Optimizing Incremental Redundancy for Millimeter Wave Wireless Communication Using Low Density Parity Check Codes

Jiho Song†, Borja Peleato†, David J. Love†, Tianqiong Luo†, Dennis Ogbe† and Amitava Ghosh∗ †School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA,
Email: {jihosong, bpeleato, djlove, luo133, dogbe}@purdue.edu ∗Nokia Networks, Arlington Heights, IL 60004 USA
Email: amitava.ghosh@nokia.com

Abstract

The high speeds and low latency expected in millimeter wave (mmWave) communications will require strong error correcting codes and efficient decoders. Channel conditions can change widely due to factors such as beam alignment, blockage, and interference, making it necessary to adjust the modulation size, coding rate and other transmission parameters frequently. It is expected that mmWave networks will use incremental redundancy schemes, where additional parity bits are transmitted whenever a codeword fails to decode successfully.

This paper studies some of the trade-offs that arise in mmWave scenarios with binary LDPC error correction codes (ECC) and incremental redundancy. First, it considers the pros and cons resulting from different mappings of bits and codewords to the available time, frequency, and modulation resources. Then it proposes a dynamic programming scheme for adaptively adjusting the number of incremental bits based on the quality of the previously received symbols. The proposed scheme can significantly increase the average throughput in scenarios where the channel quality is uncertain. Finally, the paper addresses the problem of acknowledgement bundling and analyzes the tradeoff between using individual incremental redundancy for each codeword and erasure codes over all codewords in a bundle.
I. INTRODUCTION

Wireless communication systems working at millimeter wave (mmWave) frequencies are being spotlighted in fifth generation (5G) wireless networks [1]–[5]. The wide bandwidths at mmWave frequencies enable a high transmission data rate by exploiting multiuser multiple-input multiple-output (MIMO) techniques, but channel conditions are unfavorable for reliable wireless communications because of the higher expected path loss and channel attenuation. Directional transmission is thus necessary to compensate the unfavorable channel conditions by facilitating large beamforming gain. However, the large number of antennas needed to create a highly directional beam complicates the estimation of channel state information (CSI) and computation of transmit beamformers [6]–[9].

Recently, some promising techniques have been proposed for selecting a transmit beamformer without estimating CSI [10]–[12], but the low signal-to-noise ratio (SNR) conditions of mmWave links interrupt correct beam alignment. The misaligned beamformers combined with directional characteristics might disconnect mmWave links intermittently and make beamforming gain fluctuate. Thus, securing a seamless radio connection is therefore a major challenge in mmWave systems. Typical wireless systems compensate for link gain fluctuations by periodically adjusting the size of the modulation and the rate of the error correcting codes (ECC), but such an approach can only address slow fading, not the fast fading components. A conservative configuration of these parameters leaving enough noise margin to withstand gain fluctuations would solve this issue, but would jeopardize the high throughput and low latencies expected from mmWave communications.

Incremental redundancy (IR) schemes offer a significant increase in reliability with negligible impact in throughput and latency. The basic idea is to use an aggressive configuration of the system (high rate and modulation order) for the first transmission of each information block, and then provide additional parity bits (effectively reducing the rate) for whichever blocks were not correctly received.

Both Low Density Parity Check (LDPC) and convolutional turbo codes have been considered for incremental redundancy in mmWave communications [13]–[15]. Convolutional codes can accommodate IR with near optimal performance, but their decoding process is significantly more computationally intensive than that of LDPC codes. LDPC codes can be decoded very efficiently using parallel shift registers, but they require longer codewords, i.e., higher latency,
and their performance degrades when they are punctured or extended to modify the rate. After a lengthy discussion, it was decided to include binary LDPC codes in the standard [16]. Hence, this paper will focus on LDPC codes.

In this paper we provide a general framework for addressing the incremental redundancy needs of millimeter wave systems. We explore multiple practical aspects of coding that may have significant impact on the research, design, and standardization of millimeter wave for 5G and beyond systems.

We first address the bit-to-symbol mapping and its effects. We also study how codewords should be mapped/interleaved across the time-frequency and constellation resources. This mapping can significantly impact performance, and we show that if the channels are not sufficiently diverse interleaving can actually decrease performance. This motivates either changing the outlook on interleaving or moving to a wideband codeword mapping.

Though feedback does not increase capacity of a system operating over a memoryless channel, it can significantly impact the error exponent [17] and was a focus of past research (e.g., [18], [19]). This has led to a revival of research into feedback-based schemes [20]–[23]. We show how feedback improvements can be easily integrated into millimeter wave systems by using adaptive incremental redundancy. The basic idea is to leverage multiple bits of acknowledgement/negative acknowledgement (ACK/NACK) feedback to a codeword. The system can then adapt the incremental redundancy as a function of this feedback.

Adding additional feedback, however, comes at a cost. This control overhead can eat up valuable resources, particularly when used on the uplink. To address this problem, we show that the ACK/NACK feedback rate can be adapted as a function of the channel conditions. This generalizes the notion of ACK/NACK bundling which has already been included in LTE-Advanced [24]–[26].

The paper will be organized as follows. Section II will introduce the notation and system model used throughout the rest of the paper. Additionally it will describe how the information is encoded and decoded using LDPC codes and how these codes are extended to generate incremental redundancy. Section III describes multiple possible mappings of bits and codewords to the resources available in a multi-carrier system and explores the advantages and drawbacks for each of them. Sections IV and V propose methods for optimizing the length and structure of the incremental redundancy so as to maximize the information throughput. Specifically, Section IV introduces a dynamic programming scheme for adaptively adjusting the number of incremental
bits requested by the receiver, and Section V provides guidelines for bundling feedback bits in the ARQ protocol. Finally, Section VI presents simulation results to evaluate the performance of the proposed methods and Section VII concludes the paper.

II. SYSTEM MODEL

Throughout this paper, \( \mathbb{C} \) denotes the field of complex numbers, \( \mathbb{B} = \{0, 1\} \), \( \mathcal{CN}(\mu, \sigma^2) \) denotes the complex normal distribution with mean \( \mu \) and variance \( \sigma^2 \), \([a, b]\) is the closed interval between \( a \) and \( b \), \( U(a, b) \) denotes the uniform distribution in \([a, b]\), \( E[\cdot] \) is the expectation operator, \( \text{mod}(a, n) \) is the \( a \) modulus \( n \), and \( \otimes \) is the Kronecker product. Also, \( a^H \) denotes the conjugate transpose of the column vector \( a \).

A. System Model

We consider multiple-input single-output (MISO) systems that exploit \( M \) antennas at the transmitter and a single antenna at the receiver. Assuming a narrowband block fading channel framework, the input-output expression for the \( k \)-th subcarrier at the \( \ell \)-th wideband channel use (e.g., an OFDM symbol channel use) is defined as\(^1\)

\[
y[k, \ell] \overset{(a)}{=} \sqrt{\rho} h^H[k, \ell] f[k, \ell] x[k, \ell] + n[k, \ell],
\]

where \( y[k, \ell] \in \mathbb{C} \) is the received baseband signal, \( \rho \) is the transmit SNR, \( h[k, \ell] \in \mathbb{C}^M \) is the fading channel, \( f[k, \ell] \in \mathbb{C}^M \) is the unit norm transmit beamformer, \( x[k, \ell] \in \mathbb{C} \) is the transmit symbol constrained such that \( E[|x[k, \ell]|^2] \leq 1 \), and \( n[k, \ell] \sim \mathcal{CN}(0, 1) \) is additive white Gaussian noise (AWGN). We consider a block-fading structure with block size \( L_{\text{chan}} \), which means that

\(^1\)Note that (a) assumes that the received signals are perfectly synchronized.
\( h[k, 1] = \cdots = h[k, L_{\text{chan}}] \). For convenience, we use the notation \( h_t[k] \) to refer to the \( t \)th channel block, meaning \( h[k, 1] = \cdots = h[k, L_{\text{chan}}] = h_0[k] \).

Due to the lack of scatterers at mmWave frequencies, the MISO channel can be modeled as a combination of a small number of beams [27]–[29]. Therefore, we can write the channel as a sum of \( R + 1 \) paths weighted paths

\[
    h_t[k] = \sum_{r=0}^{R} \alpha_{t,r}[k] d_M(\psi_{t,r}[k]),
\]

where \( \alpha_{t,r}[k] \) is the complex weight of the \( r \)th path for the \( t \)th block and \( d_M(\psi_{t,r}[k]) \) is the \( M \) antenna array manifold vector corresponding to the \( r \)th path’s angle of departure.

Because of the channel’s block-fading structure, the beamformer will also possess a block-based structure with \( f[k, 1] = \cdots = f[k, L_{\text{chan}}] = f_0[k] \). For the \( t \)th block, the transmit beamformer is chosen according to

\[
    f_t[k] = \arg\min_{\bar{f} \in \mathcal{F}} |h_t^H[k]\bar{f}|^2,
\]

where \( \mathcal{F} \) is the set in which the transmit beamformer is restricted. Common sets in millimeter wave include the set consisting of the \( M \) columns of the \( M \times M \) discrete Fourier transform (DFT) matrix (denoted here as \( \mathcal{F}_{\text{DFT}} \)) [30] and Kronecker product structured sets [31].

To explain the coding structure, we first concentrate on the first codeword block. A codeword block is assumed to be mapped to \( K \) subcarriers \( \{k_1, \ldots, k_K\} \) and \( L_{\text{chan}} \) wideband channel uses. This means that the total codeword block length is \( L = K L_{\text{chan}} \) channel uses. For expositional purposes, we assume that \( h[k_1, \ell] = \cdots = h[k_K, \ell] \), meaning that each codeword goes over a single millimeter wave channel realization. Because of this structure, the analysis and explanation of the millimeter wave coding scheme can be explained using a narrowband discussion. Therefore, we drop the subcarrier index \( k \) for simplicity.

The set of received signals for the first block are rewritten as

\[
    y = [y[1], \ldots, y[L]]
    = \sqrt{\gamma_0} x + n \in \mathbb{C}^L,
\]

where \( \gamma_0 = \rho|\mathbf{f}_0|^2 \) denotes the total link gain parameter, \( \mathbf{x} = [x[1], \ldots, x[L]] \), \( \mathbf{n} = [n[1], \ldots, n[L]] \), and \( L \) denotes the number of channel uses to be exploited for each codeword. Note that each channel block includes static channels with identically distributed link gain, as depicted in Fig. 1.
We now discuss signal processing techniques on the transmitter side. Let $s \in \mathcal{B}^S$ denote the message to be transmitted, containing $S$ information bits. By exploiting a LDPC encoder\(^2\) with code rate $R' = S/U$, the message is encoded into a codeword consisting of $U$ bits,

$$u = \text{mod}(G^T s, 2)$$

$$= [s^T \ p^T]^T \in \mathcal{B}^U,$$

where $p \in \mathcal{B}^{U-S}$ are the parity check bits and $G \doteq [I_S (C^{-1}D)^T] \in \mathcal{B}^{S \times U}$ is the generator matrix corresponding to the parity check matrix

$$H \doteq [D \ C] \in \mathcal{B}^{(U-S) \times U}. \tag{3}$$

The encoded codeword $u$ is then modulated as

$$x \doteq [x[1], \ldots, x[L]] \in \mathbb{C}^L, \tag{4}$$

where $L = U/N$ for each codeword for a predefined modulation rate of $N$ bits/symbol. The transmitter performs binary phase shift keying (BPSK) modulation with $N = 1$, 4-QAM (quadrature amplitude modulation) with $N = 2$, and 16-QAM with $N = 4$.

**B. Incremental redundancy**

It is clear that the frame error rate (FER) improves with the number of parity check bits, i.e., $U - S$, but an excessive amount of redundancy degrades the data rate. In order to strike the optimal trade-off between these two trends, incremental redundancy schemes typically send each codeword with a very high rate at first, often higher than the channel capacity, and then progressively reduce the code rate by sending additional parity bits if the decoding fails. This process is depicted in Fig. 1.

LDPC codes are typically designed to provide performance very close to capacity at a specific rate. However, in order to accommodate incremental redundancy, these codes need to be punctured (to increase the rate) or extended (to reduce it). The puncturing and extension patterns need to be carefully designed to maintain the good performance without significantly increasing the decoder complexity [16].

In this work, we will limit the number of retransmission rounds to a maximum of $T$ and the number of additional parity check bits transmitted in the $t$th round will be denoted as $\Delta_t$. If the

\(^2\)We utilize a IEEE 802.11n parity check matrix in [32], giving $U = 648$ block length codewords.
receiver fails to decode the received signal $y$, the additional parity bits $p_{\Delta_1} \in B^{\Delta_1}$ should be designed to satisfy

$$\begin{bmatrix} H & 0_{\Delta_1 \times \Delta_1} \\ E_1 & I_{\Delta_1} \end{bmatrix} \begin{bmatrix} u \\ p_{\Delta_1} \end{bmatrix} = 0_{U-S+\Delta_1 \times 1},$$

where $E_1 \in B^{\Delta_1 \times (U-S)}$ is the expanded parity matrix\(^3\) for the 1-st retransmission round. Incremental redundancy blocks will be transmitted in a different frame than the initial codeword so that the link gains for additional parity bits may be different from that for the initial codeword, as shown in Fig. 1.

The signal sent during the $t$-th retransmission round is assumed to experience channel $h_t$. The receive SNR for the $t$-th retransmission round is $\gamma_t = \rho |h_t^H f_t|^2$. Therefore, incremental redundancy provides time diversity since the original codeword and the $T$ rounds of parity bits experience i.i.d. channels.

### III. CODEWORD MAPPING AND MODULATION

At the time of writing, there does not yet exist a commonly agreed standard for the modulation and frame structure of mmWave communications [34]. Additionally, the number of codewords transmitted in each frame will depend on the order of the modulation, code length, etc. This paper assumes that each frame extends over multiple subcarriers and channel uses, and that channel sounding and ARQ feedback are transmitted once per frame.

The high frequency radio waves used for mmWave communications do not travel well through rain, walls, or other obstacles, and they attenuate fast with distance. It is therefore necessary to use beamforming to compensate for path loss. The MISO channel can be modeled as a combination of a small number of beams [27]–[29]. Unfortunately, the small number of multipaths makes these systems vulnerable to blockage. In this paper we will consider that there is some probability of blockage or beam misalignment, which could cause a temporary drop in the effective link SNR.

In order to adapt to the changing conditions of the channel, our system will support multiple modulations and error control coding rates, which can be changed independently for each subcarrier. Specifically, information bits are encoded using a binary LDPC code chosen from a pre-defined list of codes with different rates. Additional flexibility in the rate adjustment can

\(^3\)Based on [33], we reuse the $D$ matrix in (3) to generate the expansion matrices.
be achieved by extending these codes as explained in subsection II-B. The coded bits are then mapped to modulation symbols to be transmitted using the time and frequency resource blocks.

In a multi-carrier modulation, there are two choices that need to be made to transform encoded bits into signals to be transmitted through the channel: how to map the bits into symbols and how to distribute the codewords across the subcarriers.

1) Mapping bits to modulation symbols: In BPSK each symbol carries a single bit, so the mapping is trivial. However, higher order modulations such as QPSK or M-QAM carry multiple bits per symbol and can use either of the two mappings shown in Fig. 2.

![Mapping of bits to 16-QAM constellation symbols.](image)

Gray mapping offers the lowest bit error rate (BER) since adjacent constellation symbols, between which errors typically happen in an AWGN channel, differ in a single bit. However, Gray mapping is generally harder to implement than superposition, specially in multi-user scenarios. The superposition mapping can be understood as an algebraic sum of independent BPSK modulations with different power. The transmitted symbols can therefore be constructed by scaling and adding multiple independent streams of binary data. For example, the 16-QAM symbols with superposition mapping in Fig. 2 can be expressed as

\[ s(b_1, b_2, b_3, b_4) = (2b_1 + b_2) + j(2b_3 + b_4), \]  

(5)

where \( b_1, \ldots, b_4 \in \{-1, 1\} \) are the modulated bits. Gray mapping, on the other hand, is non-linear so the modulator needs to have a block capable of interpreting the input bits and generating the corresponding symbol. Figure 4 illustrates the performance difference between both mappings.
Another factor to consider in the mapping of bits to modulation symbols is that when there are more than two bits per symbol not all the bits have the same probability of error. Regardless of whether Gray or superposition labeling is used, the second and fourth bit in Fig. 2 are approximately twice as likely to be in error as the first and third. Furthermore, the marginal distributions of the noise experienced by each bit are highly correlated. If all the bits in a symbol are part of the same codeword, the ECC decoder should take into account this correlation and the variations in reliability among the bits.

Ideally, the system would use a non-binary LDPC code with a decoder capable of handling pairwise transition probabilities between all the symbols in the constellation. Non-binary LDPC codes significantly outperform their binary counterparts in terms of failure rate, but their increased decoding complexity is a problem for the high speeds expected in mmWave communications. Therefore, most practical systems use binary LDPC codes and operate as if each bit suffered independent Gaussian noise. Subsection III-2 will propose a refinement to this approach.

Fig. 3. Mapping of codewords to resources. Codewords can be assigned different time slots, frequency subcarriers, or bits in a modulation symbol.

2) Mapping codewords to subcarriers: There are multiple ways in which the LDPC codewords can be mapped to the resource blocks, each with its own advantages and drawbacks. Figure 3 illustrates the main three that will be discussed.

- Split across time: LDPC codewords are transmitted sequentially in time, using all available subcarriers until the whole codeword has been transmitted. This mapping minimizes latency,
since codewords can start being decoded as soon as they are received. Additionally, full-duplex receivers would be able to report decoding failures immediately, which might compel the transmitter to adjust the modulation size or code rate for subsequent codewords in the same frame.

On the negative side, if the subcarriers use different modulations or there is frequency selective fading, the reliability of the bits in a codeword could vary widely. Furthermore, it is very hard for the decoder to exploit the correlation between the noise suffered by the bits in each modulation symbol.

- Split across frequency: LDPC codewords are assigned to different subcarriers, each using as many time slots as necessary. Frequency selective fading can be addressed by adjusting the modulation and coding rate within each codeword, and it is easy to multiplex users in frequency by assigning them different subcarriers.

Unfortunately, latency and memory requirements would be worse than in the previous case. The receiver needs to wait until the first batch of codewords has been received before starting to decode them all in parallel. The noise correlation between bits in the same modulation symbol is still hard to exploit, but there exist multi-edge type decoders that can do it [35], [36]. Assuming that the modulation in each subcarrier remains constant throughout the codeword, the complexity of these schemes will not be as high as in the case where codewords are split across time.

- Split across modulation: If the modulation symbols store multiple bits, have each of these bits belong to a different codeword. This increases the number of subcarriers and/or timeslots used to transmit each codeword, thereby increasing latency, but this mapping simplifies the exploitation of the correlation and asymmetries between the different bits in a symbol. Specifically, when one codeword is successfully decoded, the known bits can be used to refine the information on the other bits in the same symbols, helping decode the remaining codewords.

Furthermore, bits for which the modulation suffers a higher probability of error can be encoded at a lower rate than those which suffer lower BER. It was shown in [37] that the maximal coding rate achievable for a finite blocklength decreases with the channel dispersion (variance of the information density). Grouping the bits into codewords according to their probability of error will create several parallel channels with smaller dispersion than the original channel, thereby increasing the overall finite length capacity.
Fig. 4. Average number of information bits recovered per 16-QAM symbol sent, with different groupings of bits into codewords.

Fig. 4 shows the information rate achieved with a 16-QAM modulation at different SNR when all the bits in each symbol are part of the same codeword (solid curves) and when bits are grouped into codewords of different rate according to their probability of error (dashed curves). For these simulations we used the LDPC codes of length 648 proposed in the 802.11n standard [32]. It can be observed that splitting the bits according to their probability of error increases the rate for both bit mappings in Fig. 2.

An interesting observation that arises from the results in Fig. 4 is that interleaving is detrimental to the data rate. This seems counter-intuitive, since most practical systems use it extensively to compensate for channel variations, but it can be proven by a very simple example: The information theoretical capacity of two parallel binary symmetric channels (BSC) with error probabilities $p_1$ and $p_2$ is

$$C_{\text{parallel}} = 2 - H(p_1) - H(p_2),$$

where $H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)$ represents the binary entropy function. This expression assumes that each channel is carrying codewords of a different rate. However, if both channels are interleaved into a single one with error probability $\frac{1}{2}(p_1 + p_2)$, the resulting capacity
is
\[ C_{\text{interleaved}} = 2 - 2H\left(\frac{1}{2}(p_1 + p_2)\right). \] (7)

Some simple calculations show that \( C_{\text{interleaved}} \leq C_{\text{parallel}} \), which proves our argument\(^4\). The reason why interleaving is used in practice is that each round of incremental redundancy carries a certain amount of overhead and causes delay. Therefore, it is more efficient to configure the system conservatively and make the channel as uniform as possible.

IV. Adaptive Incremental Redundancy (IR)

The link gain \( \gamma \) is an important factor in the performance of mmWave systems. Significant variations of \( \gamma \) can be expected due to channel fading and blocked or disconnected radio links. Most systems compensate for these fluctuations by adjusting the order of the modulation and the rate of the error correcting codes, but such adjustments can only address predictable or slow changes in \( \gamma \). Unexpected drops in channel gain or pathological error patterns can prevent the successful decoding of some codewords, in which case the receiver can request additional redundancy to help in the decoding. In this section we study how feedback can be used to optimize the length of such incremental redundancy so as to minimize a given cost function. Our derivations will assume that the objective is to maximize the data rate, but the same process could be applied to other quantities.

It is a well known fact in information theory that feedback, regardless of its quality, does not increase the capacity of a memoryless channel [38]. However, the only way to achieve reliable transmission at a rate approaching the channel capacity is to use asymptotically long codewords. The maximum achievable rates for reliable transmission in the finite blocklength regime are substantially lower than the information-theoretic channel capacity [37], but feedback and variable length coding schemes can provide dramatic improvements [39], [40].

Previous papers have bounded the maximum achievable throughput with perfect feedback and unlimited single bit increments [39], and also proposed methods to optimize the incremental redundancy block lengths when there is a limit on the maximum number of retransmissions [41], [42]. Both active feedback (additional bits are chosen as the combination with lowest reliability) and non-active feedback (predetermined combinations are sent) have been studied [43], [44].

\(^4\)Alternatively, observe that \( C_{\text{BSC}} = 1 - H(p) \) is a strictly convex function in \( p \in (0, 1) \) and apply Jensen's inequality.
However, most of the existing work has focused on idealized or non-binary LDPC codes with either single bit or unlimited feedback.

We address a more practical scenario for mmWave communications with binary LDPC codes, a small number of feedback bits, and limited incremental transmissions. Unlike the work in [41], which assumed that the block length of each retransmission had to be fixed, we assume that the feedback link gives the receiver some control over the number of bits sent as incremental redundancy. The series of retransmissions is modeled as a Markov Decision Process which can be solved by backtracking. We use a dynamic programming approach to design a policy that specifies the number of bits to be requested in each incremental transmission, based on the quality of the bits received until that point and that expected in subsequent transmissions. Even with a coarse quantization of the channel outputs, the receiver can estimate the channel quality from the BER in a known pilot and successfully decoded codewords, or from the number of unsatisfied checks in the LDPC code [45].

A. Computing the probability of failure

According to [37], [41] the maximum rate at which a codeword can be decoded successfully can be well approximated by a Gaussian distribution whose mean and variance depend on the receive SNR $\gamma$. The probability of a decoding attempt being unsuccessful at rate $r = \frac{k}{N}$ and receive SNR $\gamma$ is then given by

$$F\left(\frac{k}{N} \mid \gamma\right) = 1 - Q\left(\frac{\frac{k}{N} - \mu(\gamma)}{\sigma(\gamma)}\right) = Q\left(\frac{\mu(\gamma) - \frac{k}{N}}{\sigma(\gamma)}\right),$$

(8)

where $k$ represents the number of information bits per codeword and $N$ the total number of bits transmitted.

In order to reduce the computational requirements of the proposed scheme, we assume that the probability of decoding failure after having received $\Delta$ additional bits satisfies the Markov property, in the sense that it only depends on the probability of failure before receiving these bits and not on the previous rate and SNR separately. The state $s$ in the Markov Decision Process is then reduced to a scalar specifying the probability of decoding failure. If $N$ bits have been received with an estimated receive SNR of $\gamma$, the state would be

$$s = F\left(\frac{k}{N} \mid \gamma\right).$$
However, there exist multiple pairs of $N$ and $\gamma$ that would yield the same $s$. Let $\tilde{\gamma}$ be the expected SNR in subsequent transmissions and $N_0$ represent the codeword length required to achieve a probability $s$ of decoding failure when $SNR = \tilde{\gamma}$, then

$$P_{\text{fail}}(\Delta | s) = \left(\frac{1}{s}\right)F\left(\frac{k}{N_0 + \Delta \mid \tilde{\gamma}}\right)$$

(9)

represents the probability of decoding failure with $\Delta$ additional bits, given that it failed without them. $N_0$ can be found by inverting the cdf in (8).

Figure 5 shows some simulation results to support our assumption that $P_{\text{fail}}(\Delta | s)$ satisfies the Markov property. We found several pairs of $N$ and $\gamma$ that resulted in similar probability of decoding failure $s$ and evaluated how their success rate increased with additional parity bits. The figures show the results when these additional bits are received with an SNR of 0dB (left) and -5dB (right). The $(N, \gamma)$ pairs with similar $s$ experience very similar curves, hence we can conclude that the probability of success after incremental redundancy is mostly determined by the previous probability (state $s$) and the additional information received.

Fig. 5. Effect of incremental redundancy on different $(N, \rho)$ pairs. It can be observed that the curves for $(N, \rho)$ pairs resulting in similar $s$ remain close regardless of the length and quality of the incremental redundancy.
B. Optimizing the policies

Incremental redundancy schemes first transmit each codeword with a very high rate, often higher than the channel capacity, and then progressively reduce this rate by sending additional parity bits if the decoding fails. This approach increases the overall data rate but it can also introduce latency, since most codewords will require multiple rounds of retransmission before they can be successfully decoded. This paper assumes a pre-defined rate for the first transmission, and attempts to optimize the length of subsequent retransmissions.

Let \( T \) denote the maximum number of incremental transmission rounds allowed, after which the receiver gives up on that codeword. We need to find a set of policies \( \pi_i(s), i = 1, \ldots, T \), specifying the number of additional bits to be requested in each round as a function the current state \( s \).

Although other cost functions are also possible (e.g., we could include the overhead mentioned at the end of Section III), we choose to minimize

\[
\text{Cost} = \Delta_1 + \cdots + \Delta_T + C_{\text{drop}}I_{\{\text{fail}\}},
\]

where \( I_{\{\text{fail}\}} \) is an indicator function taking value 1 if the decoding still fails after all \( T \) retransmissions and 0 otherwise. The penalty \( C_{\text{drop}} \) controls the tradeoff between sending a large number of additional bits to ensure successful decoding of the codeword and giving up on the incremental redundancy scheme. If \( C_{\text{drop}} \) is large, the resulting policies will request many incremental bits to reduce the risk of failure. If \( C_{\text{drop}} \) is small, the cost will be dominated by the number of IR bits transmitted, hence the optimal policy might prefer to drop the codeword rather than request additional bits without a certainty of successful decoding. A specially interesting case arises when \( C_{\text{drop}} \) is chosen as the expected number of bits transmitted until successful decoding of a codeword. In that case our objective can be interpreted as maximizing the average data rate:

\[
r_{\text{IR}} = \frac{k}{N + E\{\text{Cost}\}}.
\]

The number of additional bits to be sent in each retransmission should be chosen to minimize the expected cost in Eq. (10), given the current state \( s \) and previous choices. Doing this for the last incremental transmission round yields

\[
\Delta_T^* = \arg \min_{\Delta_T} E\{\text{Cost}|s, \Delta_1, \ldots, \Delta_{T-1}\}
\]

\[
= \arg \min_{\Delta} \Delta + C_{\text{drop}}P_{\text{fail}}(\Delta|s),
\]
where $P_{\text{fail}}(\Delta | s)$ represents the probability of the decoding to fail after sending $\Delta$ additional bits given that the system is in state $s$. This state $s$ should include all the necessary information about previous transmissions. With $P_{\text{fail}}(\Delta | s)$ defined in (9), the optimal number of additional bits $\Delta^*_T$ can be found through a simple line search. This defines a policy for the $T$-th retransmission and an expected future cost for each possible state after failing to decode with the first $T-1$ retransmissions given by

$$\pi_T(s) = \Delta^*_T$$

$$V_{T-1}(s) = E\{\Delta_T + C_{\text{drop}}I_{\{\text{fail}\}}|s\}$$

$$= \pi_T(s) + C_{\text{drop}}P_{\text{fail}}(\pi_T(s)|s).$$

Similarly, we can define policies and expected costs for previous incremental transmission rounds

$$\pi_i(s) = \arg\min_{\Delta_i} E\{\text{Cost}|s, \Delta_1, \ldots, \Delta_{i-1}\}$$

$$= \arg\min_{\Delta} \Delta + P_{\text{fail}}(\Delta | s)E[V_i|s, \Delta]$$

$$V_{i-1}(s) = E\{\Delta_i + \cdots + \Delta_T + C_{\text{drop}}I_{\{\text{fail}\}}|s\}$$

$$= \pi_i(s) + P_{\text{fail}}(\pi_i(s)|s)E[V_i|s, \pi_i(s)],$$

where $i = 1, \ldots, T-1$ and the expectations are computed with respect to the possible states if the decoding fails after the $i$-th retransmission. Once again, these policies can be found sequentially with a simple line search. It is worth noting that

$$\pi_1(s) = \arg\min_{\Delta_1} E\{\text{Cost}|s\}$$

$$V_0(s) = E\{\text{Cost}|s\}.$$

The Markov assumption in Section IV-A simplifies the implementation of this scheme significantly. The policies can be computed offline, tabulated, and stored at the receiver. It is not necessary to perform these optimizations in real time, the receiver only needs to estimate the state and request the corresponding number of incremental bits. If we are devoting $k$ feedback bits to specify the number additional bits to be requested, the output of the policies will be restricted to a pre-defined set of $2^k$ values, but the optimization of those values for different values of $k$ is beyond the scope of this paper.
V. Multi-codeword IR

So far we have assumed that the system devotes a specific number of feedback bits to acknowledge or request IR for each individual codeword, but this is rather inefficient in practice, since there exists significant correlation between adjacent codewords (in time or in frequency). If the SNR is high, most codewords in a packet will be correctly decoded, and in the event of blockage or beam misalignment many codewords will fail requiring a similar amount of incremental redundancy.

Instead of devoting a fixed number of feedback bits to each LDPC codeword, the system could group the codewords into bundles that can be acknowledged with a single bit\(^5\). In the event that some of these codewords fail to decode, the receiver would provide feedback about the bundle as a whole. The transmitter would then send a few additional parity bits for each codeword or a larger amount of incremental redundancy coupling all the codewords in the bundle.

If a single codeword fails in a bundle, the transmitter can provide IR for that codeword even if it is not able to pinpoint the exact codeword that failed: it just needs to compute the IR blocks for all the codewords in the bundle and transmit their bit-wise XOR. The receiver can also compute the IR for the codewords that have been successfully decoded, so it can eliminate those components to recover the IR for the failed codeword.

This same idea can be extended to more than one failure using maximum distance separable (MDS) array and erasure codes [50]–[53], as shown in Fig 6. Some of these array codes can be encoded and decoded using cyclic shifts and XOR operations only, hence simplifying the derivation of the LLRs and avoiding the finite field arithmetic required for Reed-Solomon (RS) and typical MDS codes. Linear codes (which include LDPC) have the property that any linear combination of codewords is also a codeword [54]. Therefore, when the transmitter sends a linear combination of codewords, the receiver can attempt to decode it and correct any errors that the channel might have introduced. If the total number of codewords (or combinations of codewords) that failed decoding is below the error correction capability of the erasure code, they can all be recovered by solving a linear system of equations. If the number of failures exceeds the error correction capability of the code, the values received from the channel can still be leveraged, for example by alternate decoding of the LDPC and erasure codes.

---

\(^5\)This is known as acknowledgement bundling [46], [47], not to be confused with redundant bundling [48], [49] where packets are preemptively retransmitted to reduce latency.
Instead of sending incremental redundancy for each individual codeword, we can construct an erasure code over a bundle of codewords.

There is a trade-off between using erasure codes and individual codeword IR. This trade-off was studied in [55] for the case with random and uncorrelated decoding failures. In a nutshell, erasure codes need to transmit an integer number of additional codewords (which is generally more bits than are needed), but these codewords can be decoded which makes erasure codes more reliable. On the other hand, erasure codes need additional feedback to specify the number of failed codewords to be recovered.

Let $b$ denote the number of codewords per bundle and $P_b(x)$ the probability that a bundle suffers $x$ decoding failures, for $x = 1, \ldots, b$. Assuming a single IR round and that the whole bundle is retransmitted in case of failure, the average data rates when the transmitter sends $\Delta(x)$ individual parity bits or $\beta(x)$ additional codeword combinations in response to $x$ failures are respectively given by

$$r_{\text{ir}} = \frac{k}{N} \cdot \frac{1 - P_{\text{fail}}}{1 + \frac{1}{N} \sum_{x=1}^{b} P_b(x) \Delta(x)}$$

$$r_{\text{er}} = \frac{k}{N} \cdot \frac{1 - P_{\text{fail}}}{1 + \frac{1}{b} \sum_{x=1}^{b} P_b(x) \beta(x)},$$

where $\frac{k}{N}$ is the rate of the LDPC code and $P_{\text{fail}}$ represents the probability of failure after IR. Figure 10, described in the next section, illustrates this trade-off.

VI. SIMULATION RESULTS

In this section, we evaluate our IR algorithms through Monte Carlo simulations. Our simulation scenario is a MISO system with a ULA of $M = 16$, 32 antennas communicating over a channel
with a single line of sight (LOS) path and $R = 3$ non-line of sight (NLOS). The channel is modeled as

$$h = \sqrt{\frac{MK}{1+K}}\alpha_0 d_M(\psi_0) + \sqrt{\frac{M}{R(1+K)}} \sum_{r=1}^{R} \alpha_r d_M(\psi_r),$$

where $\alpha_r \sim \mathcal{C}\mathcal{N}(0,1)$ is the complex channel gain, and $\psi_r \sim \mathcal{U}(-1,1)$ is the beam direction of the $r$-th ray-like beam. Note that the Ricean-$K$ factor set to $K = 13.5$ dB based on channel observations [11], [29], [56].

The probability of decoding failure described in Eq. (8) depends heavily on the error correcting code being used. Therefore, before evaluating the algorithms, we need to characterize the probability of failure for the code that will be used, namely the 802.11n with length $U = 648$ and code rate $R' = 2/3$ [32]. Fig. 7 shows the empirical frame error rate for different modulations, SNR, and rates (obtained by extending the mother code). As predicted by [37] and [41], each of these curves can be well approximated by a Gaussian cumulative density function. We model the corresponding mean and variance as

$$\mu(\gamma_{\text{lin}}) = a_m \gamma_{\text{lin}}^{b_m} + c_m \quad (16)$$

$$\sigma(\gamma_{\text{lin}}) = a_s \gamma_{\text{lin}}^{b_s} + c_s \quad (17)$$

where the effective SNR is in linear scale ($\gamma_{\text{lin}} = 10^{\frac{\gamma}{10}}$) and the curve fitting parameters for each modulation are summarized in Table I.

We now evaluate the data rate of the adaptive IR algorithm proposed in Section IV, assuming that the number of incremental transmission rounds is limited to $T = 4$ and that the model in
Table I is accurate. Observe that the maximum possible data rate is $2/3$ for BPSK, but it can surpass that value for higher order modulations.

Fig. 8 shows the average data rate per transmitted symbol as a function of the SNR variance for Chase combining, fixed-length IR, and our adaptive IR scheme. The signal to noise ratio (SNR) is assumed to follow a Gaussian distribution with mean 0 dB for BPSK, 5 dB for 4-QAM, and 10 dB for 16-QAM. All three schemes provide approximately the same data rate when the SNR is constant, since the ECC and modulation order have been chosen to avoid decoding failures, but they diverge as the amount of uncertainty in the SNR increases.

In Chase combining, the whole codeword is re-transmitted whenever the decoding fails. It is the simplest to implement, but the resulting average data rate is significantly lower than that with the other schemes. The scheme labeled “Fixed $\Delta$” transmits a different number of bits in each re-transmission round, optimized based on the average SNR as proposed in [41]. The proposed scheme, labeled “Adaptive $\Delta$”, estimates the SNR of failed codewords and requests a different number of incremental redundancy bits depending on that estimate: when the SNR is low it will request more IR bits than when it is large. That way, this scheme is able to maintain a relatively
high data rate despite the SNR uncertainty, while the one with fixed increment lengths suffers significant degradation.

One alternative is to use the feedback bits to specify which bit combinations should be transmitted, instead of using them to specify the number of additional bits to be transmitted. This is known as active-feedback and has been recently studied in other papers [43], [44]. A third alternative, specially useful when the codewords are bundled, is using the feedback bits to specify the number of codewords whose decoding failed. The transmitter can then use erasure codes as described in Section V.

We now consider the case where the number of feedback bits is significantly smaller than the number of codewords. Since there is not enough capability to provide individualized feedback, codewords need to be bundled. Two schemes are compared: an incremental redundancy (IR) scheme with one feedback bit for each bundle of codewords and an erasure coding scheme with two feedback bits for each bundle of twice as many codewords. A single bit of feedback can only convey ACK/NACK information, so the IR scheme will transmit a small number of incremental redundancy bits for each codeword in the bundle if any of them fails to decode. The erasure code scheme can either request parity codewords or IR depending on the number of failures.

Whenever codewords are bundled, it is important to consider the correlation introduced by the channel. If the link gain drops for one codeword, it is likely to remain low for the next one, and vice versa. Our simulations will model such correlation with the Markov chain in Fig. 9, which has two states with different SNR and a probability $p$ of transitioning from the high to the low SNR state. Once in the low SNR state, the system is expected to remain there for an average of 9 codewords before returning to the high SNR state.

Figure 10 shows the average data rate achieved by both schemes in a system with 128 code-
words per frame and a single round of retransmission. The number of incremental redundancy bits has been optimized for each value of \( p \) in Fig. 9, but independently from the actual link gain observed for each codeword. The number of codewords per bundle will be \( b = \frac{128}{N_{\text{feedback}}} \) for the IR scheme and twice that for erasure coding, where \( N_{\text{feedback}} \) represents the number of feedback bits available. The number of parity codewords requested by the latter changes depending on the number of feedback bits: the receiver can request one or two erasure parities when \( N_{\text{feedback}} = 8 \) and two or four when \( N_{\text{feedback}} = 4 \). If the number of failures is higher than that, the erasure scheme will request IR for all the codewords in the bundle.

The results show that erasure codes are better in the high SNR regime, when the number of failures is small. This makes sense, since the IR scheme cannot know which codeword failed in each group, and would need to send IR for all of them. However, as it becomes more frequent to have codewords with low SNR, it is preferable to use incremental redundancy, since there is a high chance that the additional codewords will also fail to decode. It can also be observed that, in the high SNR regime, all the curves are close to the upper bound (found by using one feedback bit per codeword), while in the low SNR regime they are close to the lower bound (found by using a single bit to acknowledge or request IR for all 128 codewords in a frame).
VII. CONCLUSION

This paper studied some of the trade-offs arising in the use of incremental redundancy schemes in binary LDPC-coded mmWave systems. It analyzed how the mapping of bits and codewords to time, frequency, and modulation resources can affect the average latency and throughput in the network and proposed HARQ schemes using codeword-specific incremental redundancy and erasure codes over bundled codewords.

Specifically, we developed an adaptive scheme for adjusting the number of incremental redundancy bits sent for an unsuccessfully decoded codeword, based on instantaneous rather than averaged channel quality estimates. It was shown through simulations that the rapid response and flexible adaptation of this scheme can provide significant throughput gains in scenarios with uncertain channel quality.

However, in some scenarios it is possible to achieve higher average throughput by bundling multiple codewords for feedback and incremental transmission purposes. Instead of extending the failed codewords with additional parity bits, the transmitter sends combinations coupling all the codewords in a bundle. This paper analyzes the trade-off between these two approaches, as well as the effect that the size of the bundles and the number of feedback bits has in the average throughput of the proposed schemes.

ACKNOWLEDGMENT

This work was supported in part by Nokia Networks and the National Science Foundation under grant CNS1642982.

REFERENCES


