Benchmark - RLNC vs RaptorQ

Picking the right erasure correcting code (ECC) for a specific application is a challenging task. Different codes and implementations have different properties and characteristics. In this document we compare two popular ECC choices: the RLNC (Random Linear Network Codes) family of codes and RaptorQ.

RLNC vs RaptorQ - features

Compared to RLNC, Raptor codes provide extremely limited flexibility. In effect, only the number of source symbols can be selected, whereas RLNC can be highly tailored.

RLNC offers unique features that neither RaptorQ nor any conventional end-to-end codes offer, such as recoding, sliding window, on-the-fly encoding. RLNC is a rich family of codes that encompasses systematic, non-systematic, dense, sparse, perpetual, fulcrum, tunable codes. This wealth of choices means we can use the code in a flexible way, unlike RaptorQ or other conventional codes. We can instead optimize for different applications a great number of powerful code parameters, such as symbol size, symbols, underlying field(s), sparsity. RLNC can always match and generally outperform, in terms of throughput and decodability, any other existing code, through proper configuration. This rich set of features is key to performance in low latency/delay applications.

Code Capabilities	RaptorQ	RLNC	Benefits
Erasure correction	yes	yes	Corrects for lost data packets.
Generate unlimited coded symbols	yes	yes	Correct any number of lost packets.
Enables re-coding at all nodes in the network	no	yes	Multiplies benefit of coding in distributed networks such as wireless mesh.
On-the-fly adjustment of coding parameters	no	yes	Agile adaptation of the code to changing network conditions.
Adding redundancy before the entire block is present	no	yes	Reduces decoding delay for interactive sessions.
Support Heterogeneous devices	no	yes	Support multiple device types (phones, computers etc.) with a single stream.
Sliding window coding	no	yes	Low in order delay decoding for streaming.

Raptor vs RLNC performance

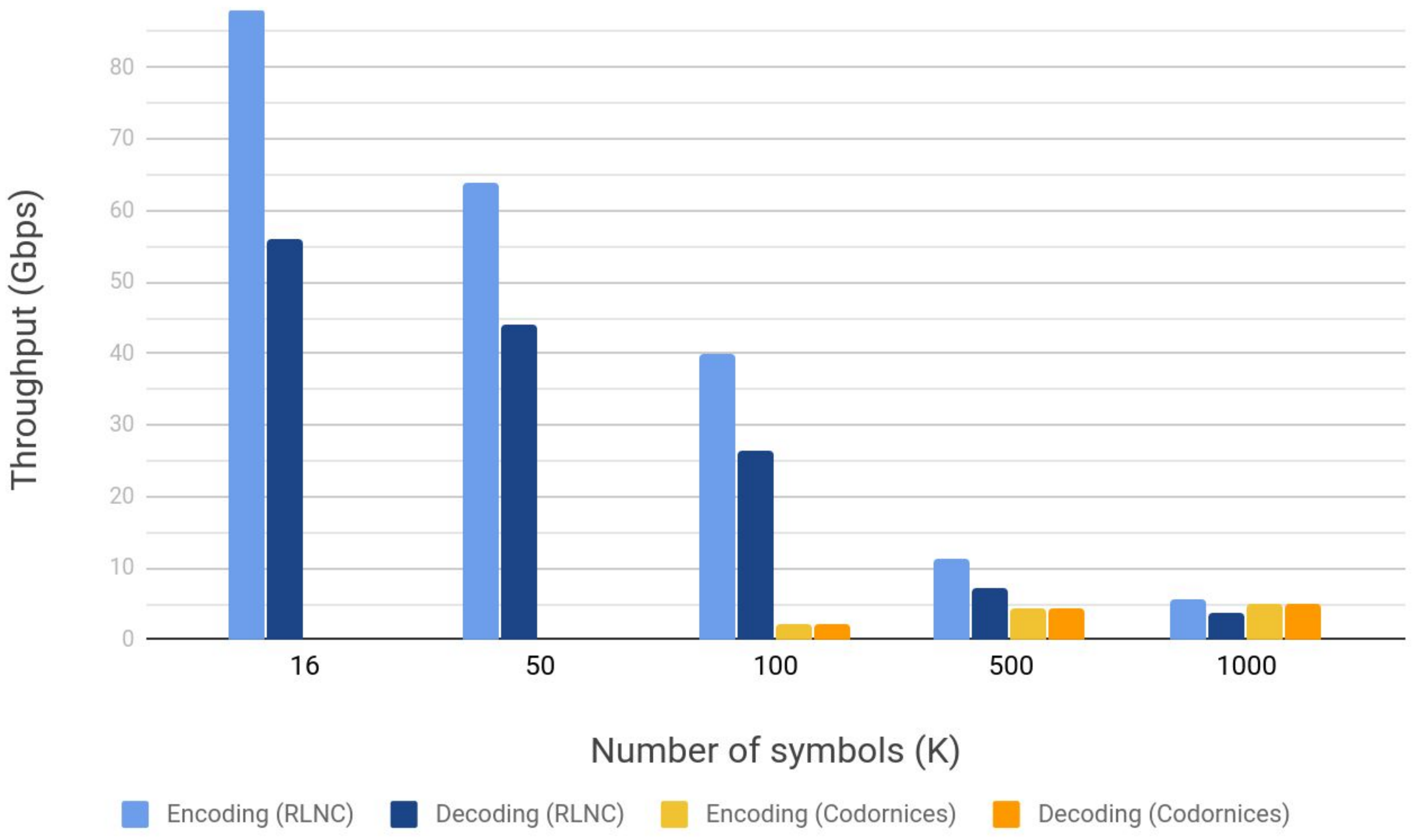
These benchmarks were run by Steinwurf ApS to measure the performance of the Kodo erasure coding library implementing Random Linear Network Coding (RLNC) codecs. Our results show that, for low delay applications, Kodo significantly outperforms Codornices RaptorQ.

All tests are performed over a block of size of K x 1280 bytes, containing K symbols, of 1280 bytes each. Encoding generates from each block the encoder a set of coded symbols. Decoding processed the coded symbols to reconstruct the original block of. The performance is measured in Gbps (Gigabits per second) as the size of the original block divided by the average time needed for encoding and decoding, respectively. The reported number is the average of 100 test runs. All benchmarks are single-threaded, i.e. they only utilize a single CPU core. Both encoding and decoding are parallelizable over multiple CPU cores - further boosting performance.

Results

Test System: Debian 10, Dell Optiplex / Intel i7-4770@3.4GHz, Systematic coding (10% packet loss):

Number of symbols (K)	Throughput (Gbps)	
	Encoding	Decoding
16	88	56
50	64	44
100	40	26.4
500	11.2	7.2
1000	5.68	3.68



We've included the performance numbers from Codornices RaptorQ (release 2) implementation for comparison are available here: https://www.codornices.info/performance. Observing the numbers reported for RaptorQ, performance of RLNC is equal when the number of symbols (K) is 1000 but RLNC is faster for K $\leq$ 500. Note that for low latency applications a smaller K is desirable to avoid latency building up in the erasure coding (e.g. 2 seconds of latency in the case of a 5 Mbit/s stream, symbol size of 1280 and 1000 symbols). Find more information about latency and erasure coding here: http://steinwurf.com/blog/2019-07-24-coding-for-low-latency-block-codes.html.