TECHNISCHE UNIVERSITÄT DRESDEN

ComNets
Deutsche Telekom Chair of Communication Networks

Deutsche Telekom Chair of Communication Networks
Technische Universität Dresden

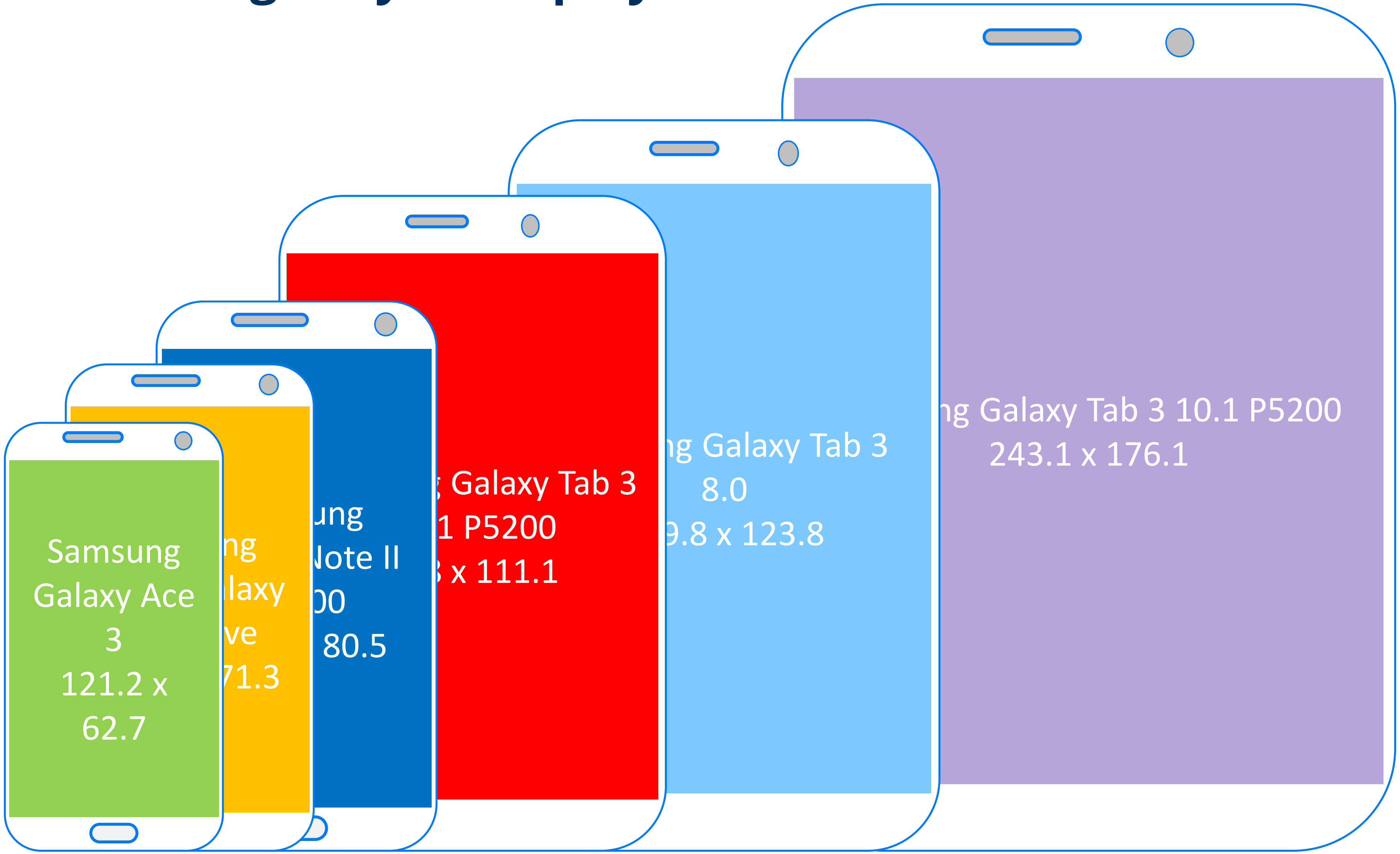
Practical Implementations of Network Coding

Frank Fitzek // Summer Semester 2020

Fulcrum RLNC

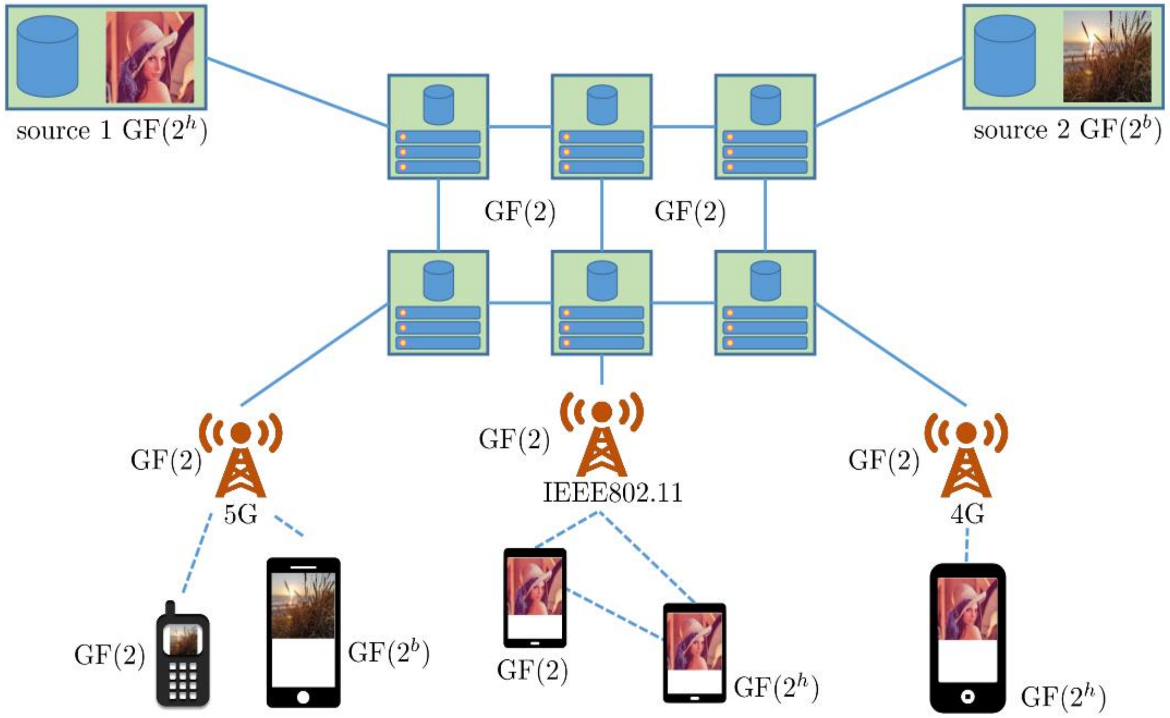
```
FulcrumEncoderFactoryBinary
FulcrumEncoderFactoryBinary16
FulcrumEncoderFactoryBinary4
FulcrumEncoderFactoryBinary8
```

Problem: Heterogenity in Display Size



General Ideas

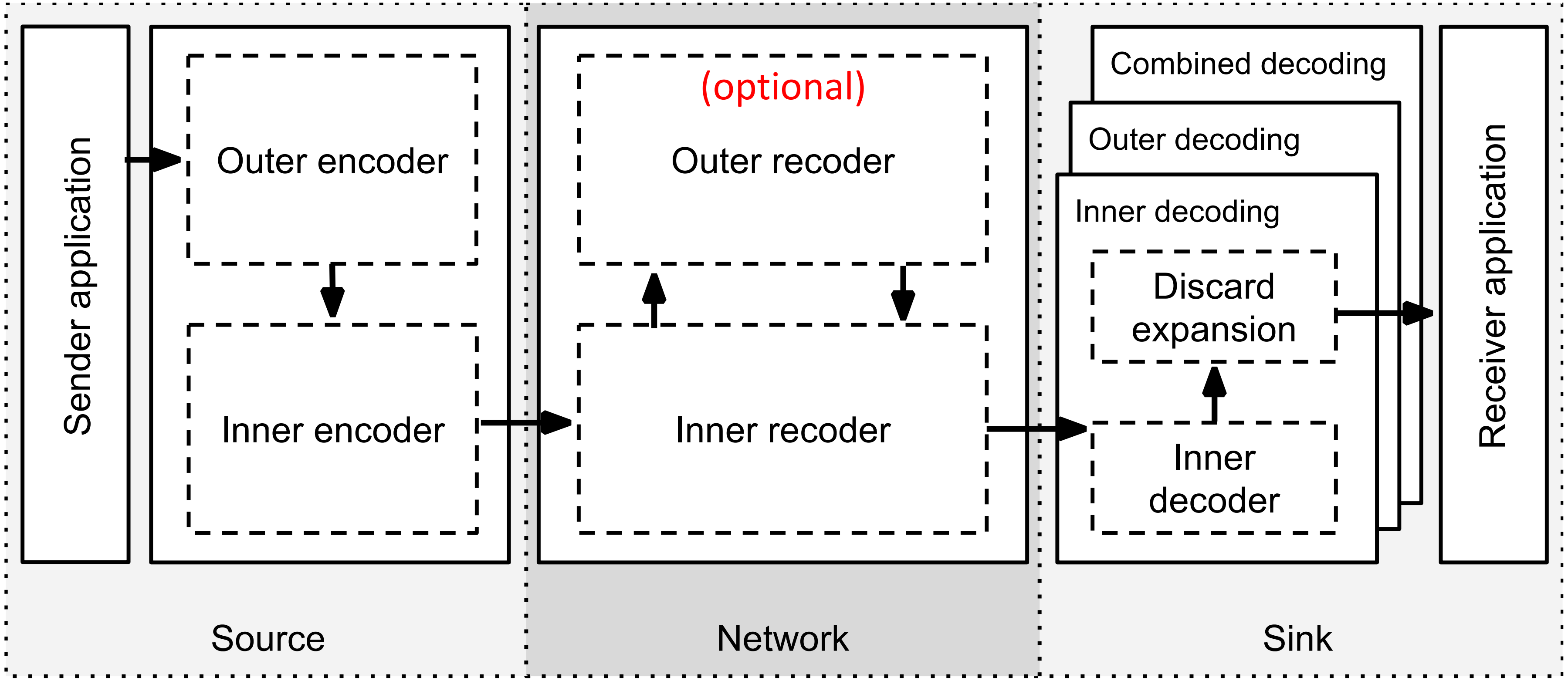
- Fluid allocation of complexity
- End devices agree on desired performance:
  - Independent from network
  - Chosen according to application requirements
- Network devices need only support a simple subset of functions
- Reduces overhead
  - Roughly 1 bit per coding coefficient
- Key: code concatenation with different field sizes



Benefits

- Simple is green, compatible, deployable
- Supports heterogeneous receivers
- Adaptive performance
- Practical recoding
- Spreads complexity to stronger devices
- Security: simple support
- New designs can be supported: backwards compatible

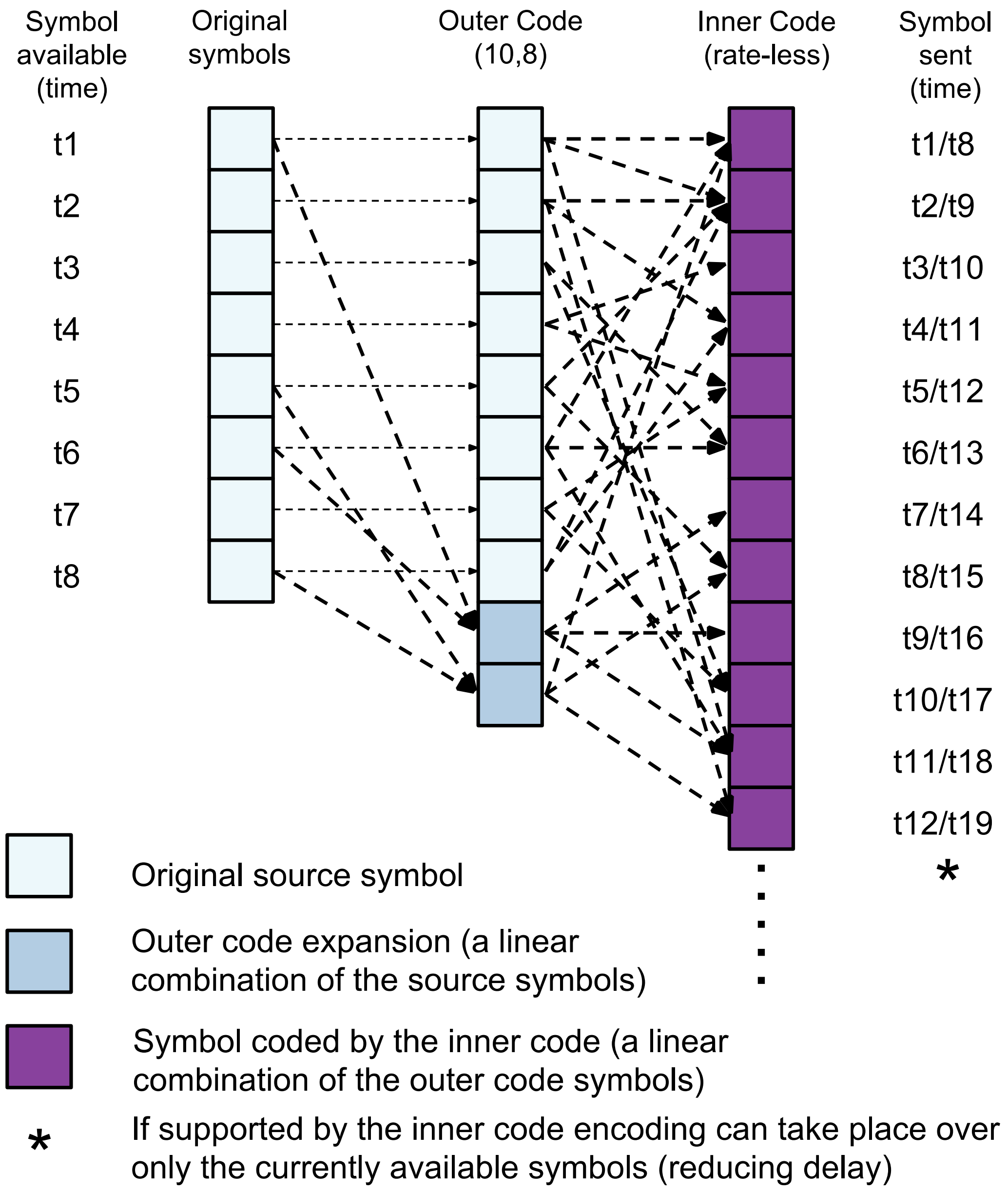
General Structure



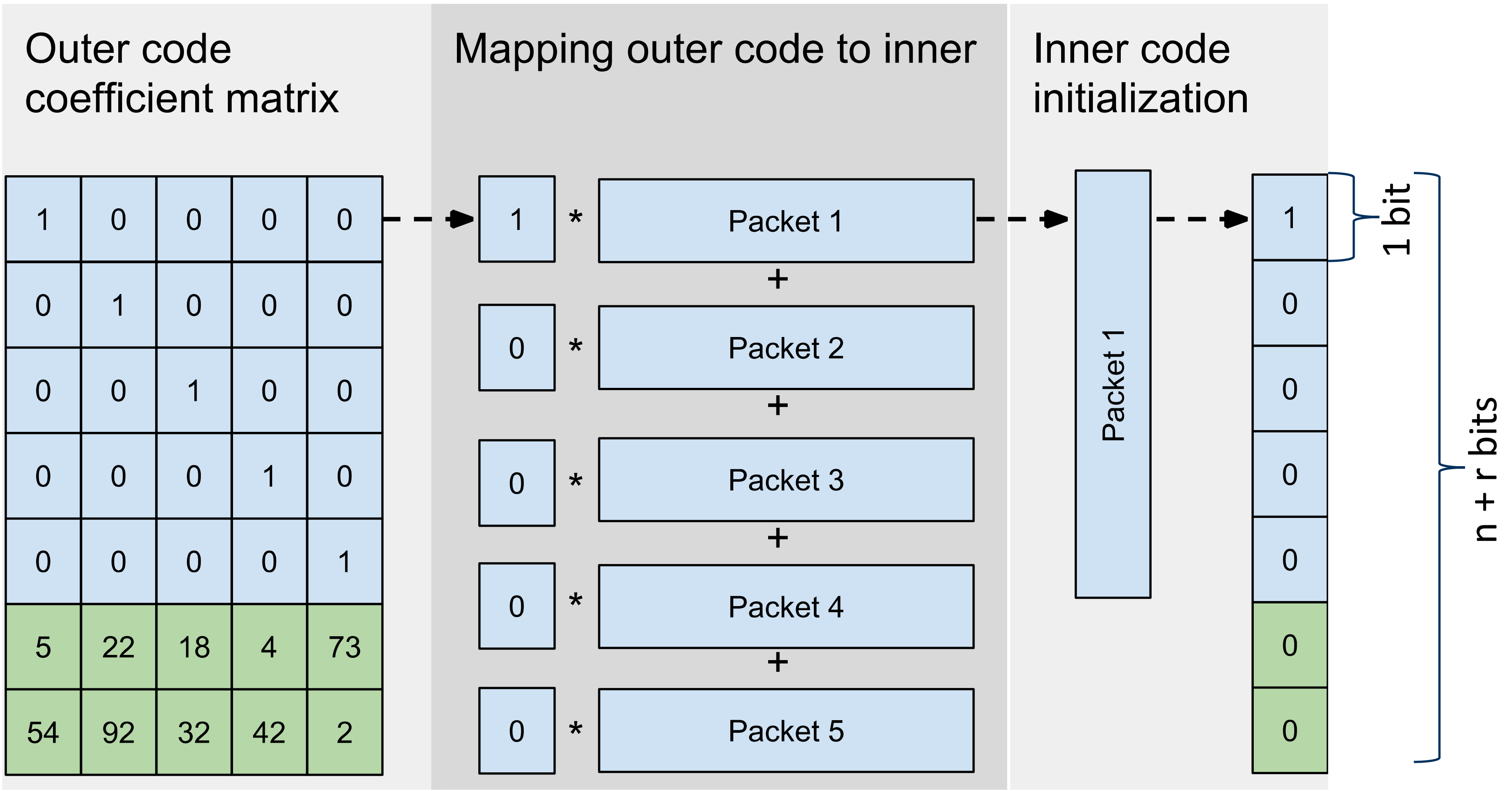
- Inner code: RLNC, Sparse RLNC, Perpetual, … GF(2)
- Outer code: (systematic) RLNC, Reed-Solomon, … GF(2^h)

Encoder

- Since inner code is created in GF(2) with pre-coded packets, then:
- n+r coefficients per inner coded packet
- This means: 1 + r/n bits per coefficient per original packet

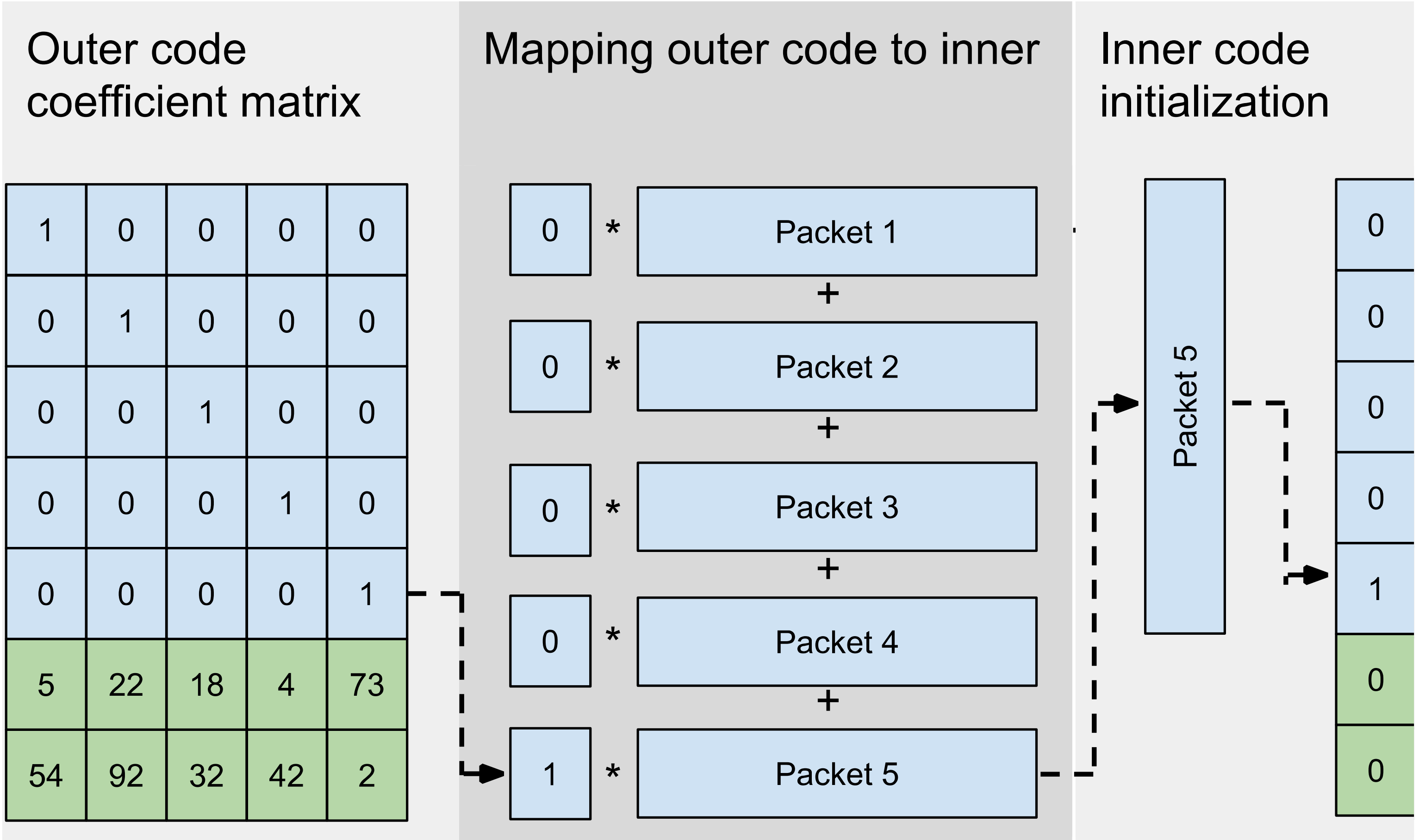


Encoder



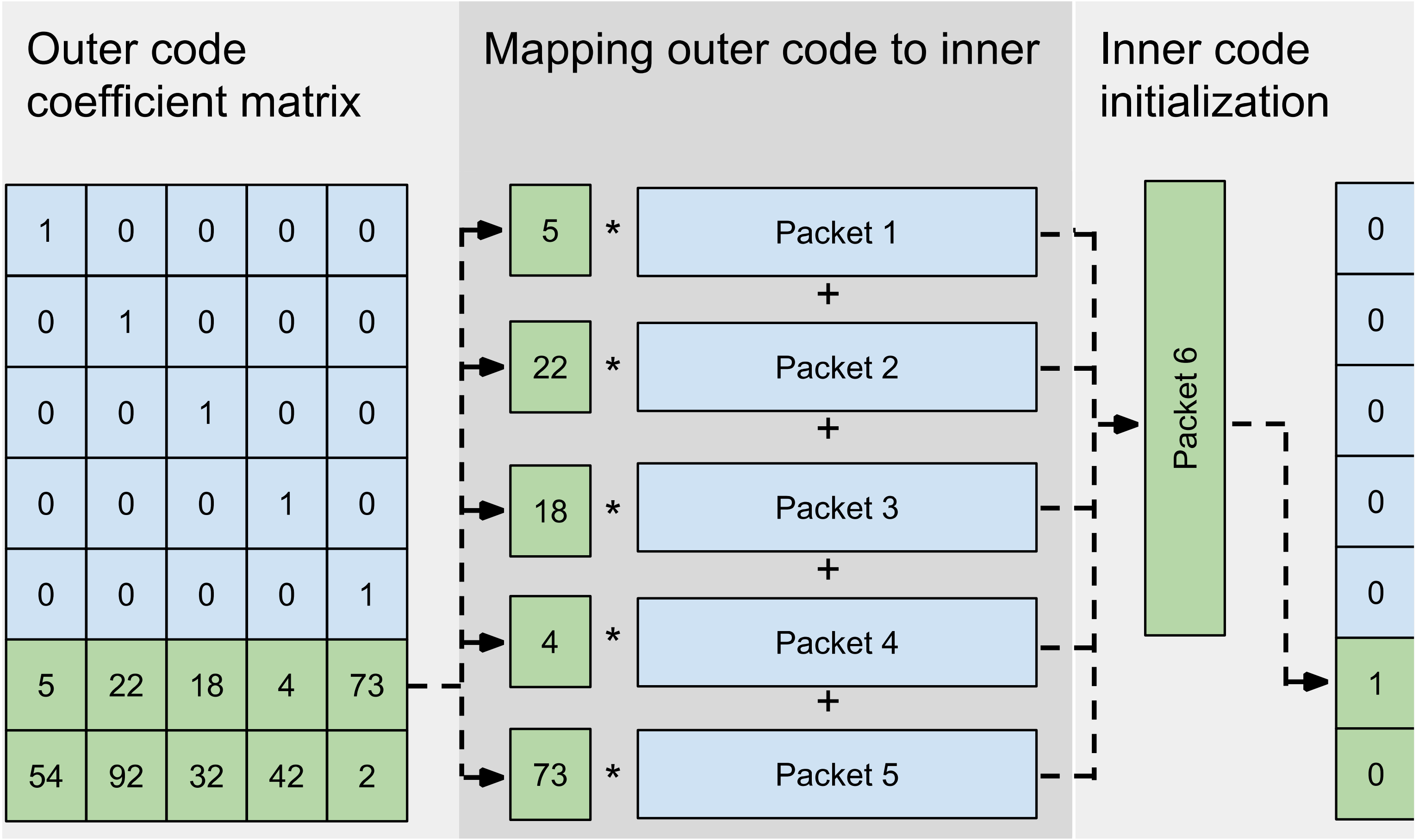
Outer code coefficient matrix
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
5 22 18 4 73
54 92 32 42 2
Mapping outer code to inner
1 * Packet 1
+
0 * Packet 2
+
0 * Packet 3
+
0 * Packet 4
+
0 * Packet 5
Inner code initialization
Packet 1
1
0
0
0
0
0
0
1 bit
n + r bits

Encoder



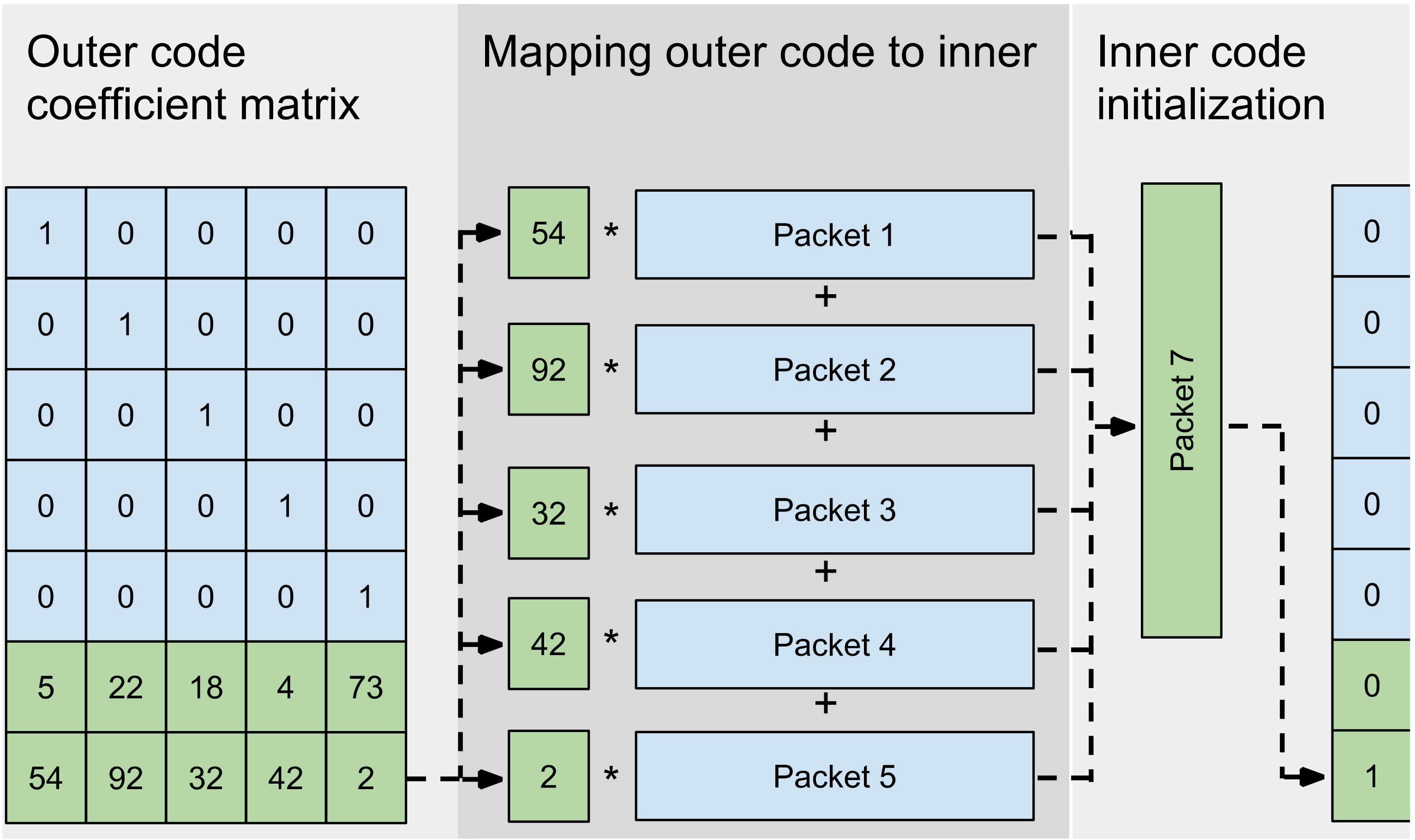
Outer code coefficient matrix
Mapping outer code to inner
Inner code initialization
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
5 22 18 4 73
54 92 32 42 2
0 * Packet 1
+
0 * Packet 2
+
0 * Packet 3
+
0 * Packet 4
+
1 * Packet 5
Packet 5
0
0
0
0
1
0
0

Encoder

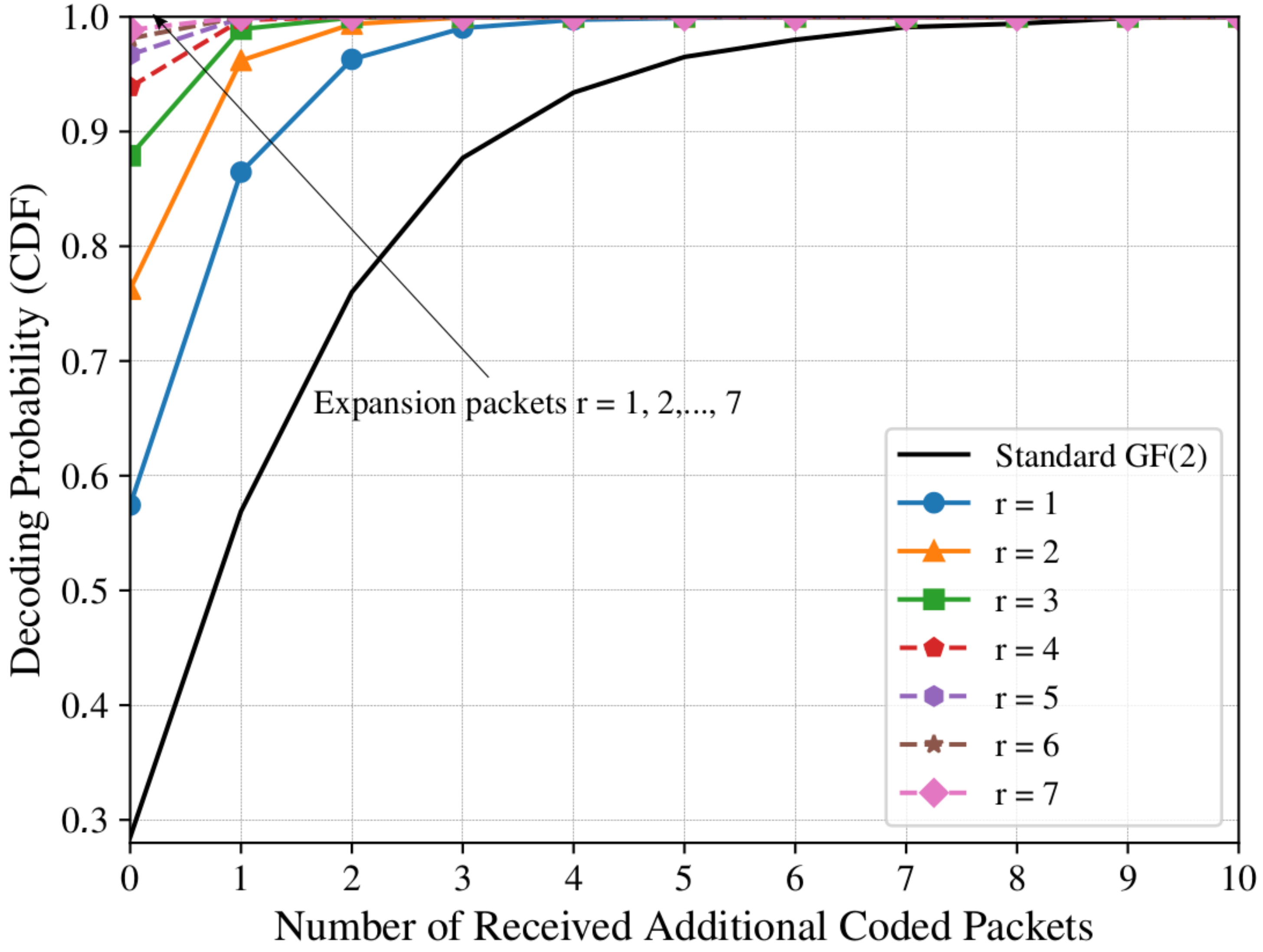


Outer code
coefficient matrix
Mapping outer code to inner
Inner code
initialization
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
5 22 18 4 73
54 92 32 42 2
5 * Packet 1
+
22 * Packet 2
+
18 * Packet 3
+
4 * Packet 4
+
73 * Packet 5
Packet 6
0
0
0
0
0
1
0

Encoder


Outer code
coefficient matrix
Mapping outer code to inner
Inner code
initialization
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
5 22 18 4 73
54 92 32 42 2
54 * Packet 1
+
92 * Packet 2
+
32 * Packet 3
+
42 * Packet 4
+
2 * Packet 5
Packet 7
0
0
0
0
0
0
1

Received Packets before Decoding



Higher r: Close to high-field performance

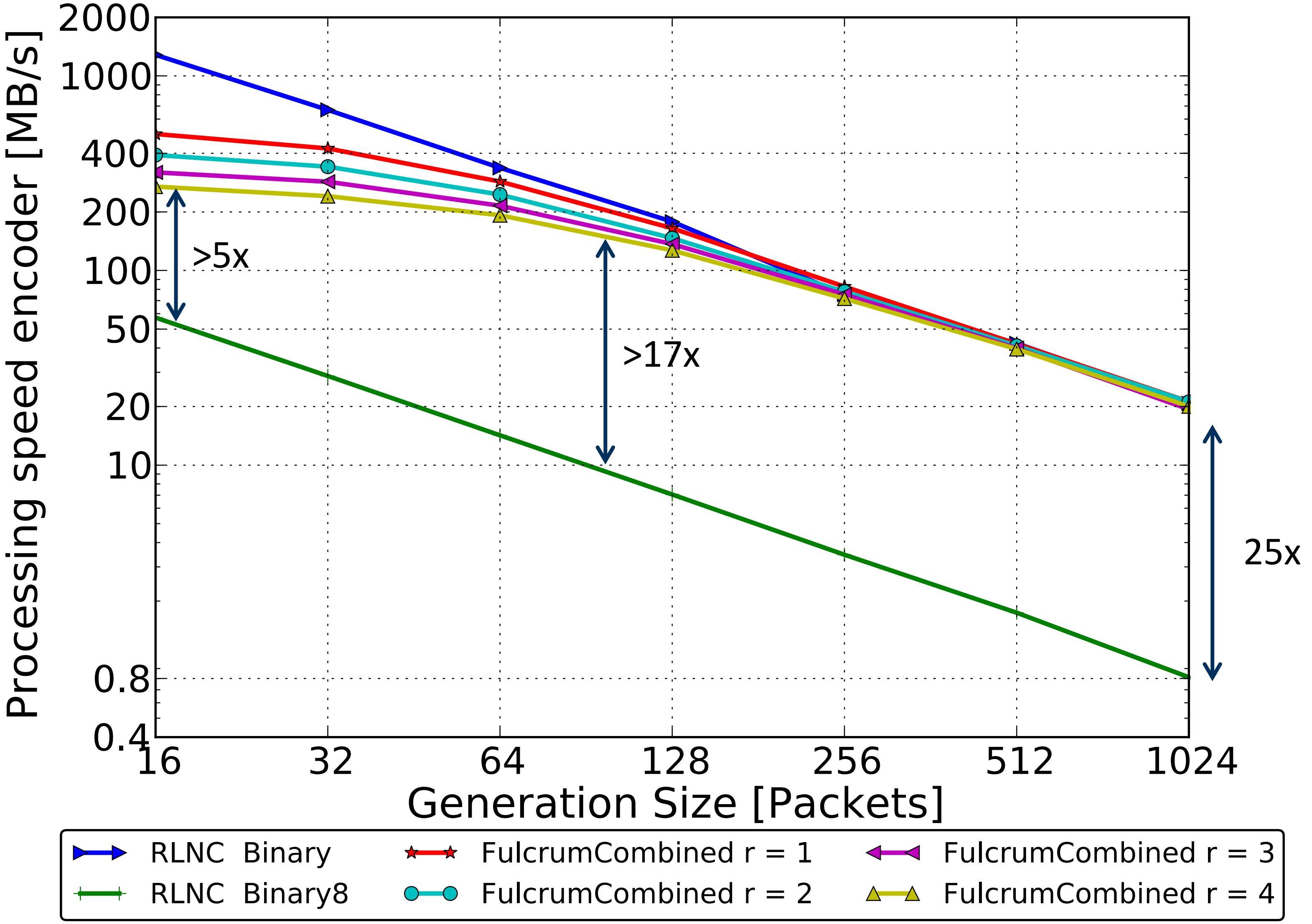
Coded Packets Sent
Mean: $n + O(2^{-r})$
Variance: $O(2^{-r})$

Received Packets before Decoding

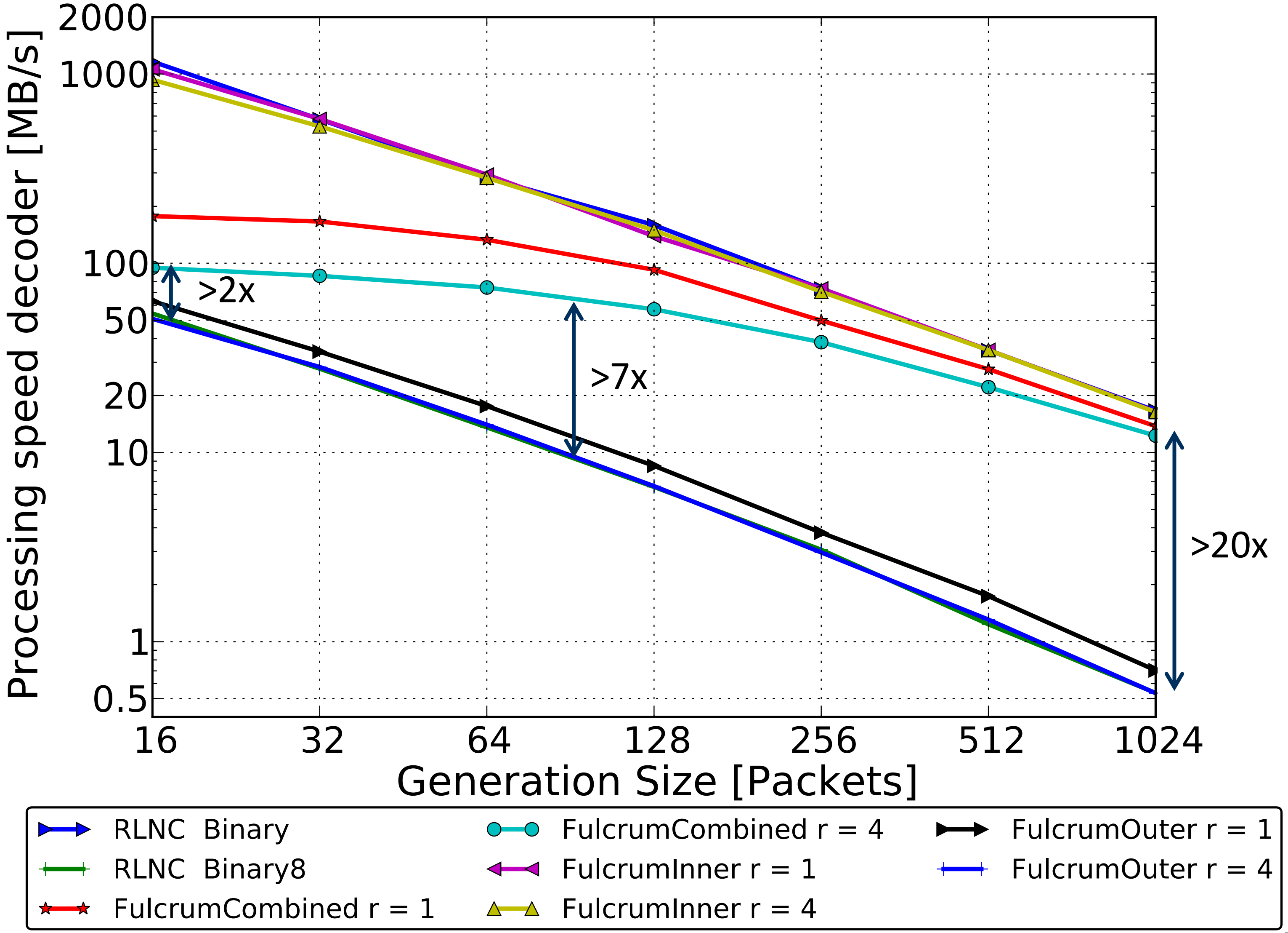
Code		Decoding after receiving (coded packets)			
		n	n + 1	n + 2	n + 3
Fulcrum	r = 4	93.87%	99.75%	99.99%	99.9997%
	r = 7	99.22%	99.996%	99.99998%	99.99999992%
	r = 10	99.90%	99.9999%	99.99999996%	99.99999999998%
RaptorQ*		99%	99.99%	99.9999%	

* Qualcomm. (2013, Dec.) Raptorq - the superior fec technology
Available: http://www.qualcomm.com/media/documents/raptorq-data-sheet

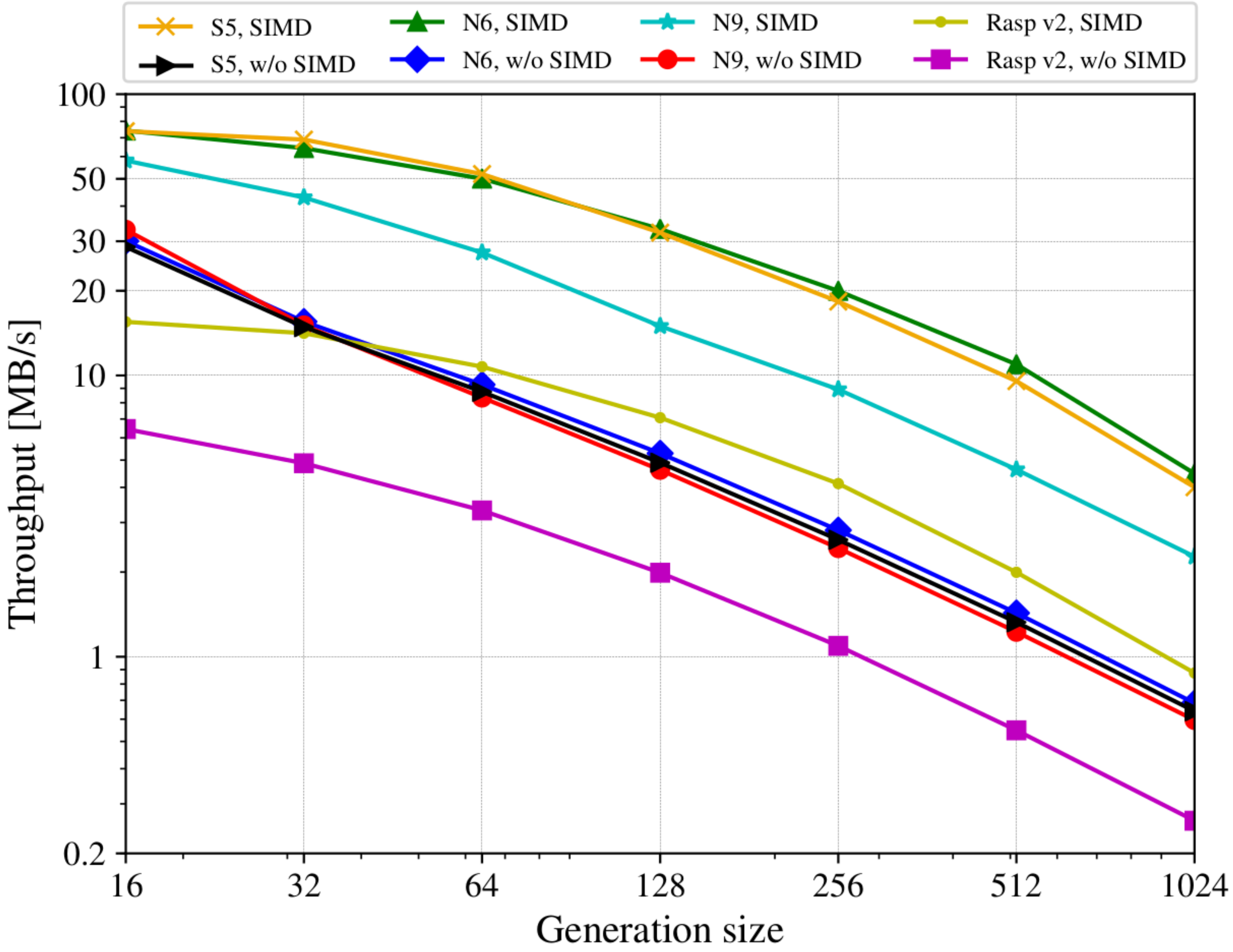
Performance Results: Encoder



Performance Results: Decoder



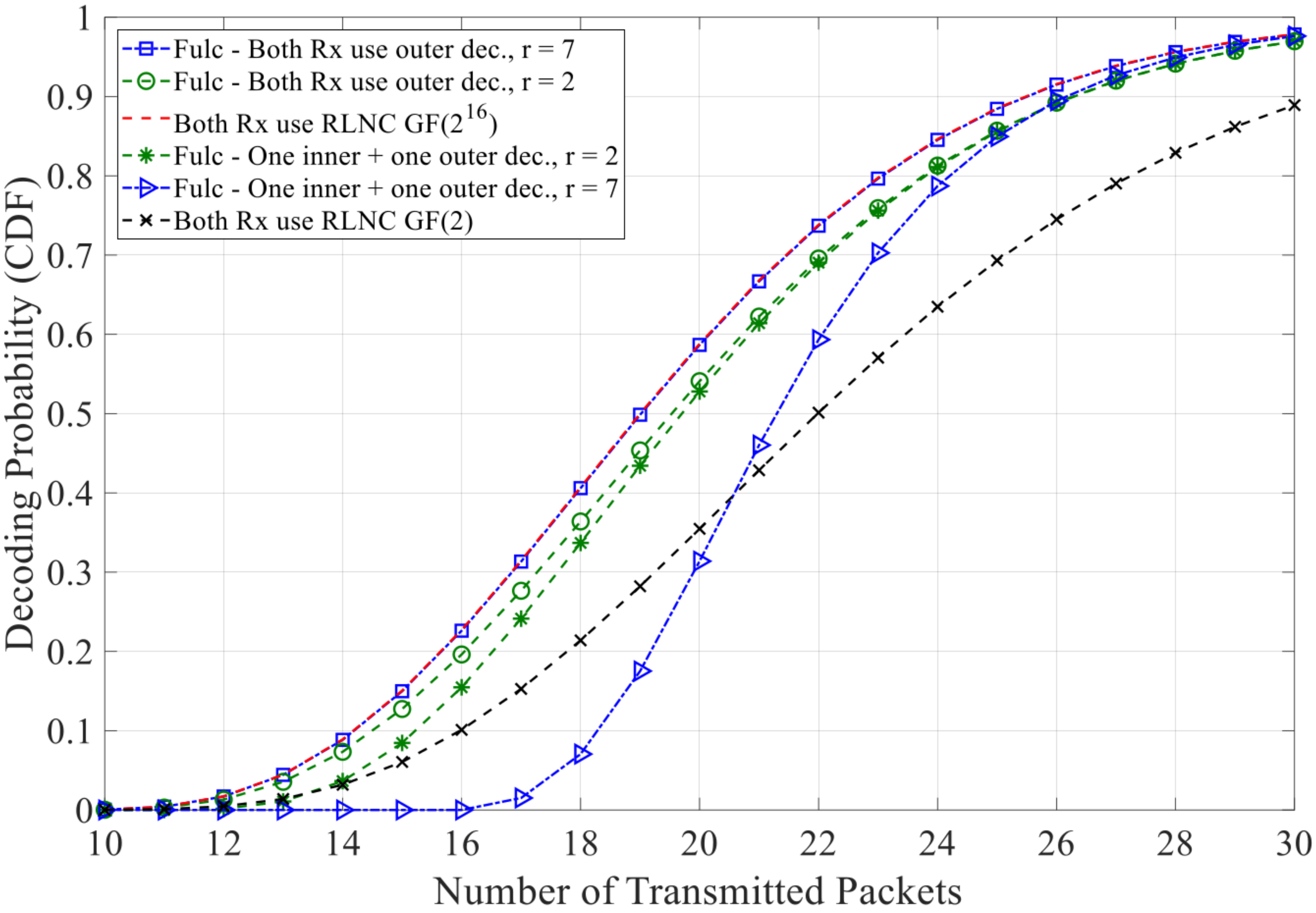
Performance Results: decoder with different devices

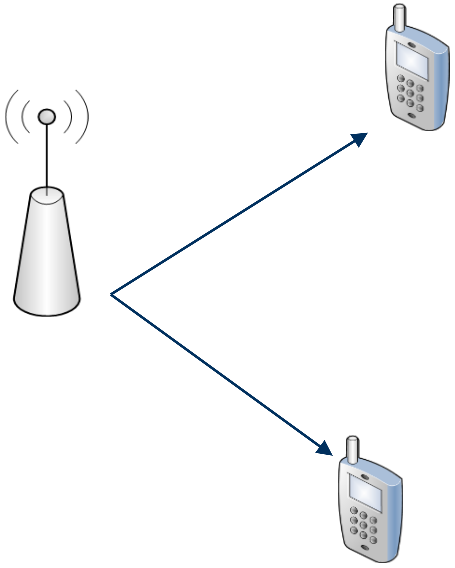


Advantages

- Low overhead:
  - 1 + r/n ~ 1 bit per coefficient per packet → like GF(2)
  - Total transmitted packets: n + O(2-r) → like higher fields
- Processing speed (complexity) compared to GF(2^8):
  - Encoder 5x to 25x faster
  - Decoder 2x to >20x faster
- Supports heterogeneous receivers
- Allows a fluid allocation of complexity
- Simple security support
- Network can implement a bare minimum:
  - Just XOR packets!

Heterogeneous Users





TECHNISCHE UNIVERSITÄT DRESDEN

ComNets
Deutsche Telekom Chair of Communication Networks

Deutsche Telekom Chair of Communication Networks
Technische Universität Dresden

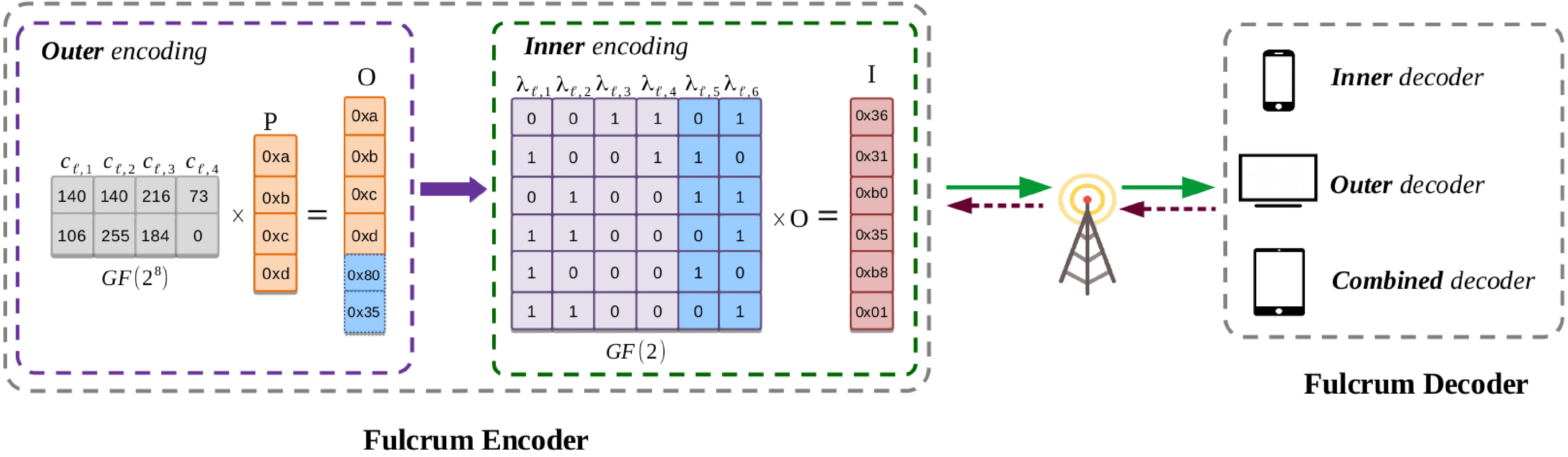
Fulcrum Network Coding

Vu Nguyen // Summer Semester 2020

Contents

- Fulcrum Codes: encoding and decoding (revising)
- Adaptive decoding for Fulcrum codes
- DSEP Fulcrum - Dynamic Sparsity and Expansion Packets
- DSEP Recoding mechanism

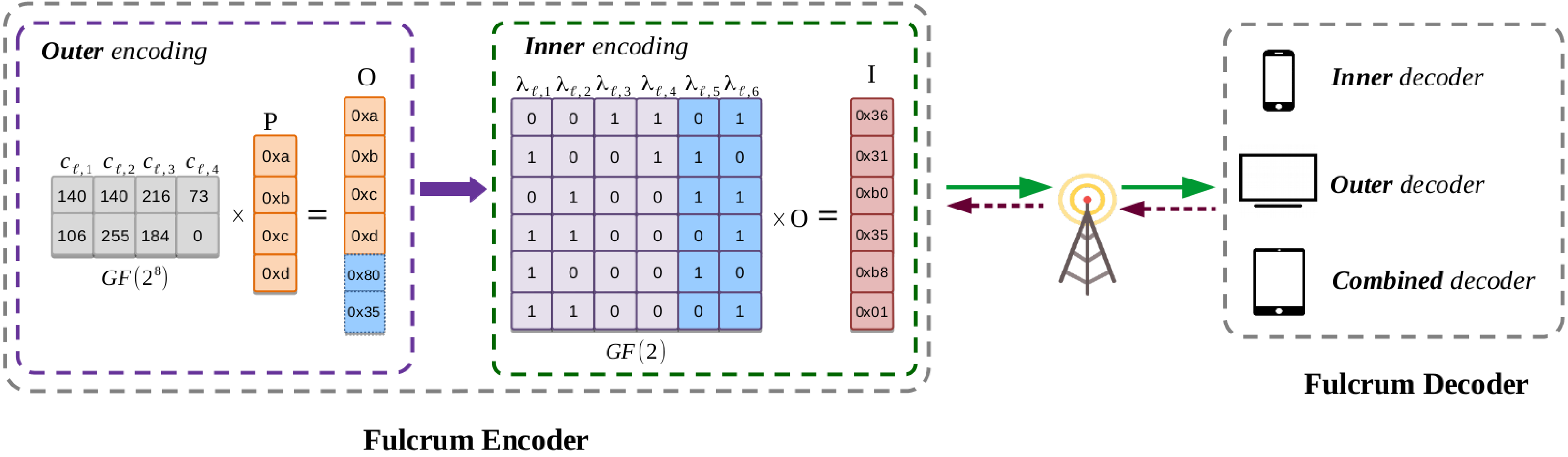
Fulcrum Codes: encoding and decoding



$$o_\ell = \sum_{j=1}^{n} c_{\ell,j} p_j, \quad \ell = 1, 2, \ldots, r. \quad (1)$$

$$I_\ell = \sum_{j=1}^{n+r} \lambda_{\ell,j} o_j, \quad \ell = 1, 2, \ldots. \quad (2)$$

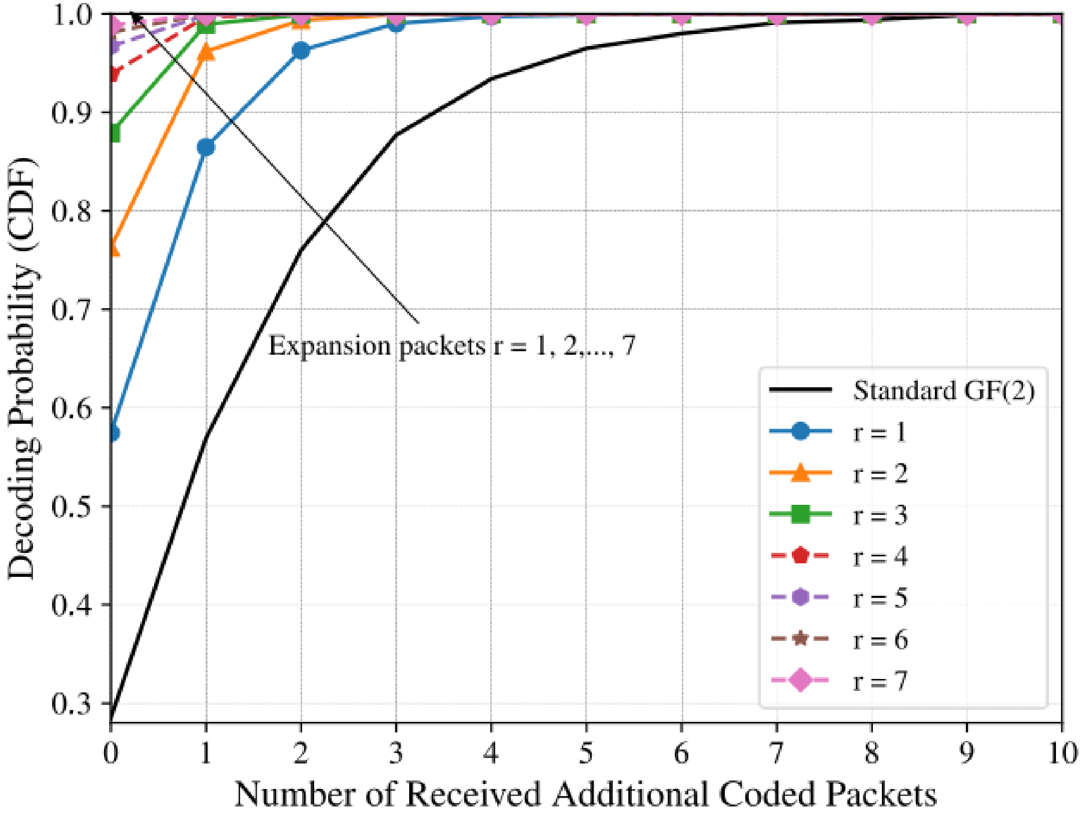
Fulcrum Codes: encoding and decoding

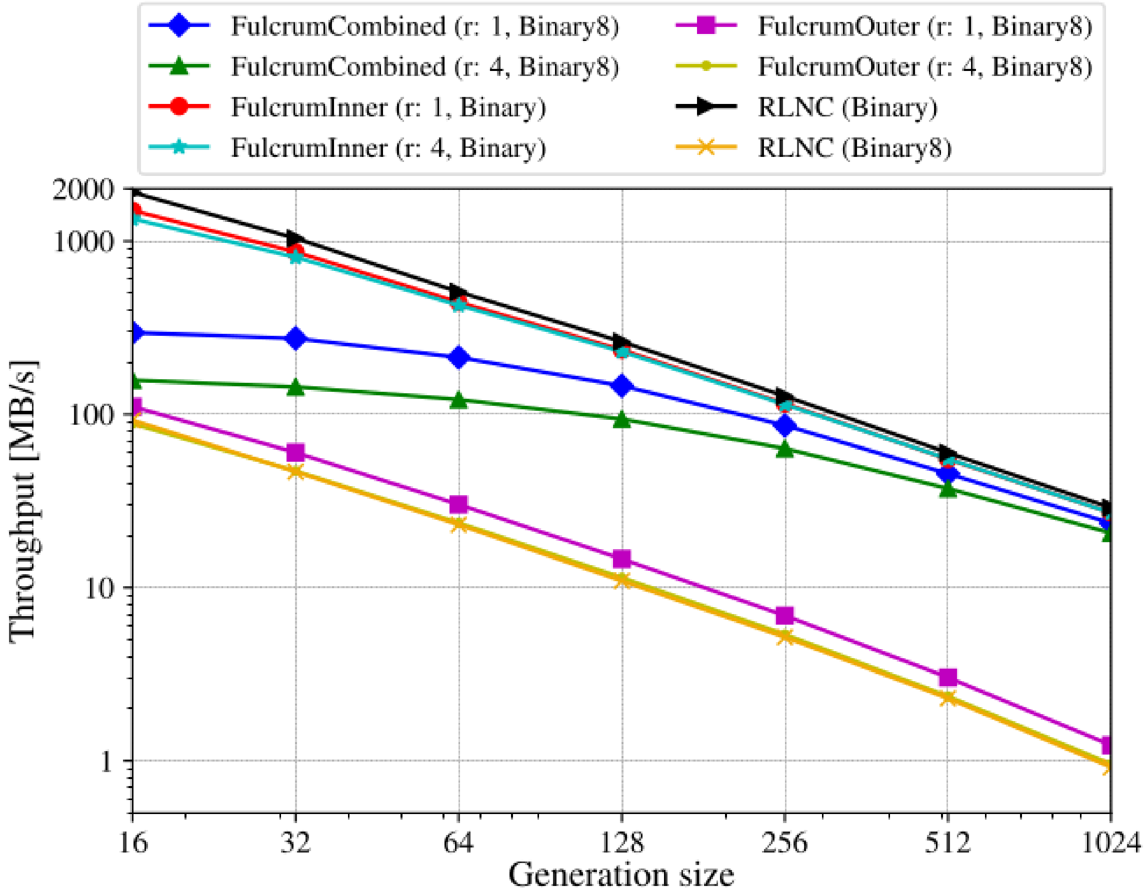


m	m < n	n ≤ m < n + r	n + r ≤ m
Operable decoder	None	Either outer or combined	All: inner, outer, combined

m: number of received coded packets at receivers

Fulcrum Codes: performance

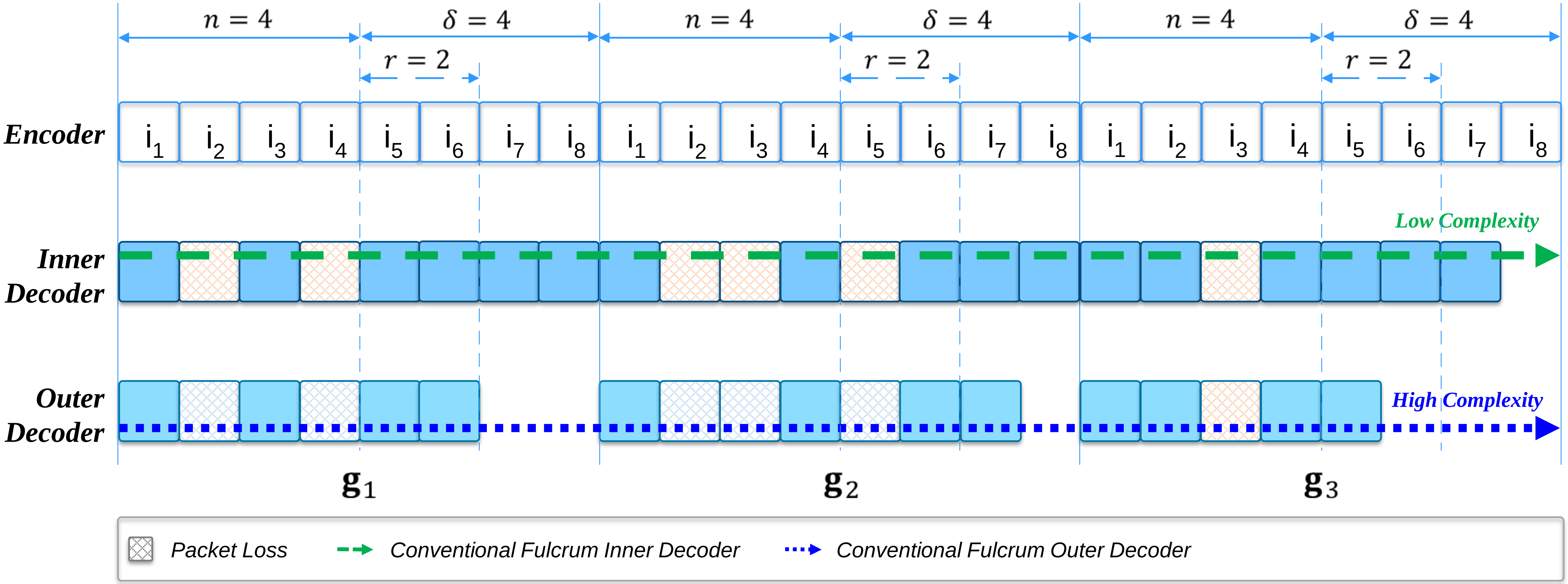




D. E. Lucani, M. V. Pedersen, C. W. Sorensen, F. H. P. Fitzek, J. Heide, O. Geil, V. Nguyen and M. Reisslein, “Fulcrum: Flexible network coding for heterogeneous devices”, *IEEE Access*, vol. 2018, pp. 1-21, 2018.

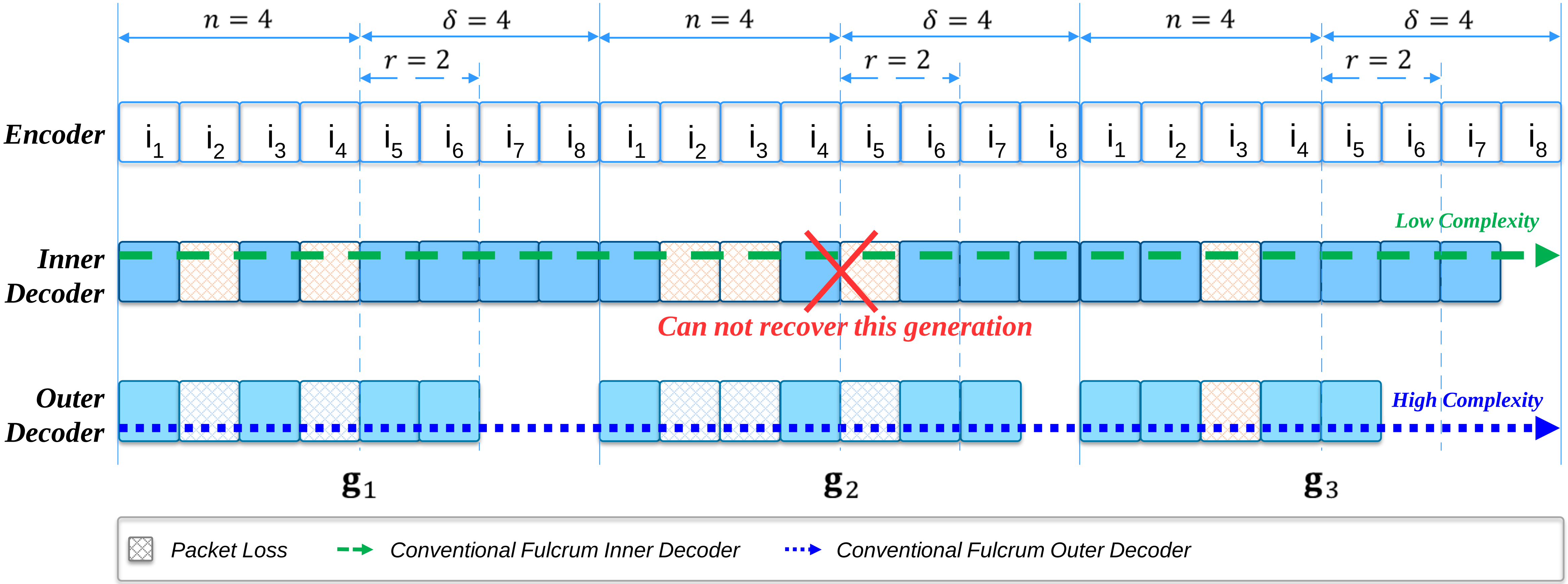
Adaptive decoding for Fulcrum codes

Fulcrum adaptive decoding: motivation example

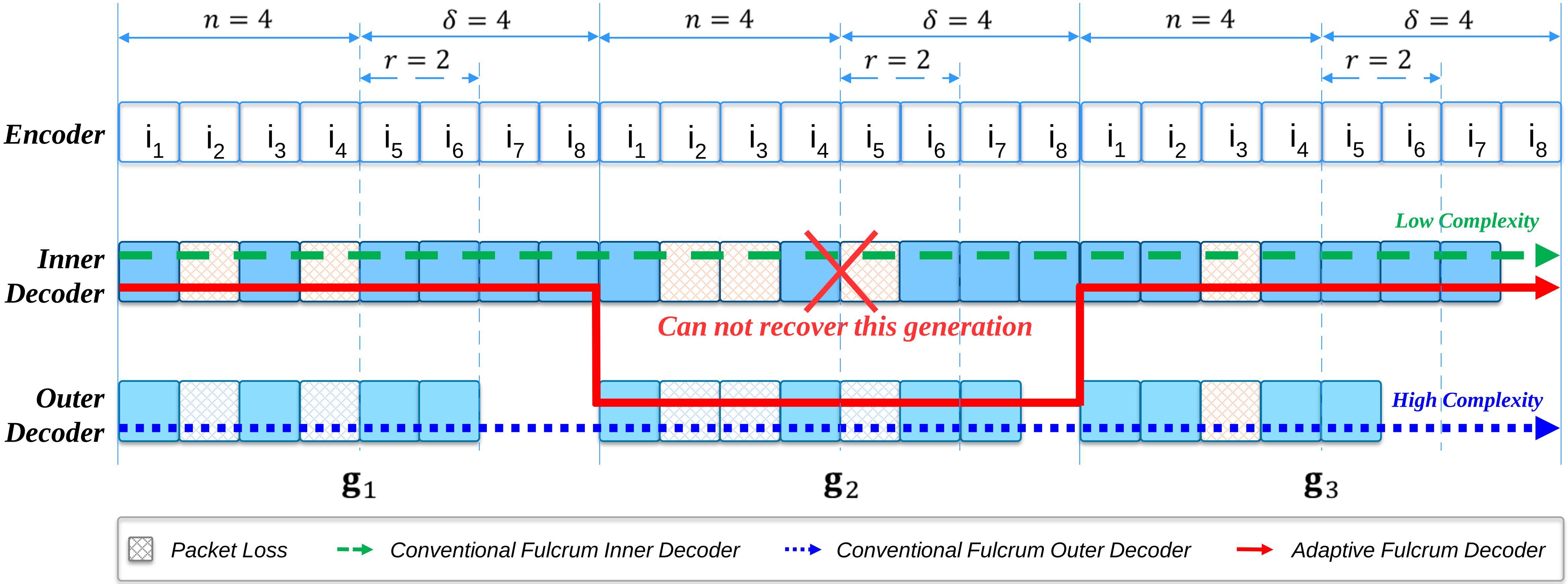


δ: number of redundant packets used to compensate for the loss

Fulcrum adaptive decoding: motivation example



Fulcrum adaptive decoding: motivation example



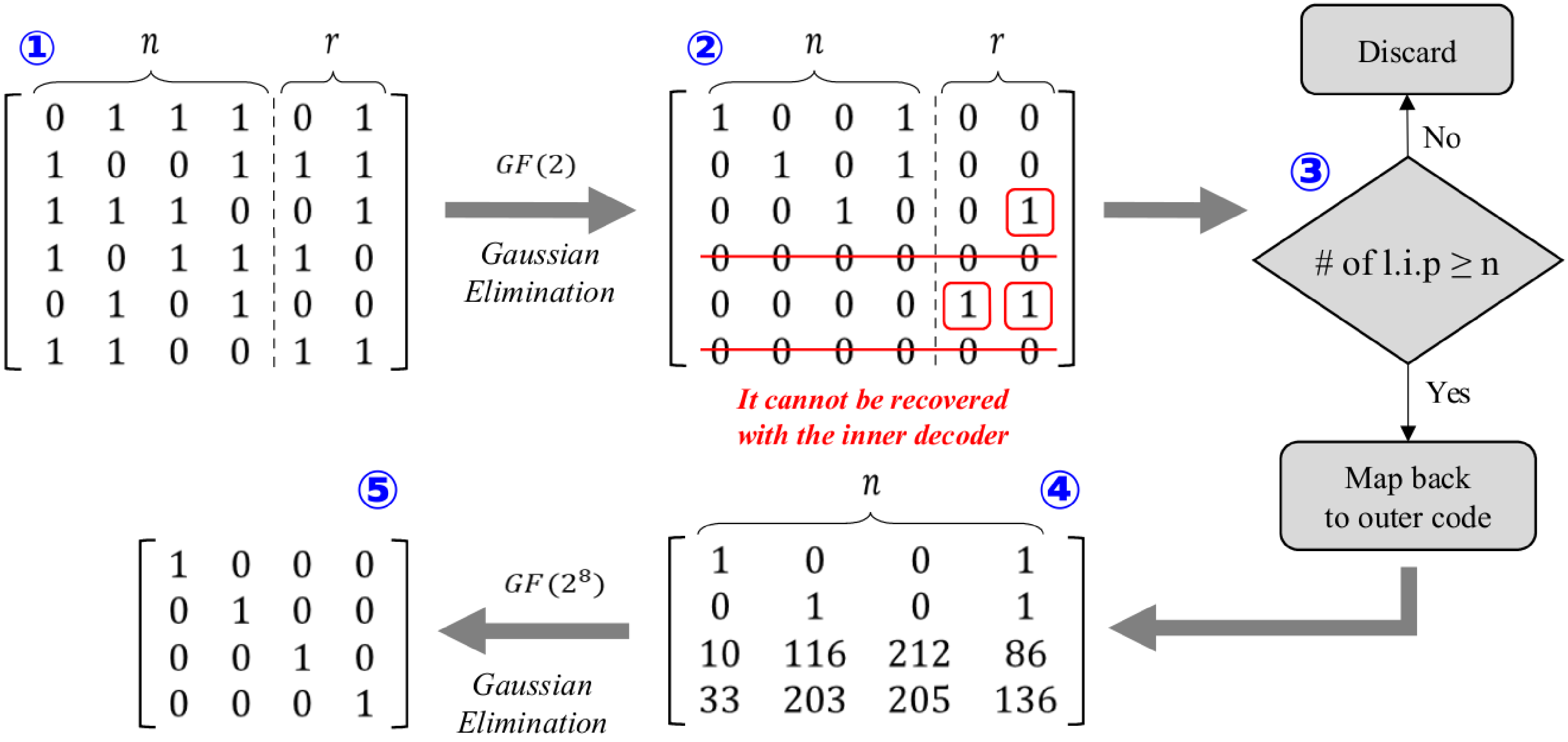
- Receivers can adaptively decide to operate on which decoder in accordance with the computational capabilities and channel environments.

Fulcrum adaptive decoding: characteristics

Decoder		Generation			Characteristics	
		g_1	g_2	g_3	Decoding Prob.	Complexity
Inner		O	X	O	Low	Low
Outer		O	O	O	High	High
Adaptive	Inner	O	-	O	High	Medium
	Outer	-	O	-		

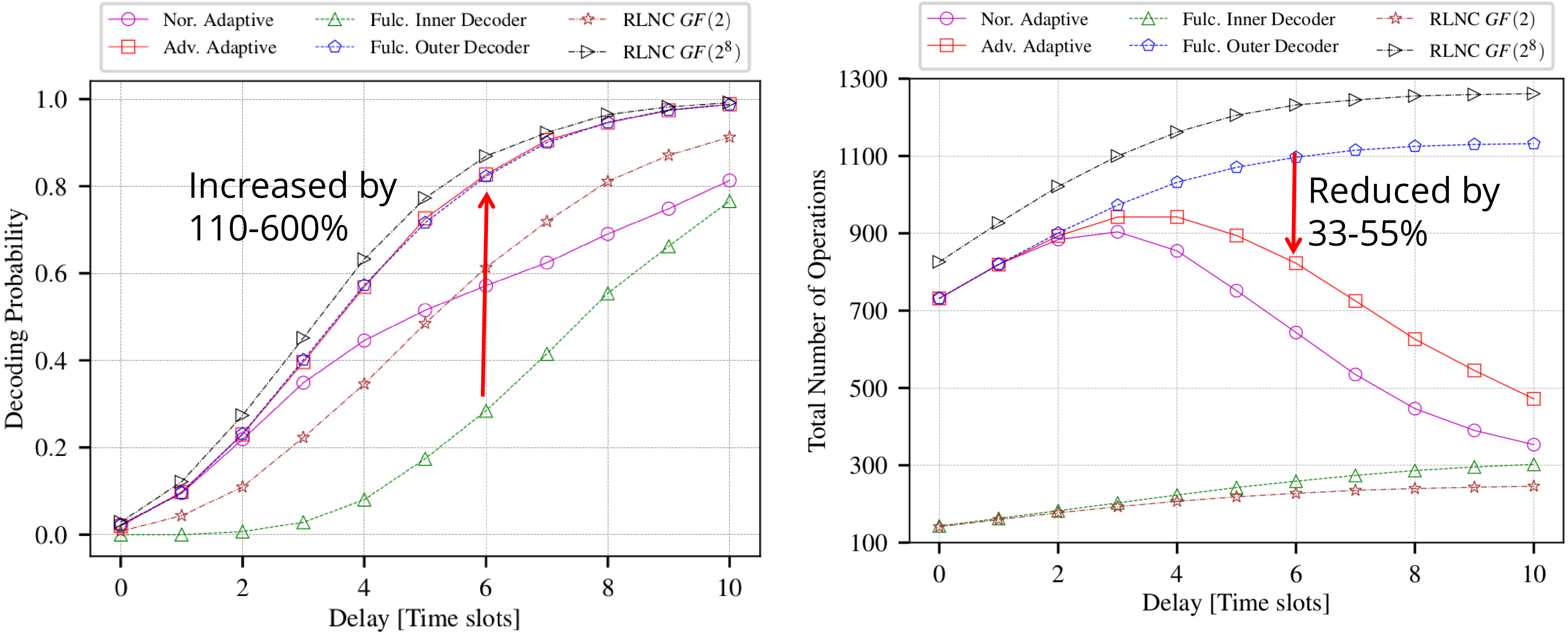
- Adaptive decoding produces the best performance in terms of decoding probability and computational complexity at end devices.

Fulcrum adaptive decoding: algorithm



Algorithm increases the chance to recover the original packets, while reducing the complexity

Fulcrum adaptive decoding: results



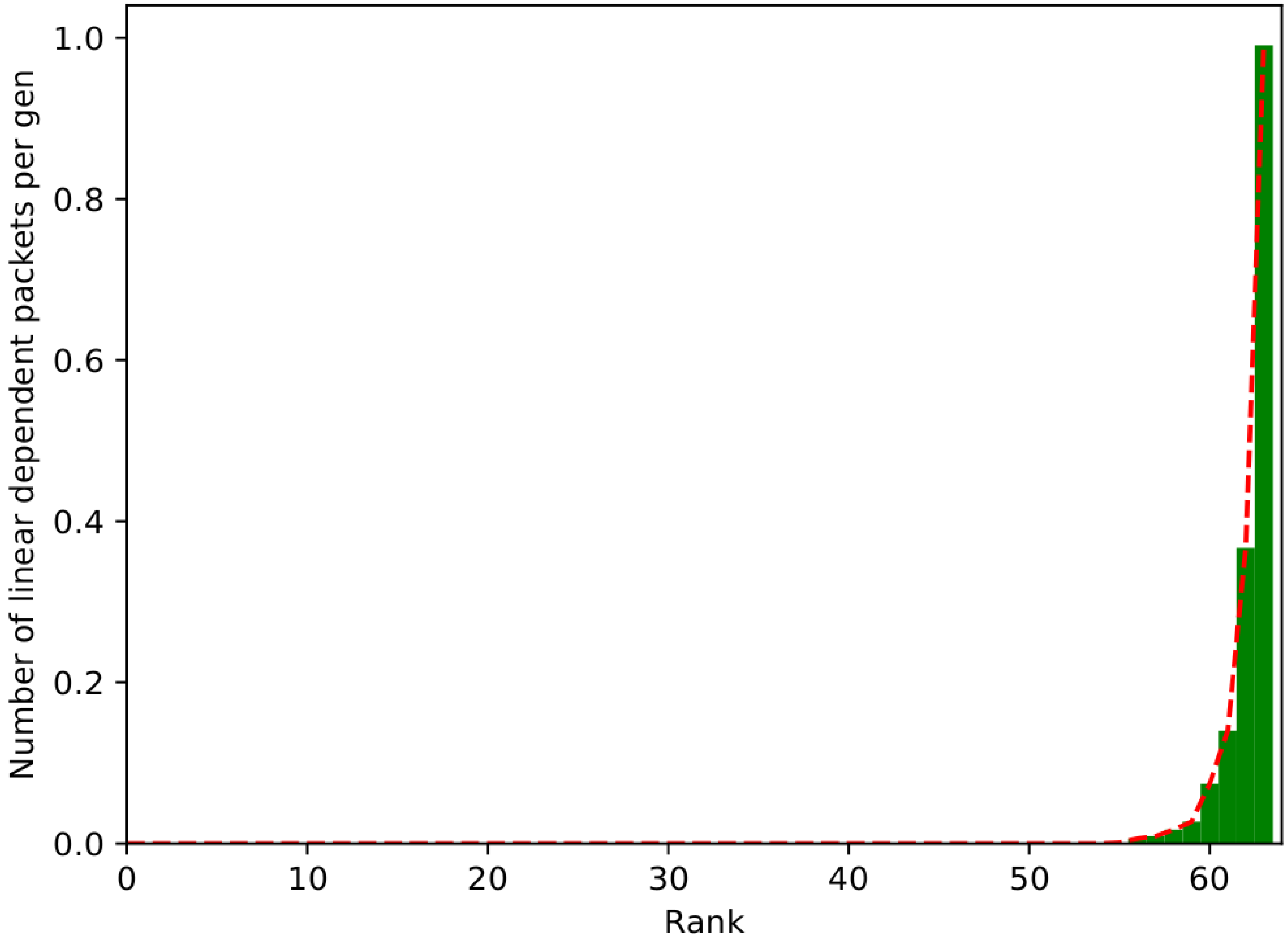
Erasure probability: 0.2

Advantages

- Receivers adaptively decide to operate with low or high finite field.
- Suitable for the change of channels.
- Reduce the complexity by up to 55%.
- Increase the decoding probability as high as outer/combined decoder.

DSEP Fulcrum
(Dynamic Sparsity and Expansion Packets)

DSEP Fulcrum: Motivation



GF(2) has high linear dependency in the end!

GF(2^8) has nearly no linear dependency, but high complexity

DSEP Fulcrum: Motivation

inner coding coefficient matrix — outer packets

generate sparse non-zero coding coefficients at the begin of generation

generate dense non-zero coding coefficients at the end of generation

$$\left[\begin{array}{ccccccc|ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \color{red}{1} & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \color{red}{1} \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}\right] \times \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ \color{red}{o_1} \\ \color{red}{o_2} \\ \color{red}{o_3} \end{bmatrix}$$

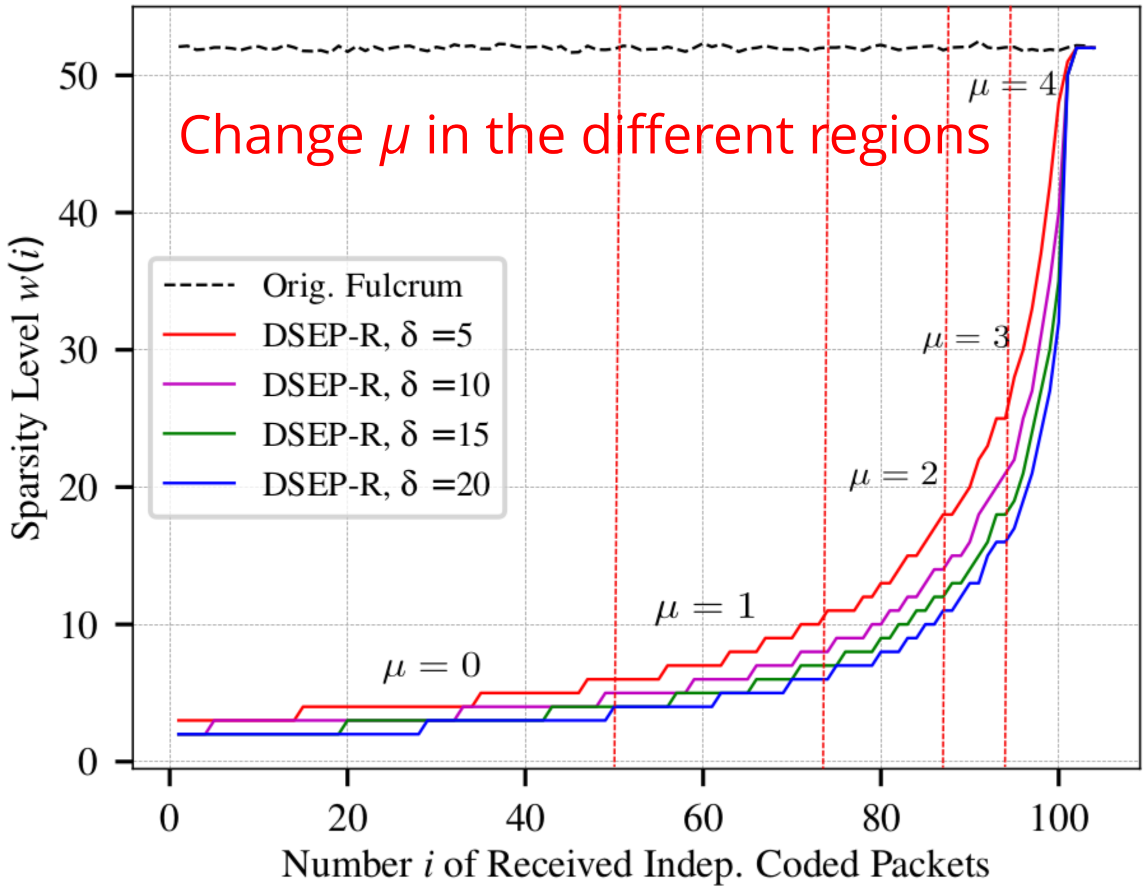
DSEP Fulcrum: general idea

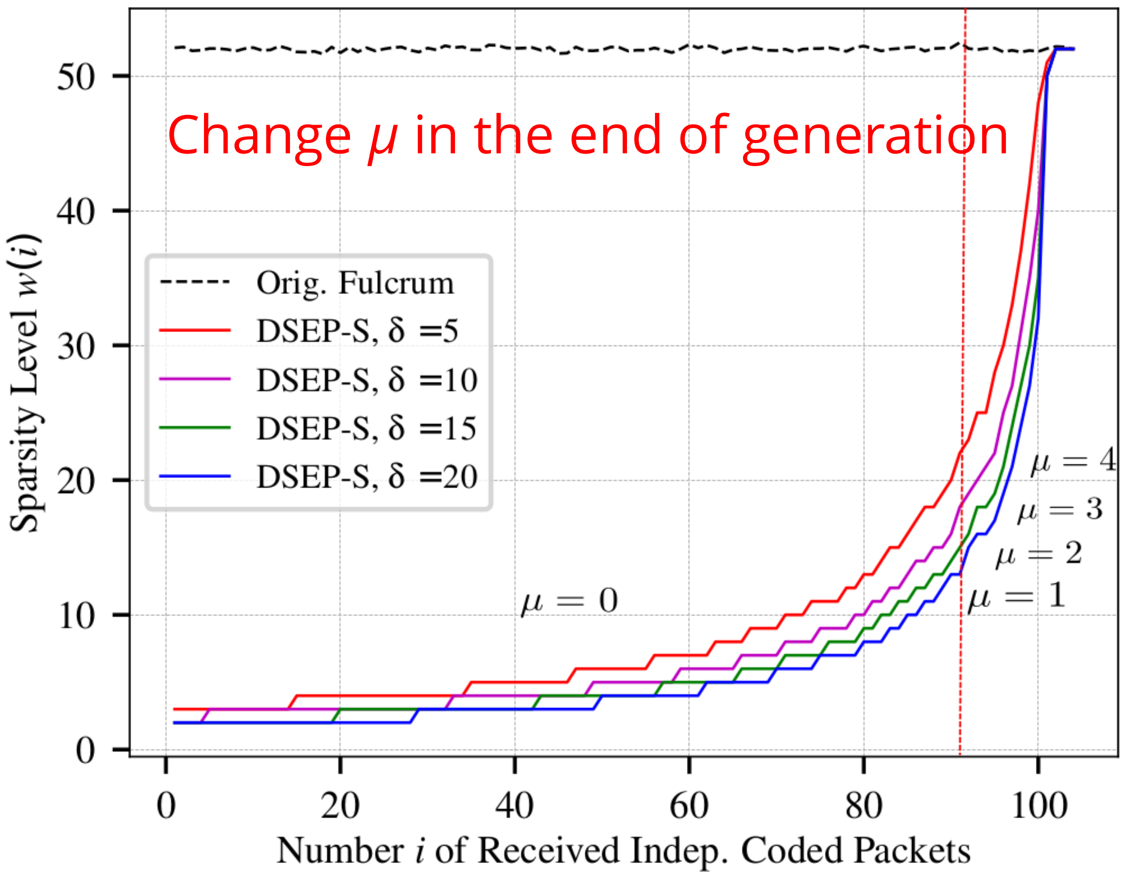
	Conventional Fulcrum	DSEP Fulcrum
Sparsity	static sparsity ω	$w(i) = (n+\mu)\min\left\{\frac{1}{2},\ 1 - \sqrt[n+\mu-i]{1 - \frac{n+r}{n+r+\delta}}\right\}$
Inner coded packets	$I_\ell = \sum_{j=1}^{n+r} \lambda_{\ell,j} o_j,\ \ell = 1, 2, \ldots.$	$I_\ell = \sum_{j=1}^{n+\mu} \lambda_{\ell,j} o_j,\ 0 \leq \mu \leq r$

- n: number of original packets
- r: expansion packets
- δ: nominal extra coded packets
- μ: considered expansion packets

DSEP Fulcrum: general idea

Dynamic expansion packets μ and dynamic sparsity level $w(i)$





DSEP Fulcrum: results

Decoding probability in percent

Policy	$r = 2$				$r = 4$		$r = 6$		$r = 10$	
	0	1	2	3	0	1	0	1	0	1
Orig. Fulc.	77.16	95.88	99.39	99.91	93.43	99.74	98.05	99.967	99.56	99.998
DSEP-R, $\delta = 5$	76.64	96.25	99.43	99.98	93.23	99.68	96.81	99.937	99.10	99.993
DSEP-R, $\delta = 20$	71.82	94.54	99.10	99.84	85.68	98.89	88.98	99.470	94.32	99.877
DSEP-S, $\delta = 5$	76.60	96.02	99.43	99.94	93.48	99.72	98.14	99.957	99.44	99.996
DSEP-S, $\delta = 20$	70.55	94.24	99.10	99.86	86.64	99.07	92.43	99.756	95.43	99.910

RaptorQ*: 99% 99.99%

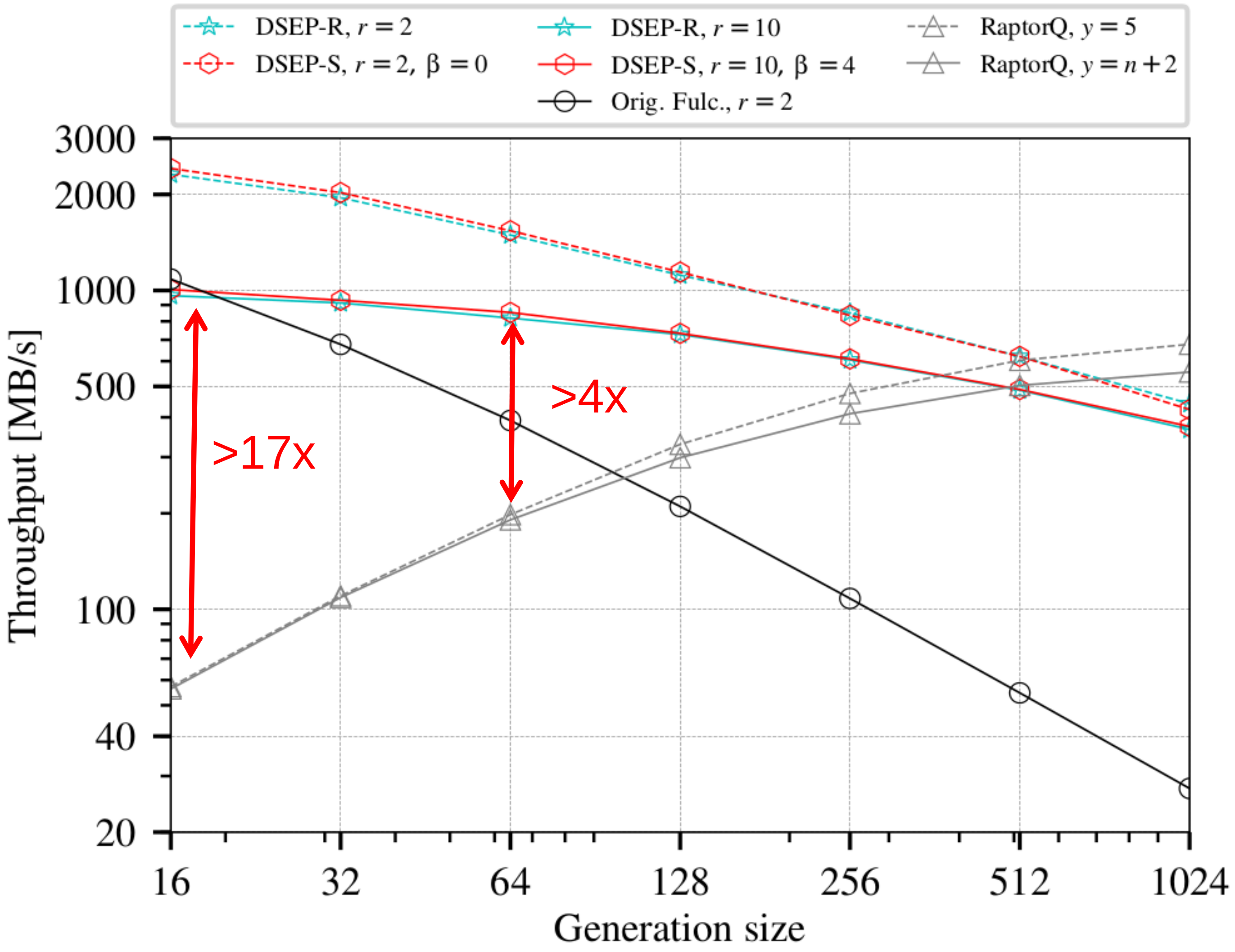
* Qualcomm. (2013, Dec.) Raptorq - the superior fec technology
Available: http://www.qualcomm.com/media/documents/raptorq-data-sheet

DSEP Fulcrum: results

y: number of repair symbols in RaptorQ

RaptorQ: using CodornicesRq (release 2.1)

Encoding throughput

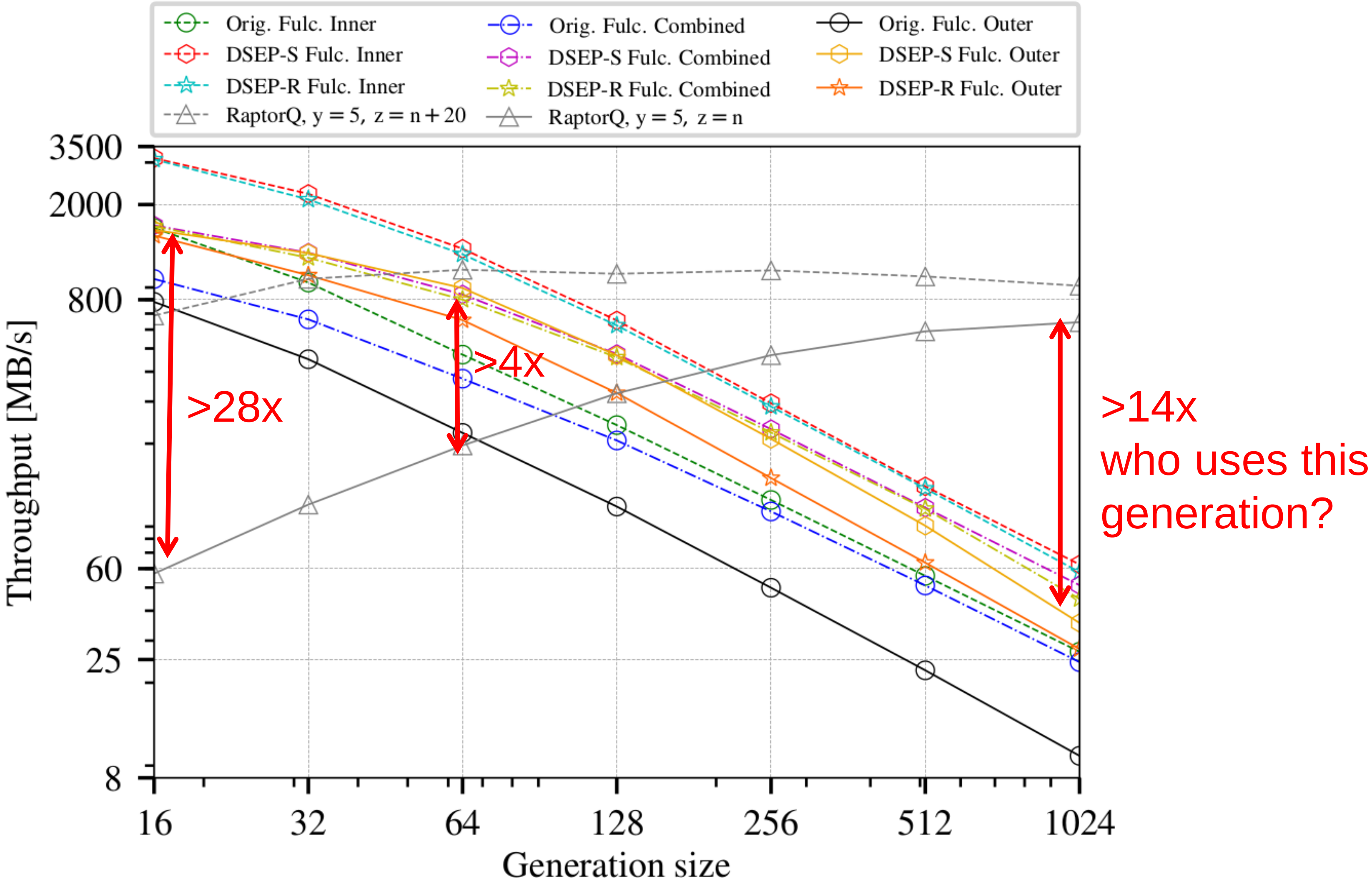


DSEP Fulcrum: results

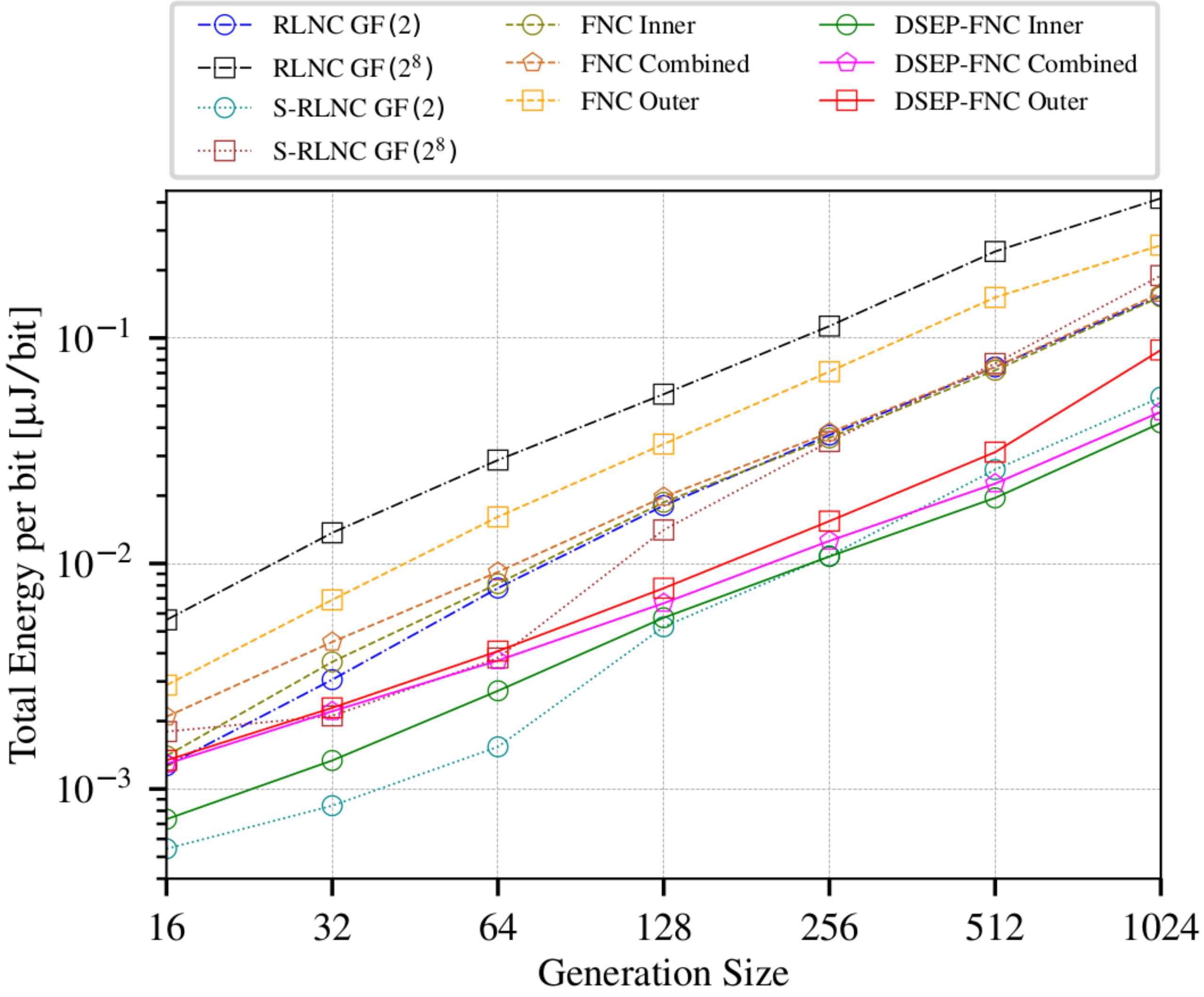
y: number of repair symbols in RaptorQ
z: received coded packets

RaptorQ: using CodornicesRq (release 2.1)

Decoding throughput



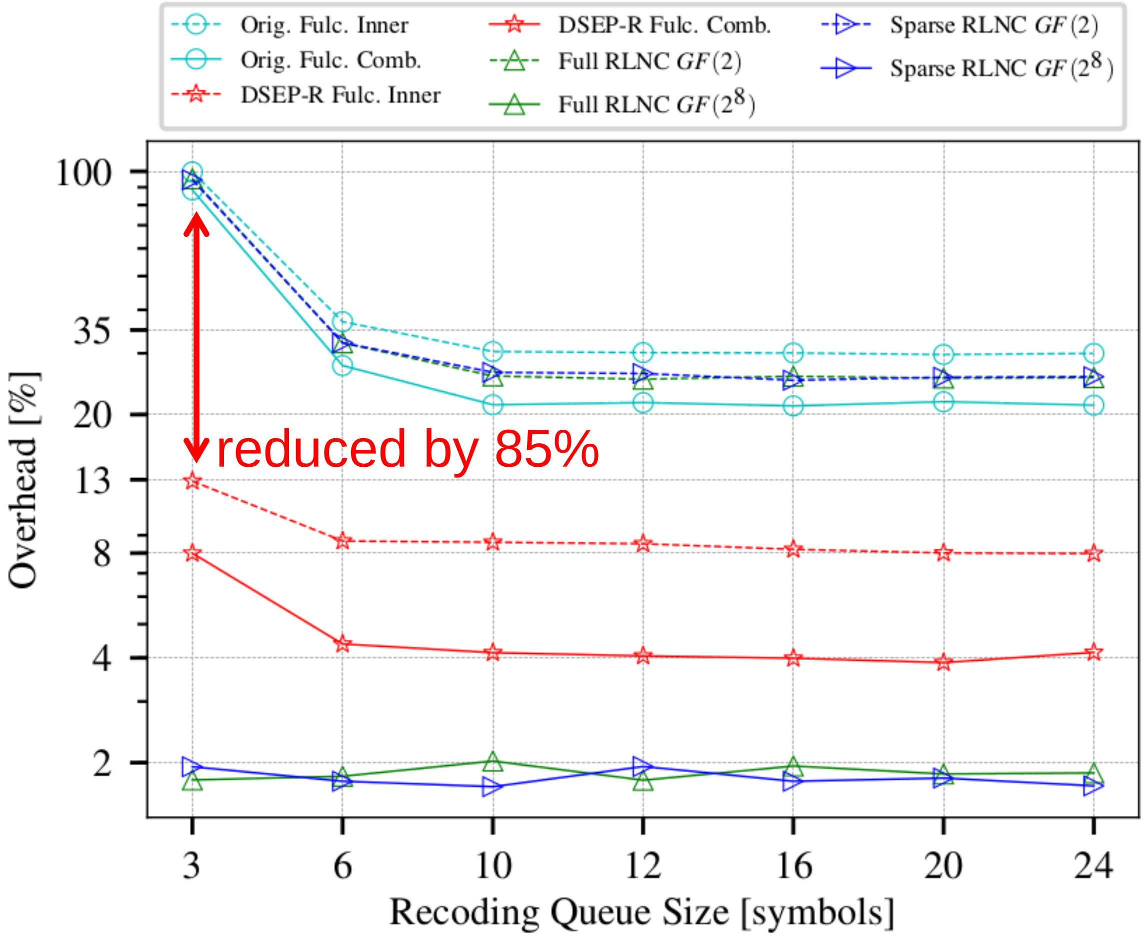
DSEP Fulcrum: results



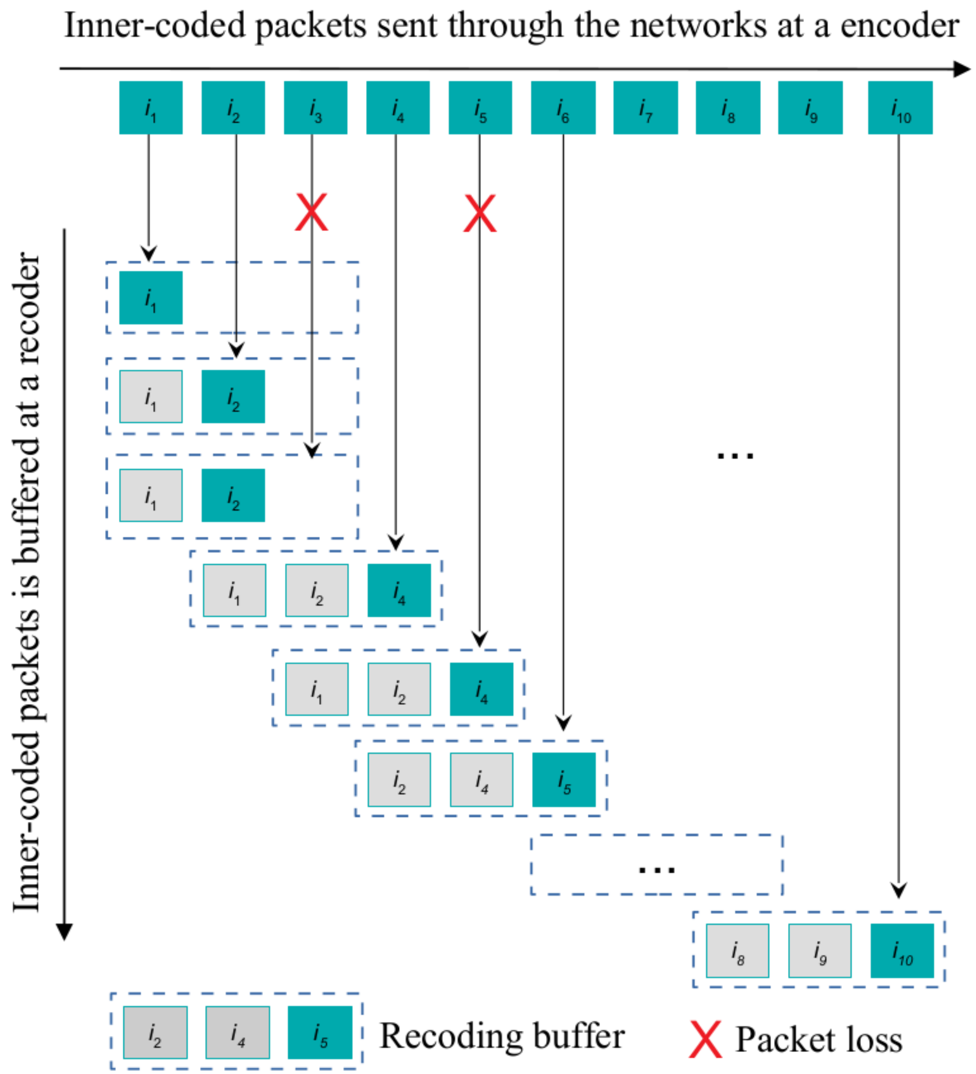
Energy consumption of Odroid-XU4 and Odroid-C2 devices.

DSEP Fulcrum: recoding mechanism

- The original recoding mechanism is not suitable for DSEP.
- It tends to cumulatively increase the coding density
- might create the linearly dependent coded packets
- faces the problem of limited memory at the intermediate nodes.
- waste the resources, if all the coded packets in a generation are stored at the intermediate nodes



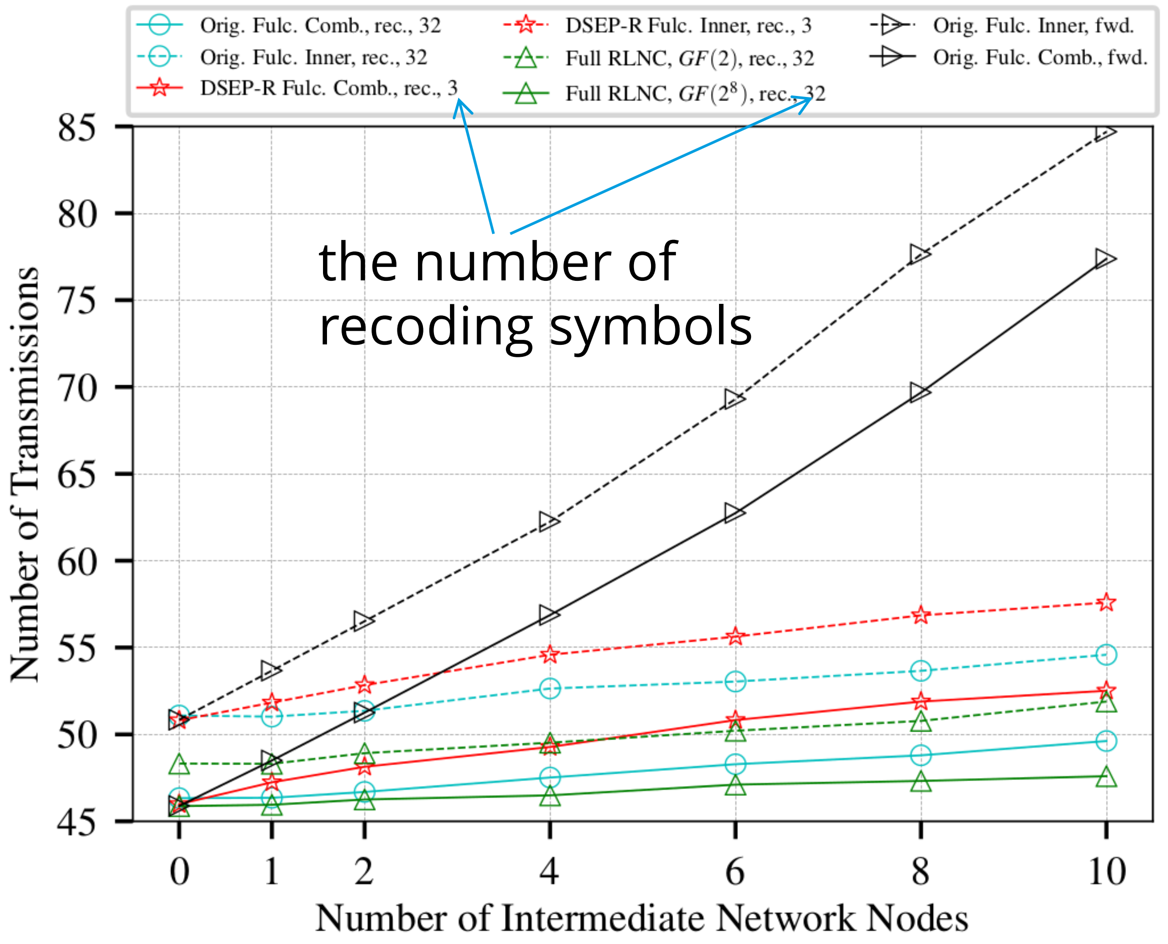
DSEP Fulcrum: recoding mechanism

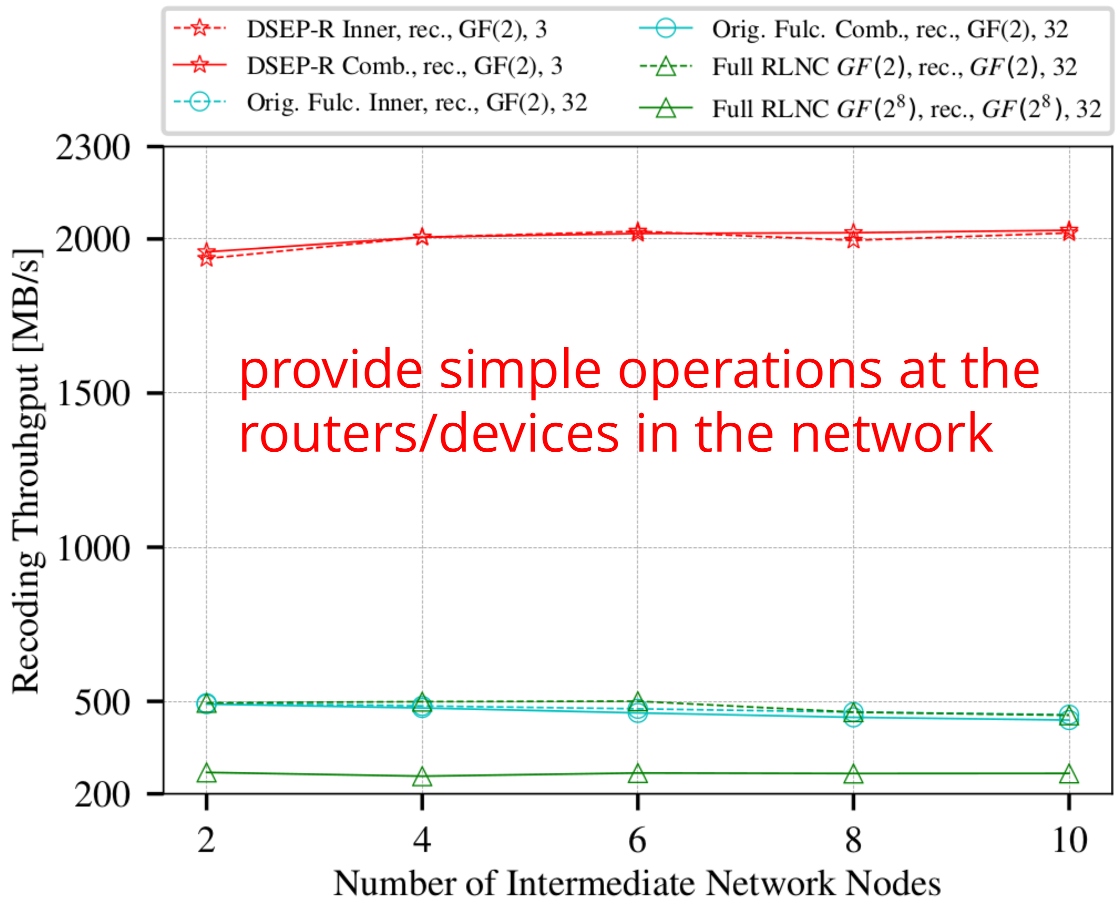


- DSEP recoding mechanism stores a few coded packets at recoder
- a new incoming coded packet is always combined with the rest ones.
- reduce the memory usage
- limit to cumulatively increase the coding density
- avoid to create the linearly dependent coded packets.

DSEP Fulcrum: recoding mechanism and results

The impact of recoding





Thank for your attention!