Short-Blocklength Non-Binary LDPC Codes with Feedback-Dependent Incremental Transmissions

Kasra Vakilinia, Tsung-Yi Chen*, Sudarsan V. S. Ranganathan, Adam R. Williamson,
Dariush Divsalar**, and Richard D. Wesel
Department of Electrical Engineering, University of California, Los Angeles, Los Angeles, California 90095
*Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60208
**Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

***Abstract*—The main advantage of feedback in a point-to-point memoryless channel is the reduction of the average blocklength required to approach capacity. This paper presents a communication system with feedback that uses carefully designed non-binary LDPC (NB-LDPC) codes and incremental transmissions to achieve $92-94\%$ of the idealized throughput of rate-compatible sphere-packing with maximum-likelihood decoding (RCSP-ML) for average blocklengths of 150-450 bits. The system uses active feedback by carefully selecting each bit of additional incremental information to improve the reliability of the least reliable variable node. The system uses post processing in the decoder to further improve performance. The average blocklengths of 150-450 bits are small enough that feedback provides a throughput advantage but also large enough that overhead that might be associated with transmitter confirmation is more easily tolerated.**

I. Introduction

Polyanskiy et al. [1] and Chen et al. [2] illustrated that by using feedback, one can approach capacity in a small number of channel uses (low latency). Polyanskiy et al. [1] introduced variable-length coding with termination (VLFT) which theoretically approaches capacity at low latencies. In VLFT, the receiver provides full noiseless feedback to the transmitter. The transmitter consequently sends additional information over the channel until it determines that the decoder has correctly decoded the message. Termination, the "T" in VLFT, indicates that when the receiver has decoded correctly, the transmitter sends a noiseless transmitter confirmation (NTC) to terminate the transmission. This termination takes place through a control channel apart from of the primary communication channel.

The classical results from [3] show that feedback does not increase the asymptotic capacity of memory-less channels. Polyanskiy et al. [4] illustrated that in a non-feedback communication system, the maximum achievable throughput is significantly lower for short blocklengths of up to several hundred bits. They later showed [1] that with feedback the maximum achievable throughput can be greatly improved at short blocklengths. They also concluded that variable-length coding in conjunction with feedback theoretically results in expected throughput very close to capacity at several hundreds of bits or less. Without using feedback one needs to use a long, capacity-achieving coding technique such as LDPC codes over several thousands of bits to achieve similar performance.

Chen et al. [2] and Williamson et al. [5] analyzed a VLFT scheme based on rate-compatible sphere-packing with an ML decoder (RCSP-ML). RCSP is an approximation of the performance of repeated incremental redundancy with noiseless transmitter confirmation (IR-NTC). The idea of RCSP is to extend sphere-packing analysis from a single fixed-length code to a family of rate-compatible codes. For the ideal family of rate-compatible codes, each code in the family achieves the perfect packing and is decoded by a maximum-likelihood (ML) decoder.

Chen et al. [6] also simulated a VLFT scheme using tail-biting convolutional codes. The simulation results are for very short, punctured rate-compatible tail-biting convolutional codes. For the 2-dB binary-input AWGN channel (BI-AWGNC), rate-compatible tail-biting convolutional codes with feedback achieve about 95% of the idealized RCSP-ML throughput (R_{RCSP}) for average blocklengths up to 50 bits. However, for average blocklengths of 100 bits and larger, the throughput of the convolutional code decreases. This performance degradation worsens as the average blocklength increases because the performance of the convolutional code does not improve as the length of code increases.

As Chen et al. mention in [6], rate-compatible codes for IR-NTC systems that approach the performance predicted by RCSP-ML in the VLFT setting still remain to be identified for expected latencies (average blocklengths) of 200 to 600 bits. The primary purpose of this paper is to demonstrate that non-binary LDPC (NB-LDPC) codes with incremental transmissions that depend on the decoder state information fed back to the transmitter can attain $92-94\%$ of the predicted RCSP-ML throughput in the VLFT setting for average blocklengths of 150 to 450 bits. This latency region is important because it is still short enough that feedback provides a real advantage but also long enough that good VLFT performance can translate to good VLF performance when ideal termination is replaced by a practical termination scheme within the primary channel.

In this paper similar to [6], an incremental redundancy (IR) scheme is used to give information to the receiver one bit at a time. A genie in form of NTC informs the receiver whether the

This material is based upon work supported by the National Science Foundation under Grant Numbers 1162501 and 1161822. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA JPL Task Plan 82-17473.

decoded codeword is correct determining if the next increment needs to be sent. Since the transmissions continue until the correct codeword is decoded, the probability of error is zero.

The remainder of the paper proceeds as follows: Sec. II provides an overview of the system and presents the NB-LDPC code design. Sec. III presents the technique for determining each bit of incremental transmission based on the decoder state information provided through feedback to the transmitter. Sec. IV describes the post-processing modifications made to the NB-LDPC decoder to further improve performance. Sec.V compares the proposed system to RCSP-ML in the VLFT setting. Sec. VI concludes the paper.

II. VLFT with Non-Binary LDPC Codes

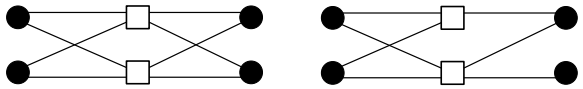
Feedback cannot increase the capacity of point-to-point channels, but it can facilitate higher throughput at significantly lower latency than systems without feedback. The latency improvement is made possible by capitalizing on favorable noise realizations and attempting to decode early, instead of needlessly sending additional symbols [5]. In case of an unfavorable noise realization, additional information is required which effectively lowers the communication rate to match the operational capacity of the channel. Therefore, incremental transmissions are necessary for coding systems with feedback.

A. System Overview

Traditionally, rate-compatible codes are designed by starting from a low-rate mother code and increasing the rate by puncturing the code. The proposed NB-LDPC coding scheme does not explicitly involve puncturing. Rather, we design a short, high-rate NB-LDPC code for which all symbols are transmitted in the initial transmission. Each subsequent transmission is a single bit carefully selected to help the decoder as much as possible given its current decoding state. The rate is gradually lowered by sending these additional bits, each of which is a function of selected bits in the binary representation of the non-binary symbols.

We choose high-rate protograph-based NB-LDPC codes in this paper. See [7] for a discussion of protograph-based LDPC design. The codes are irregular, having mostly degree-2 and a few degree-1 variable nodes. Short-blocklength NB-LDPC codes with only degree-2 variable nodes (regular) designed over large Galois field sizes are known to perform well [8]. However, we observed that for short blocklengths and large Galois field sizes, the addition of a few degree-1 variable nodes improves the convergence of the decoder in the low SNR region (see Fig. 2 and Table I). The operating SNR in this paper is 2dB, similar to the work of [6].

It is crucial that the code used for the initial transmission has a very high coding rate, even higher than the capacity. The coding rate is lowered in case of decoding failure. For example for SNR-2dB BI-AWGNC, the initial code can have a rate of 0.75 to 0.8 while the capacity of the channel is 0.685. By doing this we can take advantage of good noise realization and decode correctly at a very low latency.



(a) Only degree-2 variable nodes (regular)

(b) degree-1 and 2 variable nodes (irregular)

Fig. 1: Two protographs with only degree-2 variable nodes and a combination of degree-2 and 1 variable nodes

B. Desirability of degree-1 variable nodes

Simulation results of short-blocklength NB-LDPC codes over large Galois fields show an improved convergence at low SNR values for codes *with* degree-1 variable nodes. Fig. 2 shows the frame error rate (FER) comparison of the lifted (copied and permuted) versions of the two rate-1/2 protographs in Fig. 1 for low E_b/N_0 values. These protographs are lifted 4 times and the NB-LDPC codes are over $GF(2^8)$.

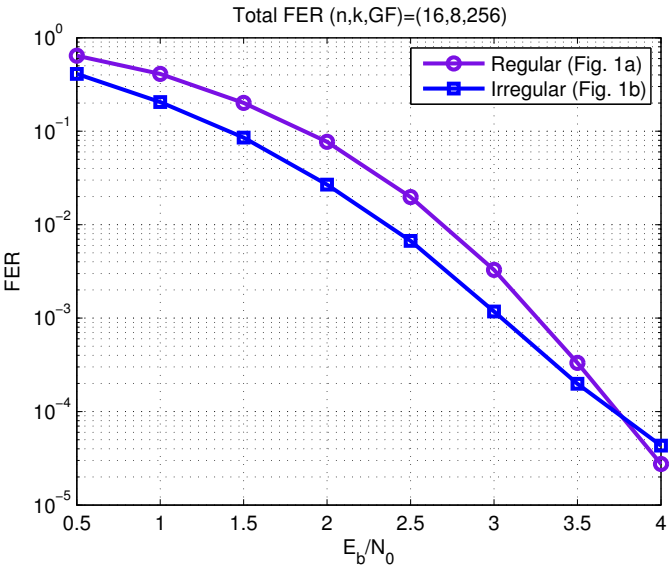


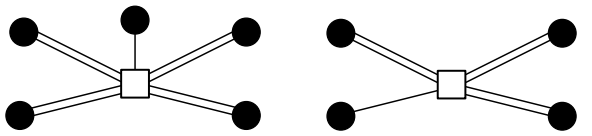
Fig. 2: FER comparison of the lifted versions of the regular (Fig. 1a) and irregular (Fig. 1b) protographs in Fig. 1

The FER for the protograph in Fig. 1b is better than Fig. 1a. The FERs in Fig. 2 may seem high, but for feedback with incremental redundancy, an initial FER of 0.02 is not unreasonable.

The better FER for the protograph in Fig. 1b than the one in Fig. 1a is justified by the smaller number of short cycles present in the lifted version of the protograph in Fig. 1b. Table I lists the number of $l-cycles$ (cycles of length l) for the two protographs in Fig. 1.

TABLE I: Small cycle count for the protograhs in Fig. 1

Protograph	4-cycles	6-cycles	8-cycles	10-cycles	12-cycles
Fig. 1.a	0	0	36	0	96
Fig. 1.b	0	0	6	0	16



(a) Rate-0.8 irregular protograph with degree-1 and 2 variable nodes

(b) Rate-0.75 irregular protograph with degree-1 and 2 variable nodes

Fig. 3: Rate-0.75 and rate-0.8 non-binary LDPC protographs

There is an issue with having degree-1 variable nodes in NB-LDPC codes. The connection of these variable nodes to cycles results in rank-deficient sub-matrices within the non-binary parity-check matrix [9] leading to a low minimum symbol distance. However, this issue is less problematic in feedback systems where the incremental transmissions gradually increase the distance among candidate codewords.

C. Non-binary Code Designs

In this paper we design NB-LDPC codes with symbols from $GF(2^8)$ for three different information blocks (k) of 96, 192, and 288 bits. The code for information block $k = 96$ ($K = 12$ $GF(2^8)$ symbols) is the 3-times lifted version of the rate-0.8 protograph in Fig. 3a. For information blocks $k = 192$ and $k = 288$ bits ($K = 24$, and $K = 36$ $GF(2^8)$ symbols), the protograph in Fig. 3b is lifted 8 and 12 times respectively to obtain rate-0.75 codes. The reason for selecting a higher-rate code for $k = 96$ is given in Sec. V and a complete description of the non-binary codes used in this paper is available online[1].

III. Incremental Transmissions

Incremental transmissions are used progressively to lower the rate of the codes designed in the previous section until the transmission is decoded correctly. This section presents how the incremental transmissions are created bit by bit in the encoder and processed by the decoder.

A. Creating a bit for incremental transmission

In order to use a NB-LDPC code over binary-input channels, each $GF(2^m)$ symbol is converted to m bits as follows: There is a primitive element α associated with each Galois field. Each non-zero element can be represented as a power of α so that $\{GF(2^m)\} = \{0, \alpha^0, \alpha^1, ..., \alpha^{(2^m-2)}\}$.

There is also at least one primitive polynomial $\rho(x)$ associated with $GF(2^m)$, satisfying $\rho(\alpha) = 0$. The primitive polynomial associated with $GF(2^m)$ and the property $\alpha^i + \alpha^i = 0$ allow the larger powers of α to be derived as polynomials of α with degrees of at most $m-1$. These limited-degree polynomials yield an m-bit binary representation. For example, consider $GF(2^3)$ with primitive element of α. A primitive polynomial for $GF(2^3)$ is $\rho(x) = x^3 + x + 1$ so that $\alpha^3 + \alpha + 1 = 0$, which implies that $\alpha^3 = \alpha + 1$. Table II shows how each element of $GF(2^3)$ can be uniquely represented in 3 bits (g_3, g_2, g_1).

[1]http://www.seas.ucla.edu/csl/resources/index.htm

TABLE II: Binary representation of $GF(8)$ elements

α^i	0	1	α	α^2	α^3	α^4	α^5	α^6
Poly.	0	1	α	α^2	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	α^2+1
$g_3g_2g_1$	000	001	010	100	011	110	111	101

The rate-$\frac{K}{N}$ non-binary LDPC codes proposed here initially encode a sequence of Km bits (K $GF(2^m)$ symbols) into a codeword of length Nm. The rate is lowered to $\frac{Km}{Nm+b}$ where b is number of additional incremental bits. Each additional bit is created by an xor $(\oplus)$ combination of bits in the binary representation of the $GF(2^m)$ symbols. For each variable node, the reliability of each of the 2^m-1 possible combinations of the bits in the binary representation is computed. For example, in $GF(2^3)$ the reliabilities of the seven possible combinations $g_1, g_2, g_3, g_1 \oplus g_2, g_2 \oplus g_3, g_1 \oplus g_3, g_1 \oplus g_2 \oplus g_3$ are computed for each variable node. The combination with the lowest reliability (looking over all combinations for all variable nodes) is transmitted.

This is a form of active feedback in the sense that the feedback is telling the transmitter *what* to transmitter rather than telling the transmitter only *whether* additional bits from a pre-determined rate-compatible code family. This is a generalization of the ideas of active hypothesis testing [10]. For comparison, we also performed simulations using non-active feedback, in which the additional bits are selected at random.

B. Using incremental transmissions in the decoder

The input frame consisting of K $GF(2^m)$ information symbols is initially encoded by the rate-$\frac{K}{N}$ NB-LDPC encoder into a sequence of length N $GF(2^m)$. These $GF(2^m)$ symbols are converted by their binary representations to bits. The Nm bits are modulated using BPSK and transmitted over an AWGN channel. The additive noise is modeled as an independent, zero-mean Gaussian random sequence with variance σ^2. The received Nm bits are grouped into N blocks of length m. These binary blocks of size m are used to compute the reliabilities (log-likelihood ratios) for each non-binary symbol. As in [6], SNR is calculated as $\frac{1}{\sigma^2}$, the ratio of the transmission power over the noise variance.

The iterative decoder sends vectors of log-likelihood ratios (LLRs) between variable and check nodes. In the initial iteration the LLR associated with the belief that $V_j = a$, the variable node j taking on the value $a \in GF(2^m)$, with respect to the reference belief that $V_j = 0$ is defined by $LLR_j^a = \log\left(\frac{P(V_j=a|Y)}{P(V_j=0|Y)}\right)$, where Y is the information received from the channel. Since the noise is independent Gaussian, the prior LLR values obtained from the channel can be calculated from the sum of the individual LLRs for each received bit in the group of m bits. For example, for a variable node j over $GF(2^3)$, the channel prior LLR for $\alpha^4 = (1, 1, 0)$

is calculated as follows:

$$LLR_j^{\alpha^4} = \log\left(\frac{P(V_j=a|Y)}{P(V_j=0|Y)}\right) \quad (1)$$
$$= \log\frac{P(g_3(V_j)=1|x_3)P(g_2(V_j)=1|x_2)P(g_1(V_j)=0|x_1)}{P(g_3(V_j)=0|x_3)P(g_2(V_j)=0|x_2)P(g_1(V_j)=0|x_1)}$$
$$= \log\frac{P(g_3(V_j)=1|x_3)P(g_2(V_j)=1|x_2)}{P(g_3(V_j)=0|x_3)P(g_2(V_j)=0|x_2)}$$
$$= -\frac{2}{\sigma^2}(x_3+x_2).$$

The decoding algorithm is initialized by sending these messages from variable nodes to check nodes. After this initialization step, the decoding process continues by sending LLR vectors iteratively from the variable nodes to their neighboring check nodes and vice versa. Iterative message passing only exchanges extrinsic information between connected nodes.

A hard decision about the original codeword is made when the messages go back to the variable nodes. If this decision satisfies the check-node constraints, the decoding is terminated. Otherwise, the messages between variable and check nodes are iteratively exchanged until a valid codeword is detected or the maximum number of iterations is reached.

Once the decoder terminates iterations, it provides feedback to the transmitter of the codeword it has identified (or that it has failed to identify a codeword) and the least reliable combination of bits identified as described in Section III-A. If the codeword is correct, the transmitter sends the NTC which terminates the transmission of that codeword. Otherwise, the transmitter combines the requested bits using the xor operation and transmits the new bit. The decoder re-computes the initialization only for the variable node that has been updated with an additional bit. Continuing the previous example, if bits 1 and 3 in variable node j are combined for the transmitted incremental bit, the new initialization LLR is computed as:

$$LLR'^{\,\alpha^4}_j = LLR_j^{\alpha^4} + \log\frac{P(g_4(V_j)=(0\oplus 1)|x_{\text{new}})}{P(g_4(V_j)=0|x_{\text{new}})} \quad (2)$$
$$= LLR_j^{\alpha^4} - \frac{2}{\sigma^2}(x_{\text{new}}). \quad (3)$$

The rest of the iterative process is not changed. The transmission of addition bits based on lowest reliability combination continues until the decoder decodes to the correct codeword.

This coding scheme makes the unreliable variable nodes more reliable and let the decoder escape from the local minima (trapping and absorbing sets) in case of non-convergence. Additionally, this form of concatenation make the valid but wrong codewords less probable than the correct codeword.

IV. Post Processing

The performance of a belief-propagation (BP) decoder is suboptimal compared to the optimal ML decoding. This suboptimality is usually due to the failure of the BP decoder to converge to a valid codeword. Post-processing techniques improve the performance of BP decoders by helping the decoder converge to valid codewords more frequently.

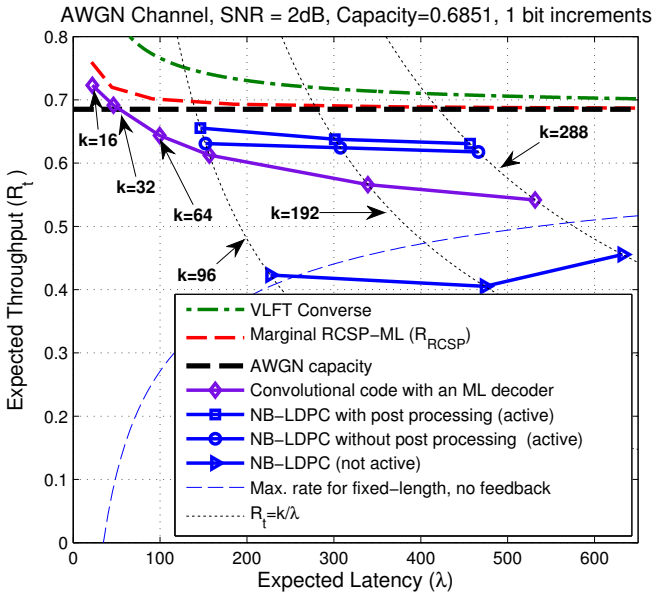


Fig. 4: Expected throughput R_t vs. expected latency λ for VLFT with non-binary LDPC codes with feedback.

The post-processing technique used here is a modified version of Random Initial State (RIS) algorithm which originally [11] resulted in improvements in the floor and waterfall regions for a binary LDPC decoder. The RIS algorithm explores the neighborhood of the received vector to find a convergence region by adding a dithering random vector to the received vector. The dithering slightly perturbs the initial vector so that the decoder decodes to a nearby codeword.

In case of non convergence to a valid codeword, dithering begins by choosing a vector with independent and identically Gaussian distributed zero-mean components with variance of σ'^2 smaller than the variance of the channel noise. This process is performed for a certain number of times t.

If the decoder fails to converge to a valid codeword after t decoding attempts, the incremental bit is transmitted. If only a unique valid codeword is detected, this codeword is fed back to the transmitter. If the decoded codeword is correct the next block of information is sent, otherwise, an additional incremental bit is transmitted to decode the current block. RIS algorithm may result in the decoder converging to multiple valid codewords in the t decoding attempts. In this case, the receiver computes the squared Euclidean distance between the detected codewords and the received sequence and selects the codeword with the smallest squared Euclidean distance.

V. Results

Fig. 4 presents simulation results of VLFT communication on the BI-AWGNC with BPSK modulation and soft-decision decoding using the NB-LDPC codes described in Sec. II-C with both active and non-active feedback as presented in Sec. III. For comparison we present simulation results for the tail-biting convolutional-code VLFT system of [6], R_{RCSP} (the throughput of RCSP-ML from [6]), the VLFT converse from

TABLE III: Throughput percentage of RCSP-ML throughput achieved by convolutional and non-binary LDPC codes

	Convolutional Code						NB-LDPC Code		
k	16	32	64	96	192	288	96	192	288
λ	22.1	46.3	99.4	156.7	339.2	531.5	146.4	301	456.6
R_t	0.72	0.69	0.64	0.61	0.57	0.54	0.65	0.64	0.63
R_{RCSP}	0.76	0.72	0.7	0.697	0.689	0.688	0.697	0.689	0.688
%	95.2	96.1	92	88	82.1	78.7	94.1	92.6	92.2

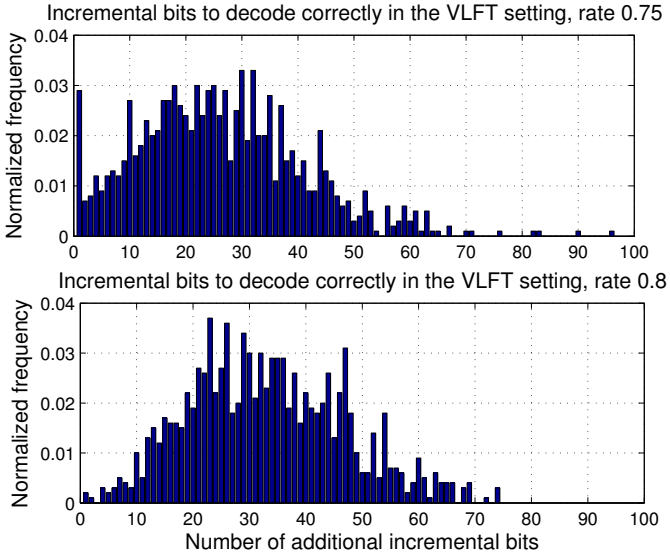


Fig. 5: Empirical probability of additional incremental bits required to decode correctly in the VLFT setting for k=96 bits and initial blocklengths of 120 and 128 bits corresponding to rate-0.8 and rate-0.75 non-binary LDPC codes respectively

[1], and throughput for transmission without feedback [4] with a decoding error probability of 10^{-6}. The simulation points follow $R_t = \frac{k}{\lambda}$. For the VLFT systems, the probability of error is zero because the transmitter informs the receiver when it has decoded successfully using NTC.

The NTC provides a benefit that increases the original channel capacity. As shown by the VLFT converse, R_{RCSP}, and even the convolutional VLFT simulation, NTC allows rates higher than the original channel capacity when the average blocklength is very small. As discussed in [6], [12], the benefit of NTC (and the overhead associated with replacing NTC with regular channel uses that reliably inform the receiver of successful decoding) becomes smaller for larger blocklengths. This is the primary motivation to consider blocklengths that are several hundred bits, which is still short enough that feedback provides a significant throughput advantage over the no-feedback curve but long enough that a system without NTC can be implemented without too much overhead for the transmitter confirmation. Thus, even though convolutional VLFT has higher throughputs (even throughputs above the original capacity), NB-LDPC VLFT may have more practical potential in the VLF setting.

Table III shows the throughput percentage of R_{RCSP}, which we view as a practical upper bound, obtained by NB-LDPC VLFT with active feedback. The NB-LDPC codes with active feedback attain throughputs higher than 92% of R_{RCSP}. Active feedback was essential. When non-active feedback was used the performance was significantly worse. Post processing provided a relatively small improvement, which was more pronounced for shorter average blocklength.

Fig. 5 shows the normalized frequency of the number of required incremental bits to decode correctly with active feedback (but before post processing) for the $k = 96$ VLFT NB-LDPC code. The reason to select a higher rate (rate-0.8 instead of rate-0.75) protograph is that very a few codewords are decoded with no incremental transmissions. When codewords are decoded with no incremental transmissions, the transmission of unnecessary symbols reduces the throughput.

VI. Conclusions

In this paper, VLFT using non-binary LDPC codes and active feedback identifying the most helpful incremental transmissions achieves $92-94\%$ of R_{RCSP} for average blocklengths in the range of 150-450 bits. This range of blocklengths is interesting because it is still small enough that feedback provides a throughput advantage but also large enough to have practical potential in the VLF setting. The application of non-binary LDPC codes and active feedback to the VLF setting is an area of ongoing research.

References

[1] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Feedback in the non-asymptotic regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.

[2] T.-Y. Chen, N. Seshadri, and R. D. Wesel, "A sphere-packing analysis of incremental redundancy with feedback," in *Proc. 2011 IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, June 2011.

[3] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.

[4] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[5] A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "A rate-compatible sphere-packing analysis of feedback coding with limited retransmissions," in *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, USA, July 2012.

[6] T.-Y. Chen, A. R. Williamson, N. Seshadri, and R. D. Wesel, "Rate-compatible sphere-packing analysis," submitted for publication.

[7] B.-Y. Chang, D. Divsalar, and L. Dolecek, "Non-binary protograph-based LDPC codes for short block-lengths," in *2012 IEEE Inform. Theory Workshop (ITW)*, Sep. 2012, pp. 282–286.

[8] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2, d_c)$-LDPC codes over GF(q) using their binary images," vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[9] S. Ranganathan, K. Vakilinia, D. Divsalar, and R. Wesel, "Design of high-rate irregular non-binary ldpc codes using algorithmic stopping-set cancellation," in *Submitted to 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, July 2014.

[10] M. Naghshvar, T. Javidi, and M. A. Wigger, "Extrinsic jensen-shannon divergence: Applications to variable-length coding," *CoRR*, vol. abs/1307.0067, 2013.

[11] F. Leduc-Primeau, S. Hemati, S. Mannor, and W. Gross, "Dithered belief propagation decoding," *Communications, IEEE Transactions on*, vol. 60, no. 8, pp. 2042–2047, 2012.

[12] A. Williamson, T.-Y. Chen, and R. Wesel, "Firing the genie: Two-phase short-blocklength convolutional coding with feedback," in *Information Theory and Applications Workshop (ITA), 2013*, 2013, pp. 1–6.