acceleration on Intel Chipsets	SSSE3, CLMUL, AVX2		SSSE3	SSSE3, CLMUL	SSE
Hardware Acceleration on ARM chipsets	NEON				
Multi-core support	X				
Simulation support	Internal, NS3				

TECHNISCHE UNIVERSITÄT DRESDEN

Comparison with State of the Art

- Coding Speed [MB/s] for 1 MB per data segment

Industry trend

F=GF(2^8) P=1MB	Kodo 17 MT (sparse=0.5)	Kodo 17 (sparse=0.5)	ISA-L	Jerasure 2.0	OpenFEC
G=8 (12)	3096/2980	3096/2980	2255/2635	1250/1365	353/292
G=9 (13)	2542/2559	2752/2898	1961/2252	1096/1185	305/264
G=10 (15)	2136/2227	2025/2126	1724/1796	997/1072	285/245
G=16 (24)	1807/1496	1264/1239	1075/1180	628/644	179/160
G=30 (45)	950/647	672/513	266/271	349/361	96/90
G=60 (90)	594/329	359/256	123/122	184/184	48/46
G=100 (150)	383/209	226/159	74/73	111/111	29/28
G=150 (225)	266/141	153/107	47/46	74/74	19/19

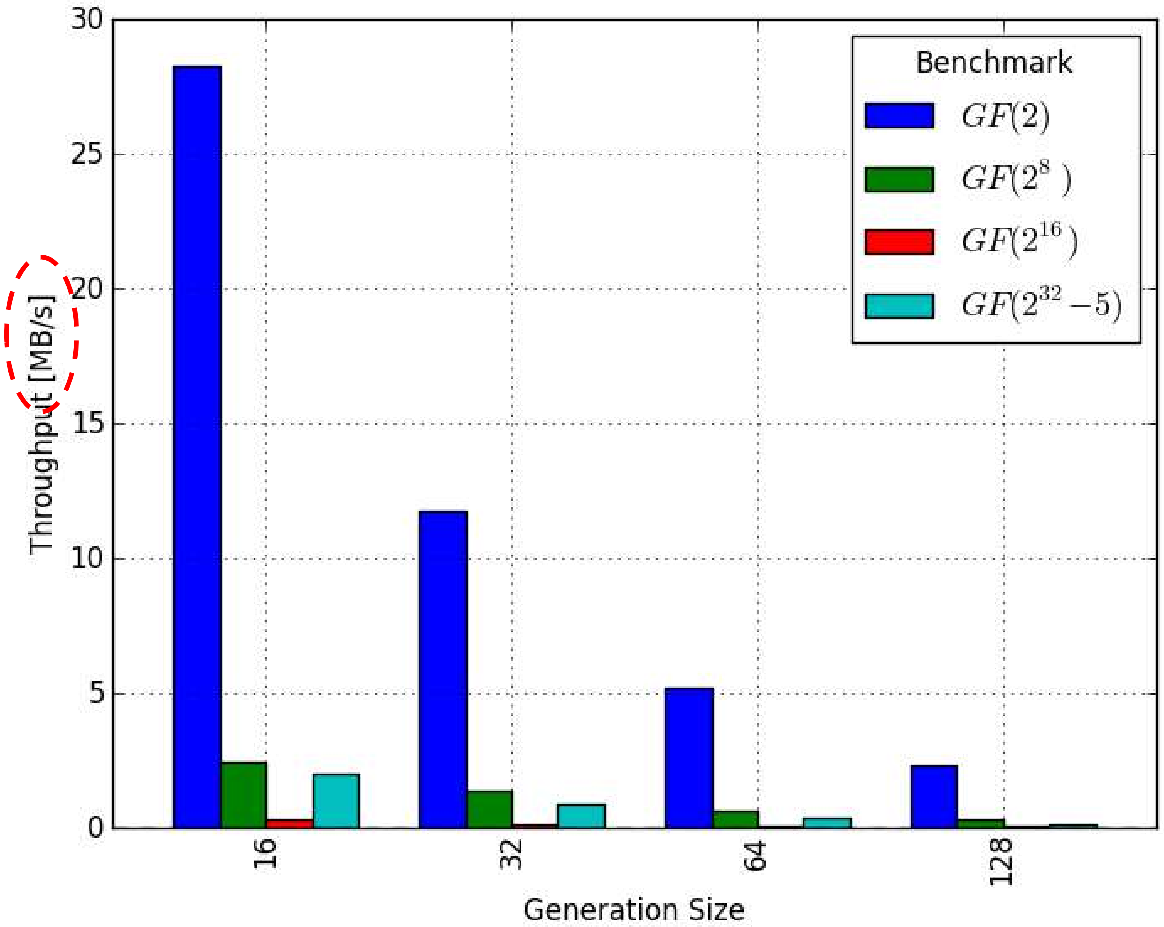
RLNC (rich feature set) RS

RS use G times more memory than RLNC for data segment recovery

Measured on Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz

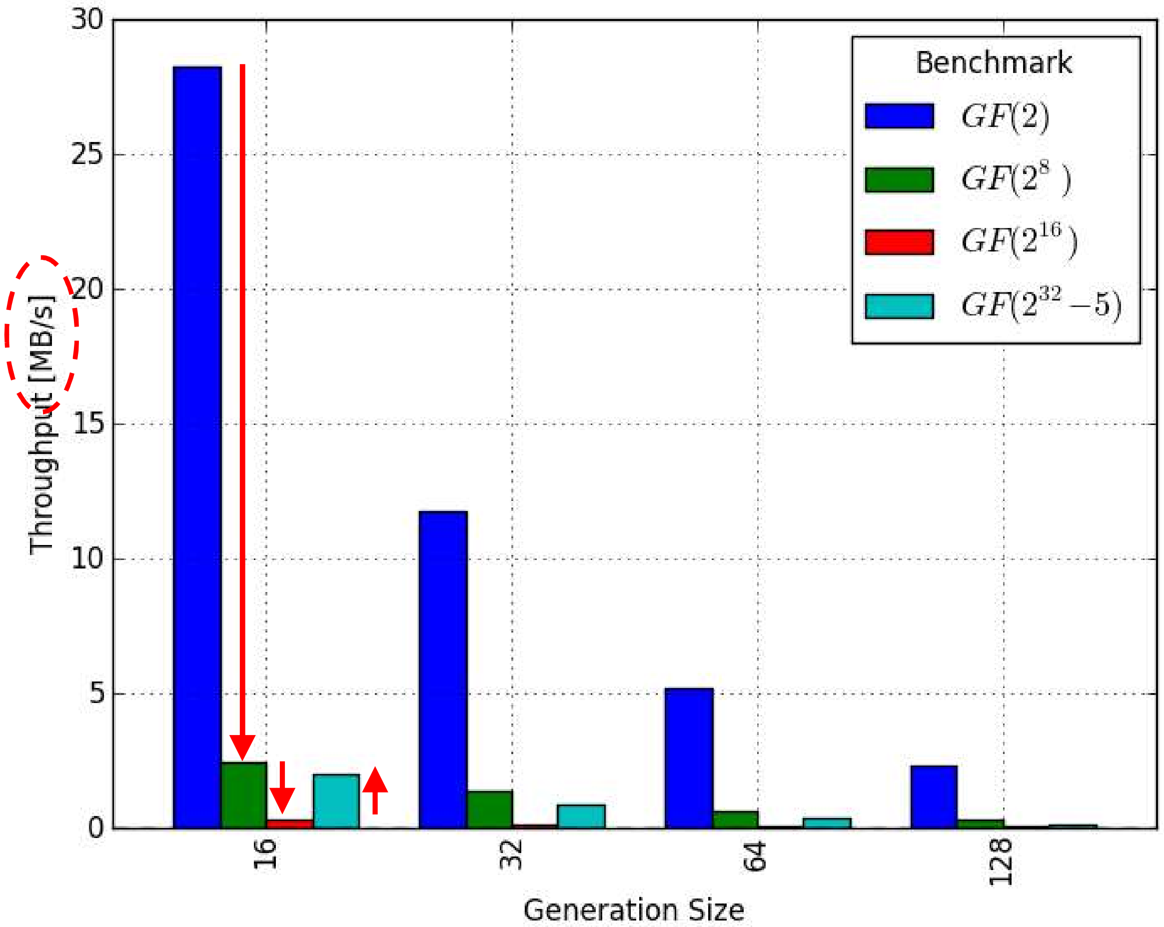
Raspberry Pi (2013)

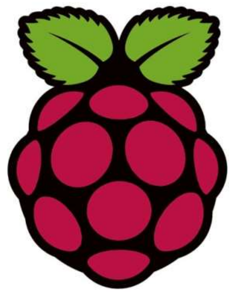
Hardware: 700 MHz CPU – KODO: full coding



Raspberry Pi (2013)

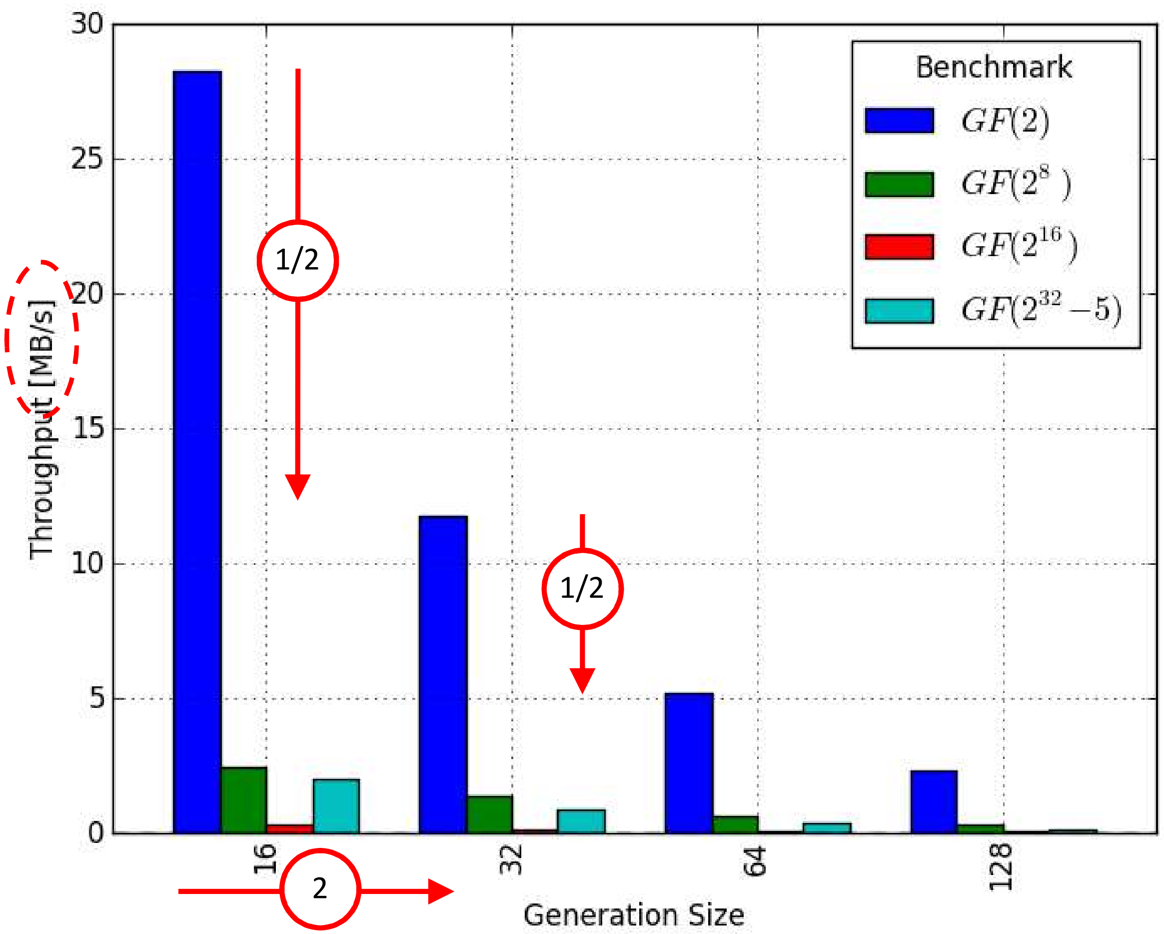
Hardware: 700 MHz CPU – KODO: full coding





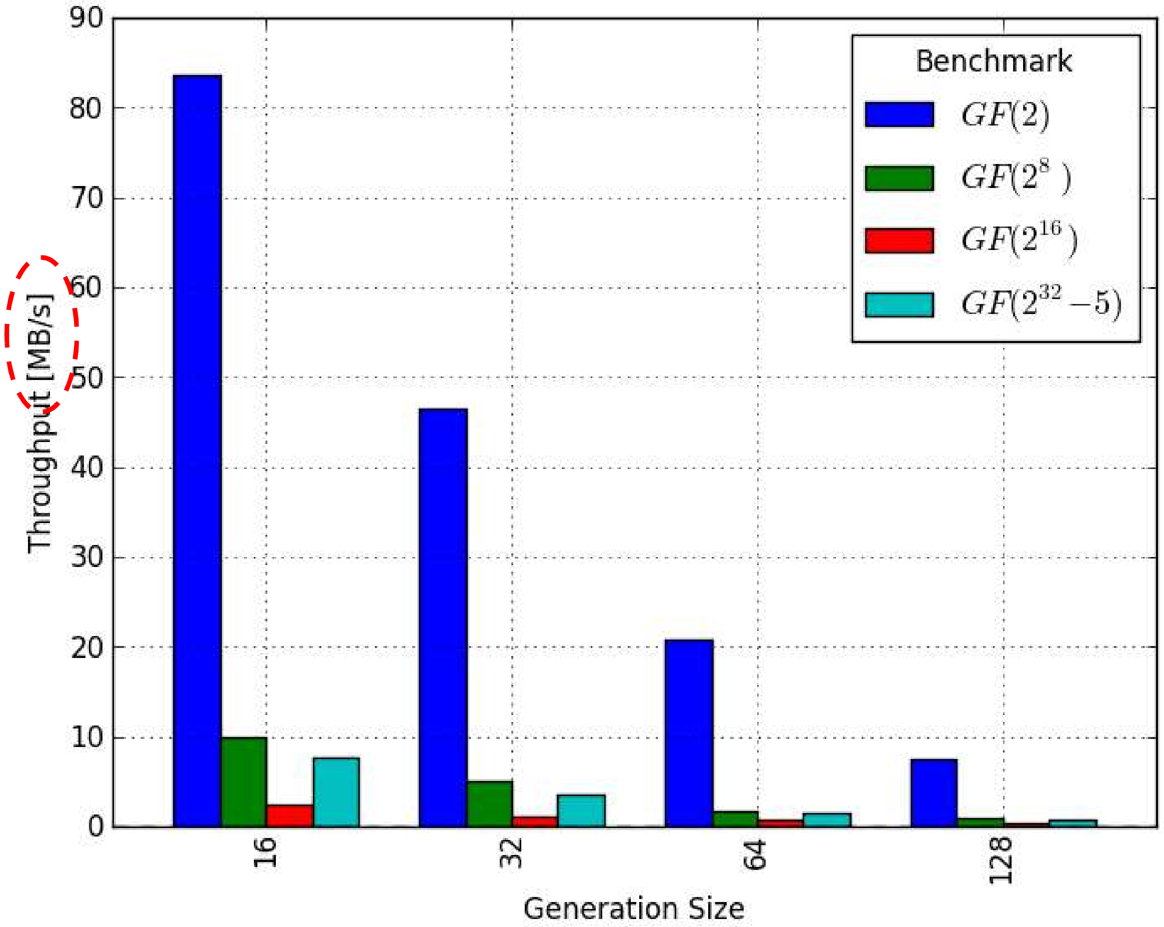
Raspberry Pi (2013)

Hardware: 700 MHz CPU – KODO: full coding



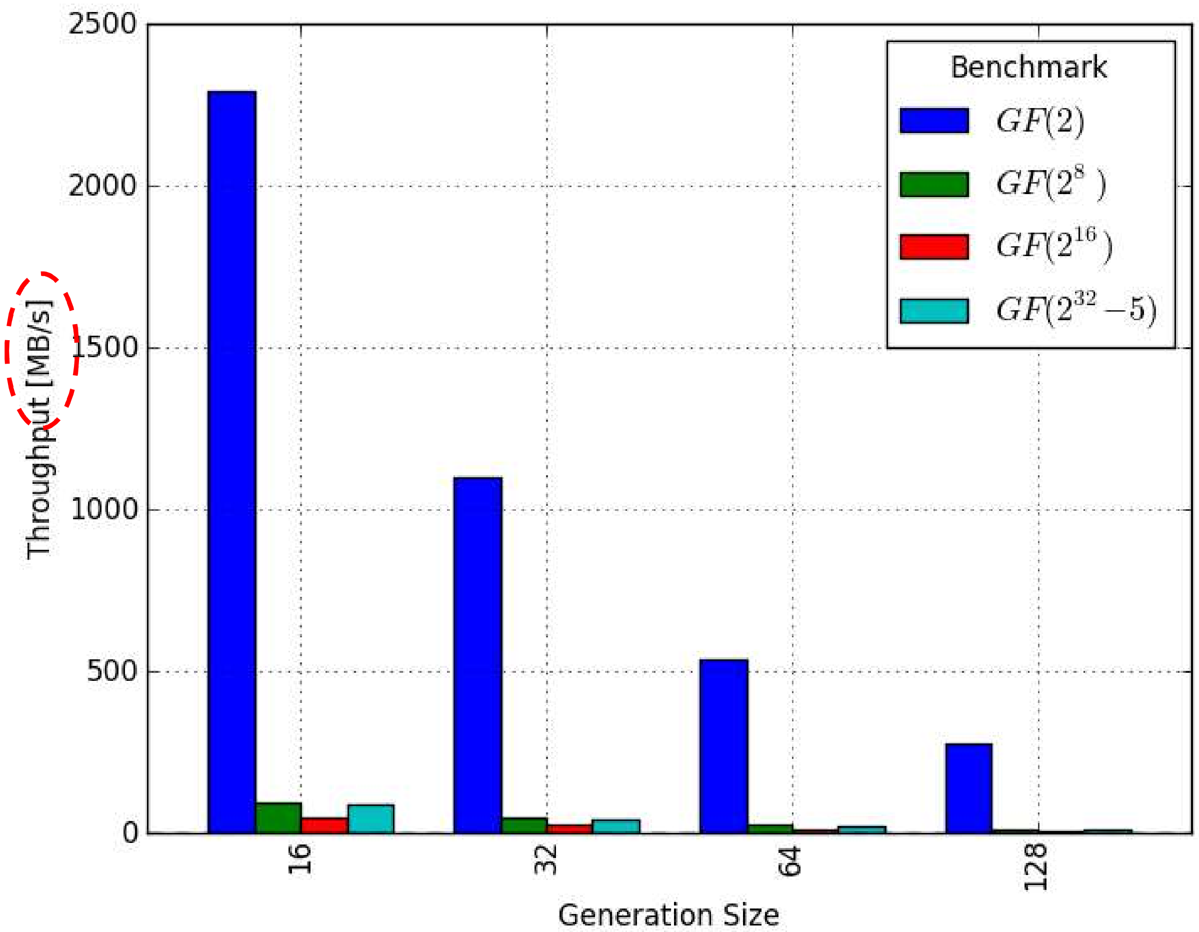
Android (2013)

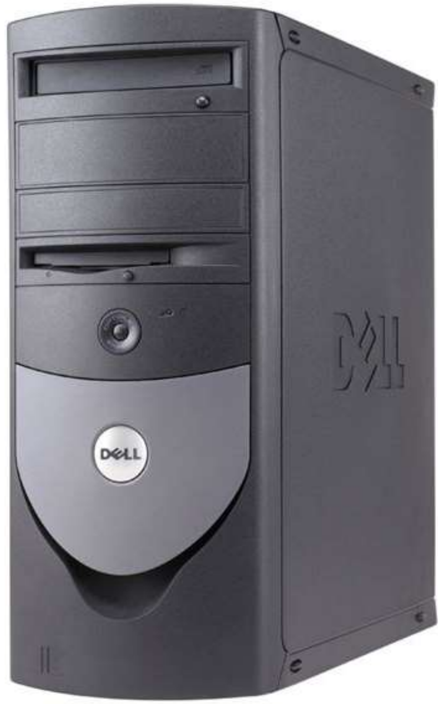
Hardware: 1.4 GHz CPU – KODO: full coding (using one core)



PC (2013)

Hardware: 3.4 GHz CPU – KODO: full coding (using single processor)

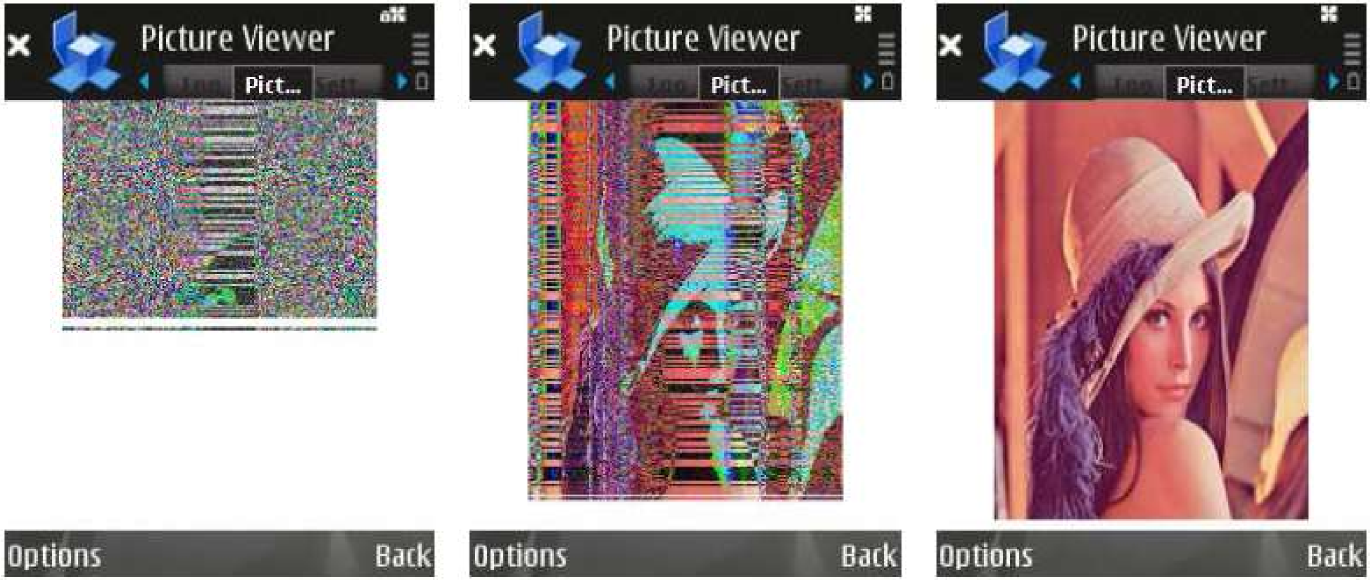




NC Modes

- FullRLNC
  - Encoding vector
  - 1.6 per generation in GF(2)
  - Adding losses
- Systematic RLNC
- Seed RLNC
- Sparse RLNC
- Perpetual RLNC
- Online RLNC
- Sliding Window
- Fulcrum

Network Coding GF(2)

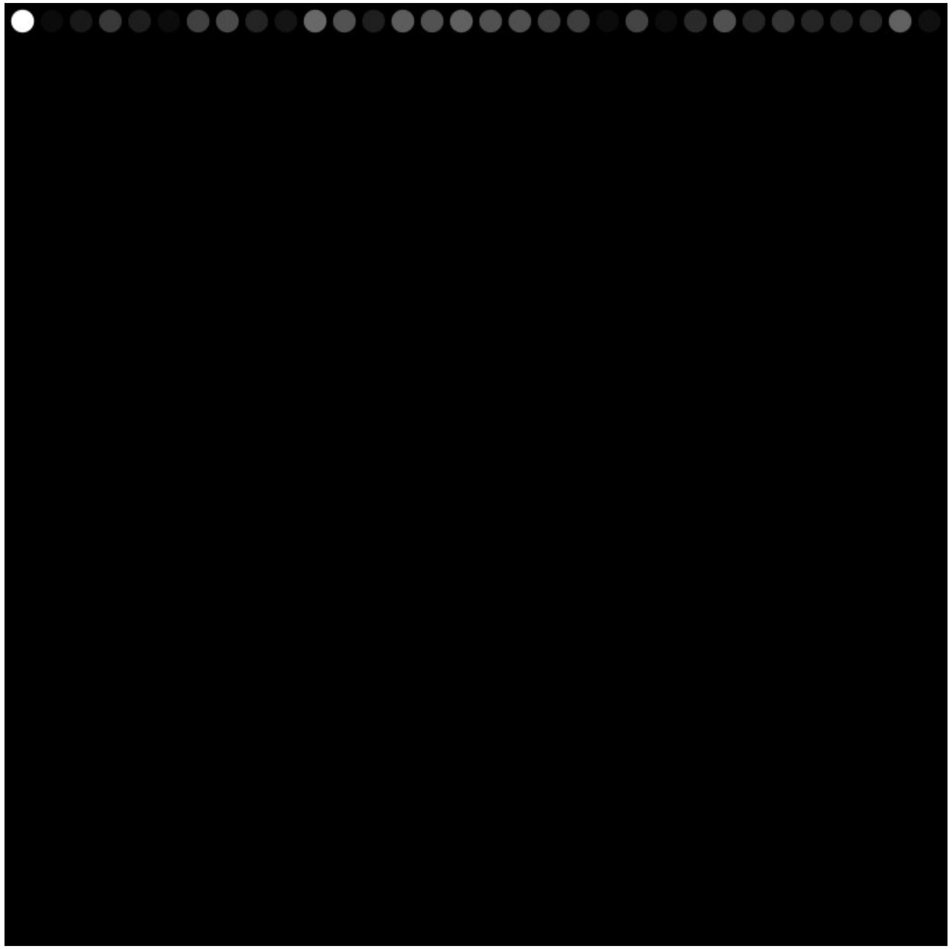


(a) (b) (c)

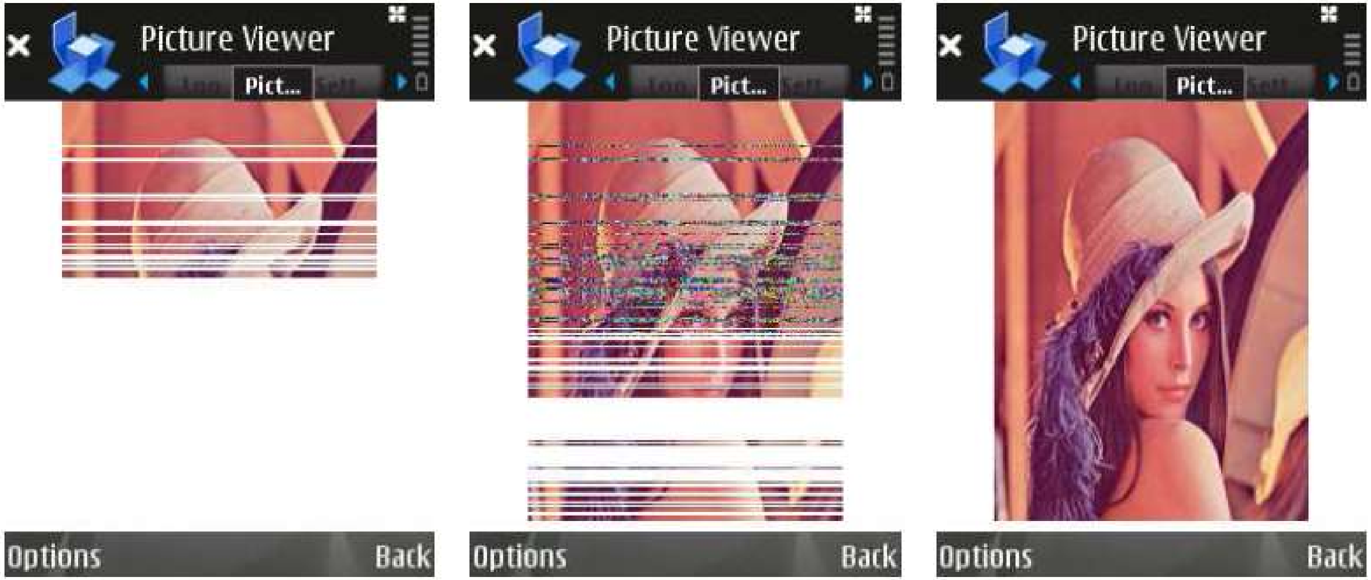
$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ & & & \\ & & & \end{pmatrix} \quad \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ & & & \end{pmatrix} \quad \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} \end{pmatrix}$$

FullRLNC

- Coding matrix is loaded with fully random elements of field size F
- Probability of zero as field element is 1/F



Systematic RLNC



(d) (e) (f)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ & & & \\ 0 & 0 & 1 & 0 \\ & & & \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ 0 & 0 & 1 & 0 \\ & & & \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ 0 & 0 & 1 & 0 \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} \end{pmatrix}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Systematic RLNC

- Starting with uncoded packets and fill wholes with fully encoded packets