

Project	IEEE 802.16 Broadband Wireless Access Working Group < http://ieee802.org/16 >	
Title	Irregular Structured LDPC Codes	
Date Submitted	2004-08-17	
Source(s)	Victor Stolpman, Jianzhong (Charlie) Zhang, Nico van Waes Nokia 6000 Connection Drive Irving, TX 75039	Voice: 972-374-0958 Fax: 972-894-5937 victor.stolpman@nokia.com , charlie.zhang@nokia.com , nico.vanwaes@nokia.com
Re:	Recirculation ballot 14c (80216-04/50)	
Abstract	<p>In this document, we describe a structured approach to irregular LDPC code construction based on “seed” matrices that are expanded using permutation matrices for purposes of error correction control. These codes have small storage requirements with good block error rate performances over a wide range block sizes. Also described in this document is a structured approach to puncturing irregular LDPC codes facilitate rate-compatibility without having to modify the connective net in the encoder or decoder while still offering a wide range of code rates for link optimization.</p> <p>This document, while a full description of LDPC code construction by itself, currently contains no specific implementation language, as it is intended primarily as placeholder for ongoing harmonization efforts.</p>	
Purpose	Adoption of proposed text as optional feature.	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	

Patent
Policy and
Procedures

The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <<http://ieee802.org/16/ipr/patents/policy.html>>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <<mailto:chair@wirelessman.org>> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <<http://ieee802.org/16/ipr/patents/notices>>.

Irregular Structured LDPC Codes

Victor Stolpman, Jianzhong (Charlie) Zhang, Nico van Waes
Nokia

Introduction:

Modern communication systems use Forward Error Correction (FEC) codes in an attempt to convey information more reliably through channels with random events. One such FEC error control system use Low-Density Parity-Check (LDPC) codes because of error correcting capabilities that rival the performance of the so-called Turbo-Codes and for their applicability over a wide range of statistical channels [1,2,3]. In fact, some random irregular LDPC constructions based on edge ensemble designs have error correcting capabilities measured in Bit Error Rate (BER) that are within 0.05dB of the rate distorted Shannon limit for the AWGN channel [3].

Unfortunately, these random LDPC code constructions require long codeword constructions (on the order of 10^6 to 10^7 bits) in order to achieve these error rates, and despite good BER performance, these random code constructions often have poor Block Error Rate (BLER) performances. Hence, these random constructions do not lend themselves well to packet-based communication systems. Yet another disadvantage of random constructions based on edge distribution ensembles is for each codeword length another random construction is needed. Thus, communication systems employing variable block sizes (e.g. TCP/IP) would require multiple code definitions that could consume a significant amount of non-volatile memory storage for a large combination of codeword lengths and code rates.

An alternative to random LDPC construction is structured LDPC constructions that rely on a general algorithmic approach to constructing LDPC matrices and require much less non-volatile memory than random constructions. Thus, the problem is to design irregular structured LDPC codes that have good overall error performance (both BER and BLER) for a wide range of code rates and block sizes with attractive storage requirements. The result of such LDPC codes is a better performing communication system with lower cost terminals. These factors make such a FEC attractive for application over a wide range of products including but not limited to IEEE802.16 and IEEE802.11n compliant products.

Thus, this exact description below for LDPC code construction succeeds at solving the above said problem while out performing competing solutions [4] in overall error performance for similar block sizes and iterations without the non-volatile memory requirements of randomly constructed codes.

Seed Matrix, Spreading Matrices and the Expanded LDPC Matrix:

In this section, we describe and define the binary “seed” parity-check matrix \mathbf{H}_{SEED} of dimension $((N_{\text{SEED}} - K_{\text{SEED}}) \times N_{\text{SEED}})$ with the code rate defined as $K_{\text{SEED}}/N_{\text{SEED}}$. Simply, the seed matrix \mathbf{H}_{SEED} is a starting matrix used to generate expanded LDPC parity-check matrices \mathbf{H} through a spreading process. In the “Simulation Results” section, we specifically define three seed matrices for code rates 1/2, 2/3, and 3/4 of dimensions (26×52) , (20×61) , and (14×56) respectively that solves the earlier stated problem definition. Also it is important to note that these given seed matrices are nearly upper triangular, and as a result all expanded LDPC matrices using these seeds in the following described approach will also have a nearly upper triangular construction that lend themselves well for encoding purposes (i.e. near linear encoding complexity with respect to codeword length) [5].

The purpose of these seed matrices is to identify the location and type of sub-matrices in the expanded LDPC parity-check matrix \mathbf{H} constructed from expanding \mathbf{H}_{SEED} with a given set of spreading matrices where all

elements of the set are of dimension $(N_{\text{SPREAD}} \times N_{\text{SPREAD}})$. The ones in \mathbf{H}_{SEED} determine the location of sub-matrices in the expanded matrix \mathbf{H} that contain a spreading matrix from the following set of matrices

$$\left\{ \mathbf{P}_{\text{SPREAD}}^{\infty}, \mathbf{P}_{\text{SPREAD}}^0, \mathbf{P}_{\text{SPREAD}}^1, \mathbf{P}_{\text{SPREAD}}^2, \dots, \mathbf{P}_{\text{SPREAD}}^{p-1} \right\}$$

where p is a positive prime integer (to be define specifically later), $\mathbf{P}_{\text{SPREAD}}^{\infty}$ denotes the all zeros matrix $\mathbf{0}$ (i.e. $\mathbf{P}_{\text{SPREAD}}^{\infty} = \mathbf{0}$ where every matrix element is a zero), $\mathbf{P}_{\text{SPREAD}}^0 = \mathbf{I}$ is the identity matrix, $\mathbf{P}_{\text{SPREAD}}^1$ is a full-rank permutation matrix, $\mathbf{P}_{\text{SPREAD}}^2 = \mathbf{P}_{\text{SPREAD}}^1 \mathbf{P}_{\text{SPREAD}}^1$, and so on in like fashion up to $\mathbf{P}_{\text{SPREAD}}^{p-1} = (\mathbf{P}_{\text{SPREAD}}^1)^{p-1}$. Specifically, we define $\mathbf{P}_{\text{SPREAD}}^1$ here as the single circular-shift permutation matrix, e.g. for $N_{\text{SPREAD}} = 5$

$$\mathbf{P}_{\text{SPREAD}}^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The zeros in \mathbf{H}_{SEED} indicate the location of the sub-matrix $\mathbf{P}_{\text{SPREAD}}^{\infty} = \mathbf{0}$ in the expanded matrix \mathbf{H} . Thus, the expanded LDPC matrix \mathbf{H} is of dimension $((N - K) \times N)$ where $N = N_{\text{SPREAD}} N_{\text{SEED}}$ and $K = N_{\text{SPREAD}} K_{\text{SEED}}$ with sub-matrices consisting of permutation matrices of dimension $(N_{\text{SPREAD}} \times N_{\text{SPREAD}})$ raised to an exponential power from the set of $\{0, 1, \dots, p-1, \infty\}$, or simply in matrix form

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}_{\text{SPREAD}}^{F_{1,1}} & \mathbf{P}_{\text{SPREAD}}^{F_{1,2}} & \mathbf{L} & \mathbf{P}_{\text{SPREAD}}^{F_{1,N_{\text{SEED}}}} \\ \mathbf{P}_{\text{SPREAD}}^{F_{2,1}} & \mathbf{P}_{\text{SPREAD}}^{F_{2,2}} & \mathbf{L} & \mathbf{P}_{\text{SPREAD}}^{F_{2,N_{\text{SEED}}}} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{P}_{\text{SPREAD}}^{F_{(N_{\text{SEED}}-K_{\text{SEED}}),1}} & \mathbf{P}_{\text{SPREAD}}^{F_{(N_{\text{SEED}}-K_{\text{SEED}}),2}} & \mathbf{L} & \mathbf{P}_{\text{SPREAD}}^{F_{(N_{\text{SEED}}-K_{\text{SEED}}),N_{\text{SEED}}}} \end{bmatrix}$$

where $F_{i,j} \in \{0, 1, \dots, p-1, \infty\}$ for $i = 1, 2, \dots, K$, $(N_{\text{SEED}} - K_{\text{SEED}})$ and $j = 1, 2, \dots, N_{\text{SEED}}$. In matrix form, the final exponential matrix is

$$\mathbf{F} = \begin{bmatrix} F_{1,1} & F_{1,2} & \mathbf{L} & F_{1,N_{\text{SEED}}} \\ F_{2,1} & F_{2,2} & \mathbf{L} & F_{2,N_{\text{SEED}}} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ F_{(N_{\text{SEED}}-K_{\text{SEED}}),1} & F_{(N_{\text{SEED}}-K_{\text{SEED}}),2} & \mathbf{L} & F_{(N_{\text{SEED}}-K_{\text{SEED}}),N_{\text{SEED}}} \end{bmatrix}$$

and is of same dimension as the \mathbf{H}_{SEED} , $((N_{\text{SEED}} - K_{\text{SEED}}) \times N_{\text{SEED}})$. The selection of the $F_{i,j}$ matrix elements of the set $\{0, 1, \dots, p-1, \infty\}$ uses \mathbf{H}_{SEED} and is described further in the following section.

Exponent Matrix Construction:

Specifically, let p be defined as the smallest prime integer that satisfies both $N_{\text{SEED}} + 2 \leq p$ and $N_{\text{SPREAD}} \leq p$. We then define an interim exponent array matrix as

$$\mathbf{E}_{\text{EXPONENT}} = \begin{bmatrix} E_{2,2} & E_{2,3} & E_{2,4} & \text{L} & E_{2,(N_{\text{SEED}}-K_{\text{SEED}})} & \text{L} & E_{2,(N_{\text{SEED}}+1)} \\ E_{3,1} & E_{3,2} & E_{3,3} & \text{L} & E_{3,(N_{\text{SEED}}-K_{\text{SEED}}-1)} & \text{L} & E_{3,N_{\text{SEED}}} \\ \infty & E_{4,1} & E_{4,2} & \text{L} & E_{4,(N_{\text{SEED}}-K_{\text{SEED}}-2)} & \text{L} & E_{4,(N_{\text{SEED}}-1)} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \infty & \infty & \infty & \text{L} & E_{(N_{\text{SEED}}-K_{\text{SEED}}+1),1} & \text{L} & E_{(N_{\text{SEED}}-K_{\text{SEED}}+1),(K_{\text{SEED}}+2)} \end{bmatrix}$$

where $E_{i,j} = (i-1)(j-1) \bmod p$.

Finally, we construct the exponential matrix \mathbf{F} used in the expanded LDPC matrix \mathbf{H} by replacing each one in \mathbf{H}_{SEED} with the corresponding matrix element (i.e. same row and column) in the interim exponent matrix $\mathbf{E}_{\text{EXPONENT}}$, and we replace each zero in \mathbf{H}_{SEED} with ∞ . Thus, the elements of \mathbf{F} belong to the set $\{0,1,\dots,p-1,\infty\}$ when using modulo- p arithmetic in the construction of $\mathbf{E}_{\text{EXPONENT}}$.

A Small Construction Example:

For example purposes only, let

$$\mathbf{H}_{\text{SEED}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \text{ thus } N_{\text{SEED}} = 6,$$

and let

$$\mathbf{P}_{\text{SPREAD}}^1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \text{ thus } N_{\text{SPREAD}} = 3,$$

so $p = 11$ is the smallest prime number that satisfies the specified conditions $N_{\text{SEED}} + 2 \leq p$ and $N_{\text{SPREAD}} \leq p$. Then, the interim and final exponent matrices as defined above are

$$\mathbf{E}_{\text{EXPONENT}} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 2 & 4 & 6 & 8 & 10 \\ \infty & 0 & 3 & 6 & 9 & 1 \\ \infty & \infty & 0 & 4 & 8 & 1 \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} 1 & \infty & \infty & 4 & \infty & \infty \\ 0 & 2 & \infty & 6 & 8 & \infty \\ \infty & 0 & 3 & \infty & \infty & 1 \\ \infty & \infty & 0 & \infty & 8 & 1 \end{bmatrix},$$

and the expanded LDPC matrix is

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Structured Puncturing:

In this section, we introduce a “seed” puncture-degree sequence $\mathbf{d}_{\text{SEED},1}$ of dimension $(L_1 \times 1)$

$$\mathbf{d}_{\text{SEED},1} = \left[d_1^{(1)} \quad d_2^{(1)} \quad \dots \quad d_{L_1}^{(1)} \right]$$

where each element $d_i^{(1)}$ indicates the degree of a variable node corresponding to a codeword element to be punctured in the code defined by \mathbf{H}_{SEED} . We expand this seed puncture sequence to create an expanded puncture-degree sequence $\mathbf{p}_{\text{DEGREE}}$ of dimension $\left(\left(\frac{L_1 N}{N_{\text{SEED}}} \right) \times 1 \right)$ that contain the variable-degrees corresponding to the columns of the expanded parity-check matrix \mathbf{H} of dimension $\left((N - K) \times N \right)$

$$\mathbf{p}_{\text{DEGREE}}^T = \mathbf{d}_{\text{SEED},1}^T \otimes \mathbf{1}_{\left(1 \times \left(\frac{N}{N_{\text{SEED}}} \right) \right)} = \left[p_1 \quad p_2 \quad \dots \quad p_{\left(\frac{L_1 N}{N_{\text{SEED}}} \right)} \right]$$

where each element p_i indicates the degree of a variable node corresponding to a codeword element to be punctured in the expanded code defined by \mathbf{H} and \otimes represents the Kronecker product. To offer more flexibility in degree selection, multiple seed sets can be used $\mathbf{d}_{\text{SEED},m}$ of possible different dimensions $(L_m \times 1)$ for $m = 1, 2, \dots, K, M$

$$\mathbf{d}_{\text{SEED},m} = \left[d_1^{(m)} \quad d_2^{(m)} \quad \dots \quad d_{L_m}^{(m)} \right] \text{ for } m = 1, 2, \dots, K, M$$

In order to achieve a particular effective code rate from a mother code \mathbf{H} of dimension $((N - K) \times N)$, say

$$R_{\text{EFF}} = \frac{K}{N - P} \text{ where } P \in \{0, 1, 2, \dots, K, (L_m N / m N_{\text{SEED}})\} \text{ for } m = 1, 2, \dots, M,$$

we use the very first P elements of $\mathbf{v}_{\text{NODE}}^T$ (i.e. $[v_1 \ v_2 \ \dots \ v_P]$) to puncture code word elements. This nested structure to puncturing, reduces storage complexity and can achieve all the possible code rates, i.e.

$$R_{\text{EFF}} \in \left\{ \frac{K}{N}, \frac{K}{N-1}, \frac{K}{N-2}, \dots, K, \frac{K}{K+1}, \frac{K}{K} \right\}$$

provided \mathbf{v}_{NODE} is long enough. By puncturing codeword elements, there is no need to change the connective net for multiple code rates given that the effective code rate is within the mother codes capability (i.e. implementation friendly). Of course, there is a useful range of code rates of which outside performance will suffer as with most punctured error correction codes.

Rate 2/3 Irregular Seed Parity-Check Matrix:

$$\mathbf{H}_{\text{SEED},61} = \begin{bmatrix}
 10000000000000000000101000000000001000000000010100001010011101 \\
 11000000000000000000000000000000100010000010000010000000011011100010 \\
 0110000000000000000000000000000010001000000100000100010000110011100 \\
 00110000000000000000000000000000101001000000010000100000001001100100001 \\
 000110000000000000000000000000001000001000000000100001000010011001001010 \\
 00001100000000000000000000000000100000000000001000000000001000001011010111 \\
 000001100000000000000000000000001000010000100000000000000101110100101 \\
 0000001100000000000000000000000010000000000000010010000000010110101110 \\
 000000011000000000000000000000001100000010010000000000001000011010011 \\
 000000001100000000000000000000001000001000000100000000000000110101010101 \\
 00000000011000000000000000000000100010000100010000010000100001000101011 \\
 0000000000110000000000000000000010000000010000000000000011011011101 \\
 0000000000011000000000010001000000000001000000100000101101011100 \\
 000000000000110000000000000000001101001000001100000000101101000 \\
 00000000000001100000000010010010000000001000100000010001011010 \\
 0000000000000011000010010000000010001000000000000010110110010 \\
 0000000000000001100001000000100000100100000001000011001100001 \\
 0000000000000000110000000000000010000001000000111110000010110 \\
 0000000000000000011110000100100000000000011000000001101100000 \\
 0000000000000000001100000000000010000001100010000000100011111
 \end{bmatrix}$$

Rate 3/4 Irregular Seed Parity-Check Matrix:

$$\mathbf{H}_{\text{SEED},56} = \begin{bmatrix} 10000000000001001000000001001111000000000100100110111010 \\ 1100000000000000000110001110000000100000000000001011011111 \\ 0110000000000000000100010000000100000010000001111101111 \\ 001100000000000000000100000001100000100000010000111111111 \\ 000110000000000000001001010100010001000010000100010101100101 \\ 000011000000000100000001000001000001000100001001111011110 \\ 00000110000001000100000010100010010011000000001110110001 \\ 000000110000000000000101001000000001000010000001011111111 \\ 00000001100000100010010010010100000101000011001100000011 \\ 00000000110000010001000000000001110000010000001111111010 \\ 00000000011000001100000000000000001110101000011001101101 \\ 00000000001100010000100100100010000001110000000101010111 \\ 00000000000110000001000000010000000000001110001111111110 \\ 00000000000011010000010001000000100000000001111010111101 \end{bmatrix}$$

Simulation Results:

The proposed codes are evaluated and compared with Samsung's structured codes [4]. The results are organized into two subsections. The first subsection includes the results for a minimal set of length $n=1152$, rate $r = 1/2$ code, $n = 1728$, $r=2/3$ code and $n = 2304$, $r= 3/4$ code. The second subsection includes the additional results required for the structured codes, including, $r=1/2$, $n=576$, 1728 , 2304 codes; $r = 2/3$, $n= 576$, 1152 , 2304 codes and $r = 3/4$, $n = 576$, 1152 , 1728 codes. All simulations used 50 iterations of the conventional Sum-Product-Algorithm (SPA).

Since our current seed matrix sizes are not exact multiples of 48 (this can easily be changed in future version under this construction scheme) our codes do not match the above "nominal" code rates and block sizes at the same time. Instead, here we present two sets of codes with rates/code word lengths close to these nominal values. The first set matches the code rate requirement, and they are represented by the red curves in the plots. These are designed to match the nominal code rates at the expense of slightly longer or shorter codeword lengths. The other set matches the codeword length requirement, and they are represented by the black curves in the plots. These used a mother code exceeding the length requirement and punctured to match the nominal code word lengths while allowing for slightly higher deviations from the nominal code rate.

At the end, additional simulation results for various punctured sets are included as to show demonstrate the rate-compatibilities of these codes. These sets are constructed via structured puncturing use an irregular structured mother code generated from the seed matrices and a puncture-degree sequence.

Minimum Set of Simulations

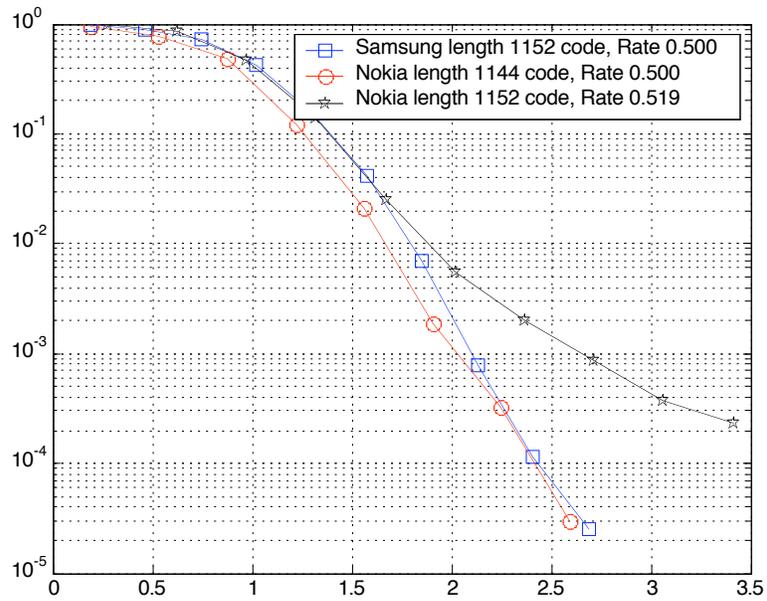


Figure 1. Performance of $n=1152, r=1/2$ code. AWGN channel, BPSK.

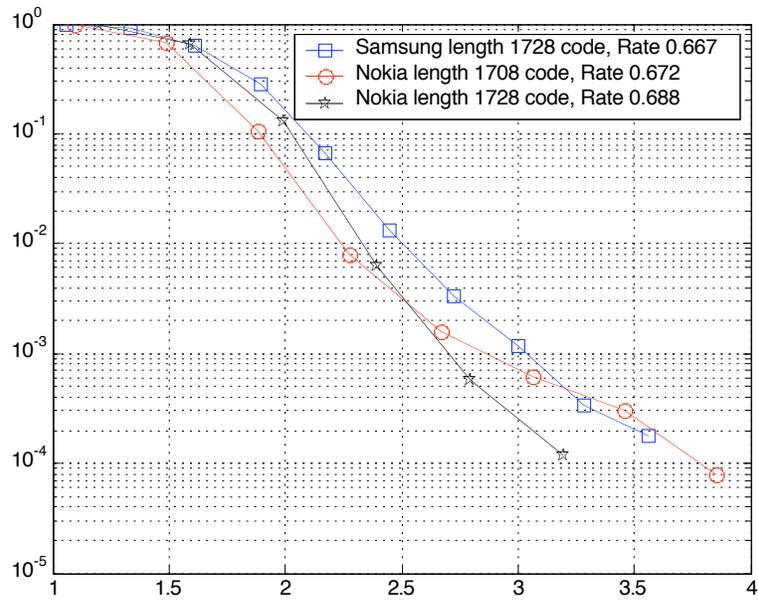


Figure 2. Performance of $n=1728, r=2/3$ code. AWGN channel, BPSK.

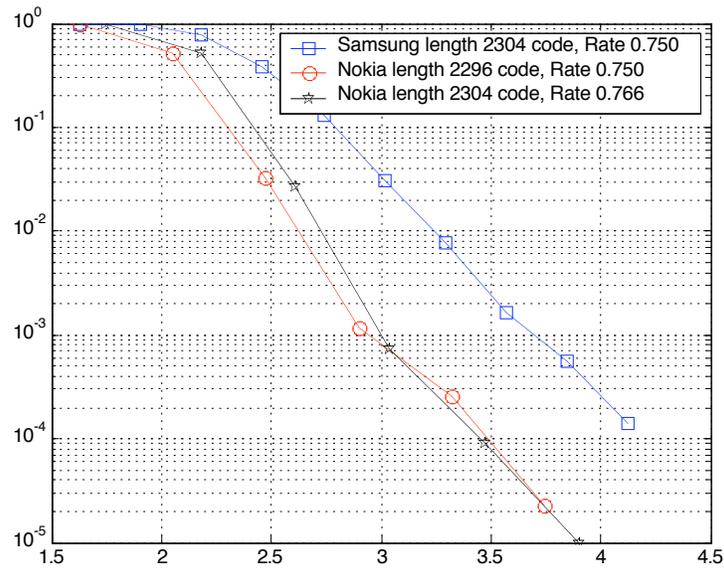


Figure 3. Performance of $n=2304$, $r = 3/4$ code. AWGN channel, BPSK.

Additional Set of Simulations
 $r = 1/2$, $n = 576, 1728, 2304$

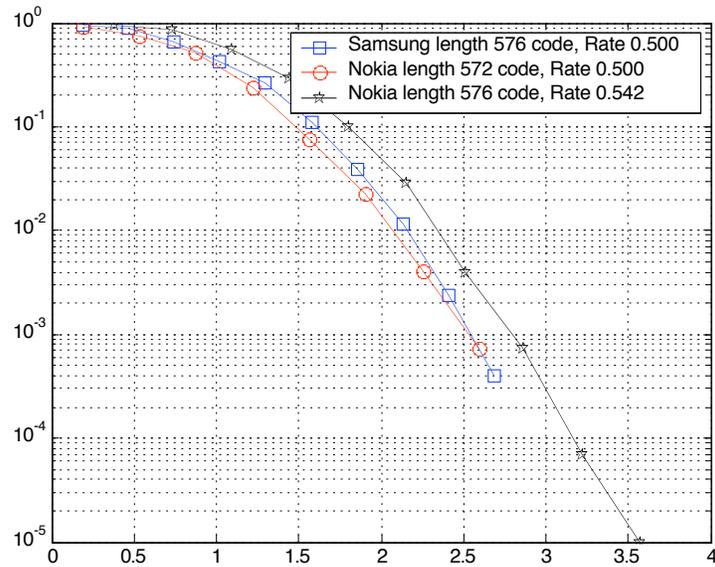


Figure 4. Performance of $n=576$, $r = 1/2$ code. AWGN channel, BPSK.

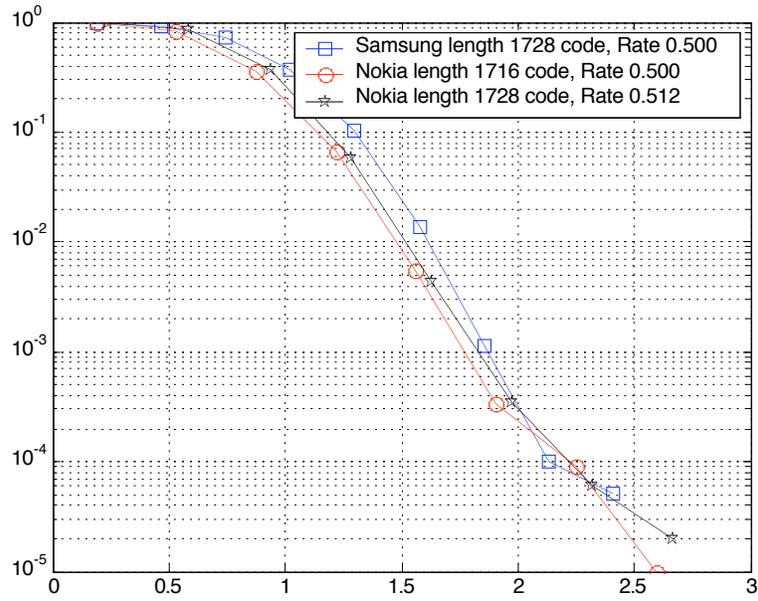


Figure 5. Performance of $n=1728$, $r = 1/2$ code. AWGN channel, BPSK.

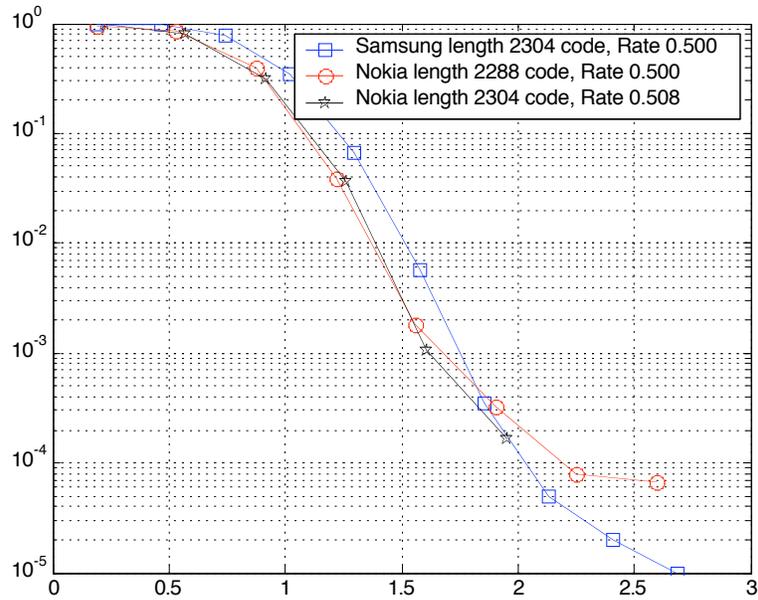


Figure 6. Performance of $n=2304$, $r = 1/2$ code. AWGN channel, BPSK.

$r = 2/3$, $n = 576, 1152, 2304$

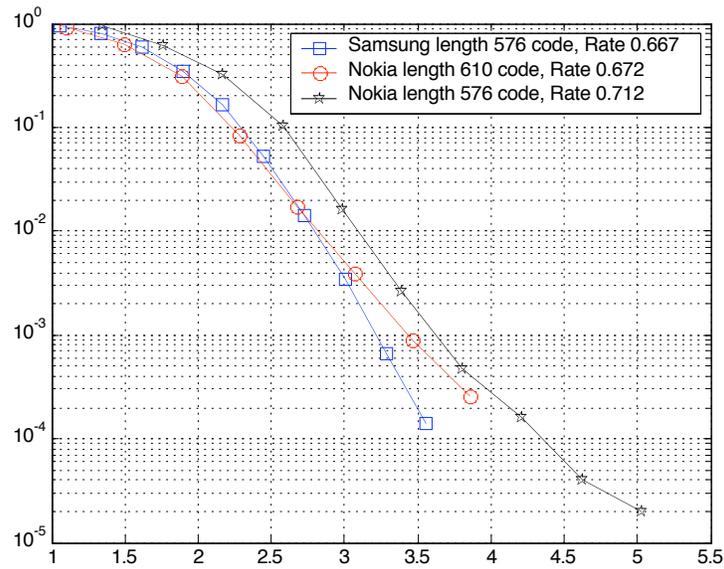


Figure 7. Performance of $n=576$, $r = 2/3$ code. AWGN channel, BPSK.

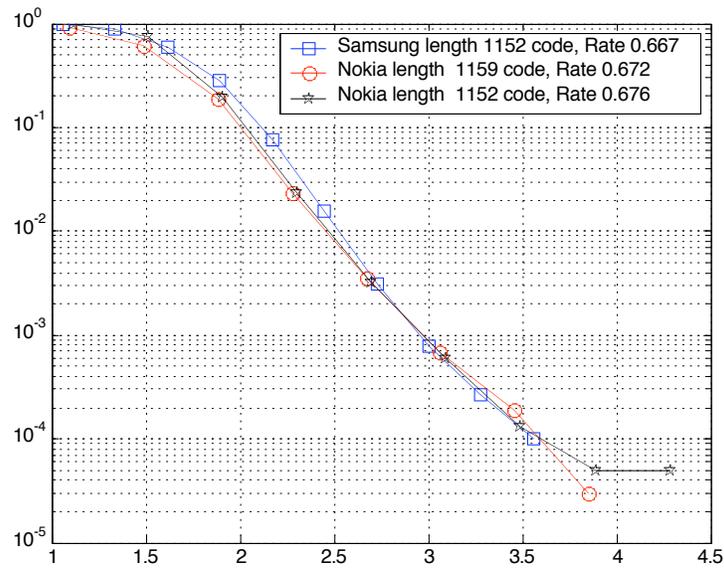


Figure 8. Performance of $n=1152$, $r = 2/3$ code. AWGN channel, BPSK.

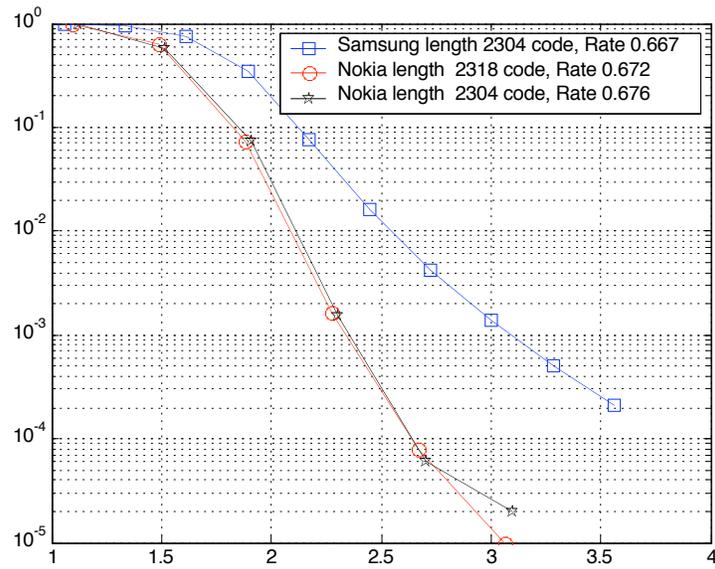


Figure 9. Performance of $n=2304$, $r = 2/3$ code. AWGN channel, BPSK.

$r = 3/4$, $n = 576, 1152, 1728$

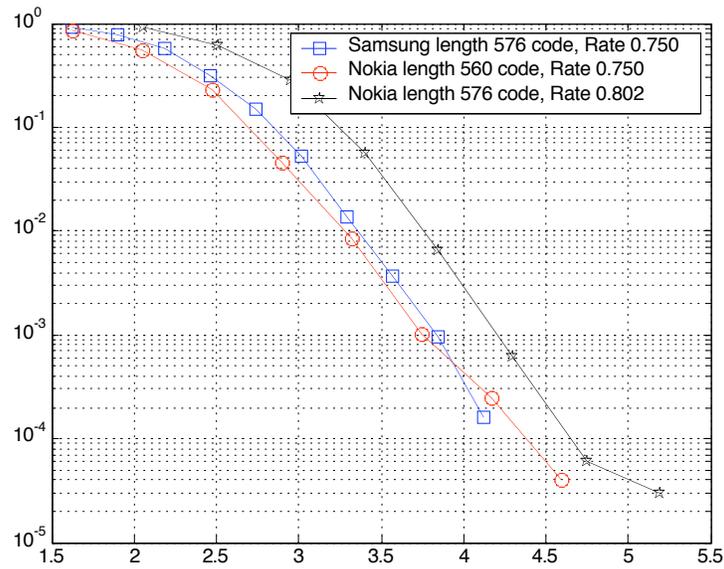


Figure 10. Performance of $n=576$, $r = 3/4$ code. AWGN channel, BPSK.

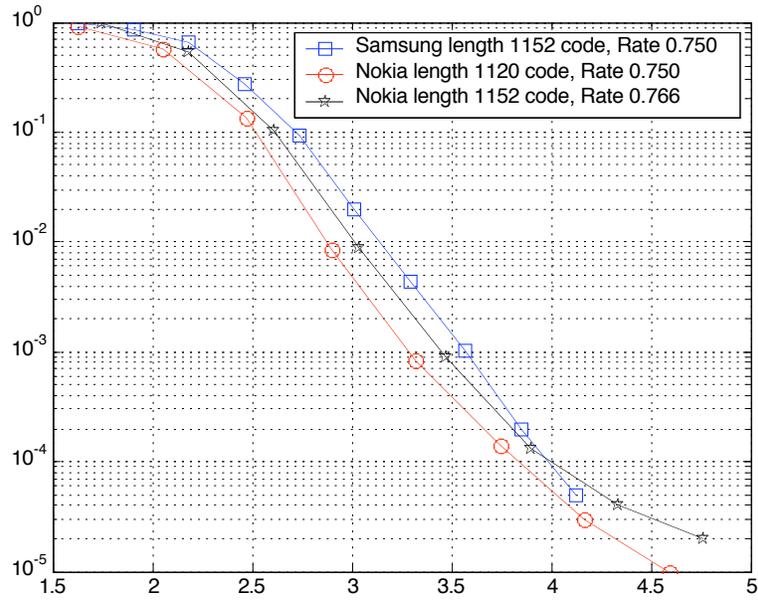


Figure 11. Performance of $n=1152$, $r = 3/4$ code. AWGN channel, BPSK.

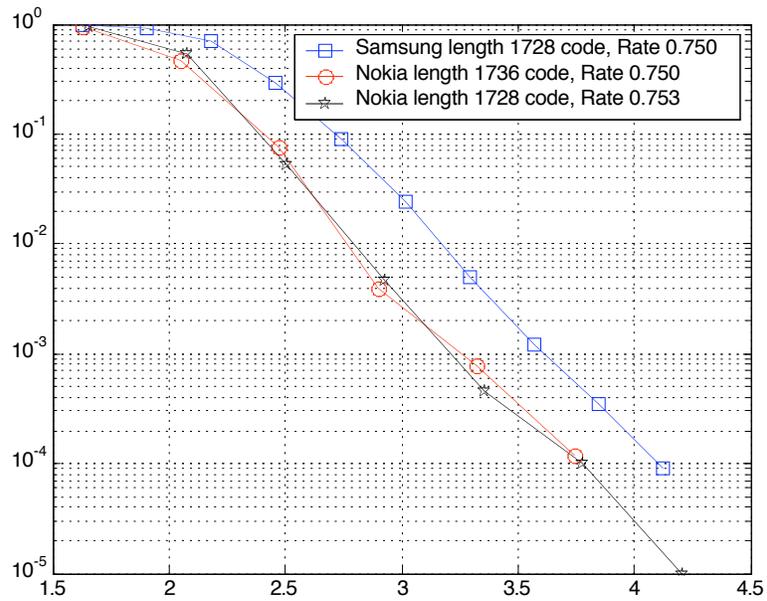


Figure 12. Performance of $n=1728$, $r = 3/4$ code. AWGN channel, BPSK.

Rate 2/3 Mother Code Example:

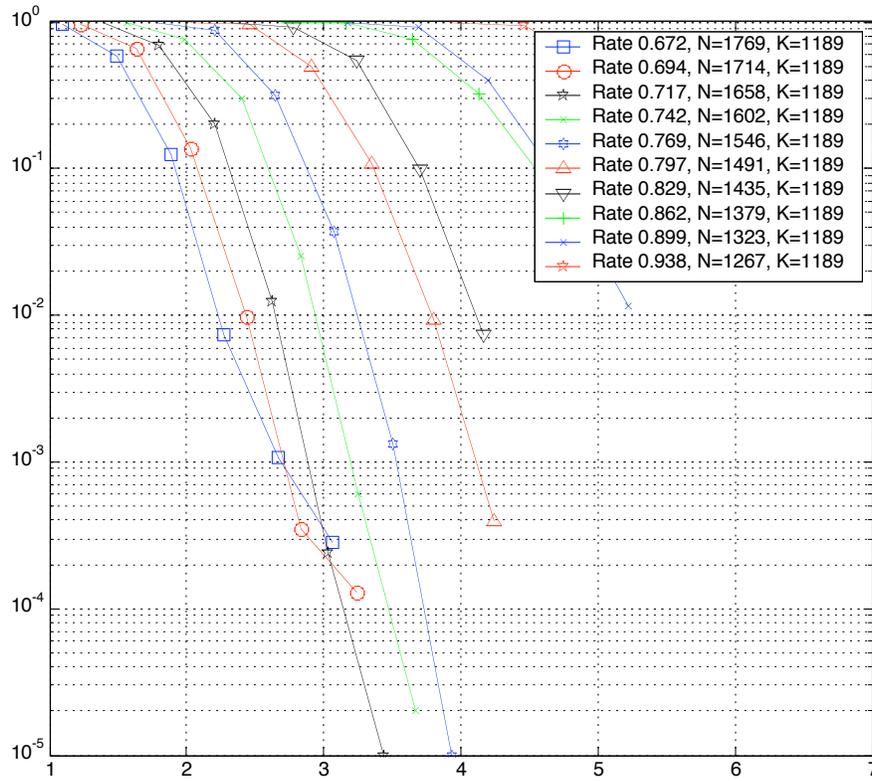


Figure 14: Rate 2/3 mother code using $N_{\text{SPREAD}} = 29$ and the following nested seed puncture-degree sequence

$$\mathbf{d}_{\text{SEED},1} = [10, 10, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 2, 2, 2, 2, 10]^T$$

Rate 2/3 Mother Code Example:

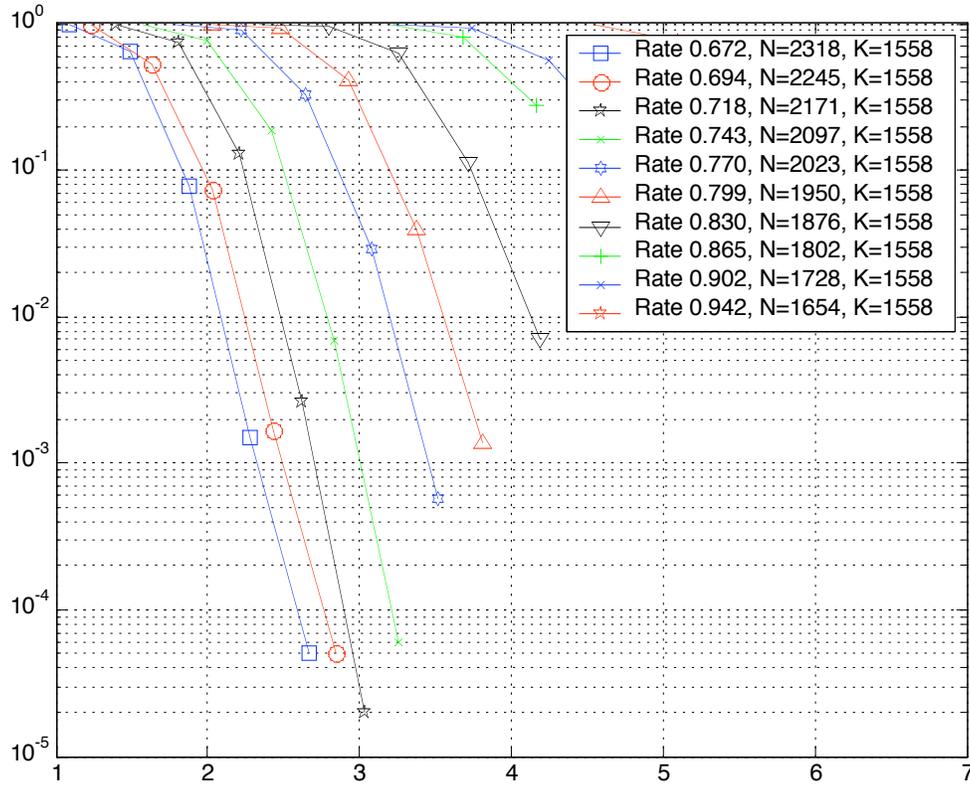


Figure 15: Rate 2/3 mother code using $N_{\text{SPREAD}} = 38$ and the following nested seed puncture-degree sequence

$$\mathbf{d}_{\text{SEED},2} = [10, 10, 10, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 3, 3, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 10, 10, 3]^T$$

Rate 3/4 Mother Code Example:

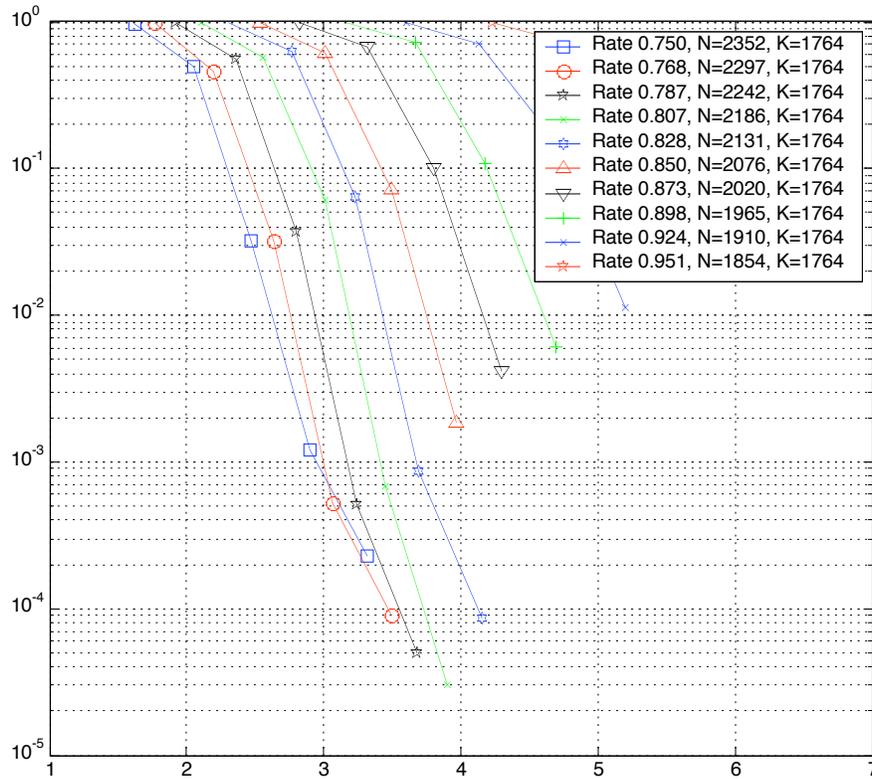


Figure 16: Rate 3/4 mother code using $N_{\text{SPREAD}} = 42$ and the following nested seed puncture-degree sequence.

$$\mathbf{d}_{\text{SEED},3} = [10, 10, 10, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 2, 10, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 10, 2]^T$$

References:

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp.21-28, Jan. 1962.
- [2] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of Capacity Approaching Irregular Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619-637, Feb. 2001.
- [3] Sae-Young Chung, *On the Construction of Some Capacity-Approaching Coding Schemes*, PhD Dissertation, MIT, 2000.
- [4] P. Joo, et al., "LDPC coding for OFDMA PHY," IEEE C802.16d-04/86r1, <http://iee802.org/16>, May 2004.
- [5] T. J. Richardson and R. L. Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 638-656, Feb. 2001.