HARQ Strategies for Relay Systems with Limited Feedback

Mai Zhang, Borja Peleato
Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907 Email: {maiz,bpeleato}@purdue.edu

***Abstract*—Hybrid automatic repeat request (HARQ) techniques based on rate compatible error correction codes (ECC) have been extensively studied to maximize the achievable data rate while maintaining error correction performance. However, previous works have focused on single-hop transmissions of a single codeword, whereas in practical situations, a direct link between source and destination is not always available and codewords are often bundled into packets. This research will propose HARQ techniques suitable for multi-hop relay systems with codeword bundling and limited feedback capabilities. It will analyze the trade-offs when choosing the type and amount of incremental redundancy (IR) as well as different relay strategies.**

I. Introduction

Hybrid automatic repeat request (HARQ) techniques based on rate compatible codes have been shown to effectively increase the data rate in wireless communication systems [1], [2], [3]. However, previous works have focused on single-hop communications and have failed to leverage the statistical distribution of received signals to choose the content of subsequent transmissions in the HARQ process. Assuming a closed loop feedback link, we will study techniques for optimizing the HARQ process in multi-hop relay communication systems with limited feedback capabilities.

Traditional ARQ forces the receiver to send an ACK back to the transmitter for every packet it successfully decodes, and a NACK otherwise. If the transmitter does not receive an ACK before the timeout expires, the entire packet will be resent. However, it is often inefficient to retransmit the whole packet when the receiver can in fact leverage some of the received information and successfully decode the whole packet with a few additional bits, known as incremental redundancies (IR). This technique is commonly known as Type-II hybrid ARQ (Type-II HARQ) [4], and it will be the focus of this paper. Often, the feedback channel has limited capacity, so multiple codewords are grouped into a bundle to be acknowledged together. A packet may consist of several bundles too.

Optimizing the HARQ strategies for a relay system will become increasingly important in future millimeter wave (mmWave) systems, which will require a dense network of access points acting as relays between a base station and the end users. Usually, each relay station has two ways of forwarding the information, namely amplify and forward (AF) and decode and forward (DF). In AF, the relay amplifies and transmits whatever signal it receives. Both the information and noise components are amplified. In DF, the relay decodes the received signal first, requesting IRs if necessary until success; it then re-encodes and transmits the message to the next hop.

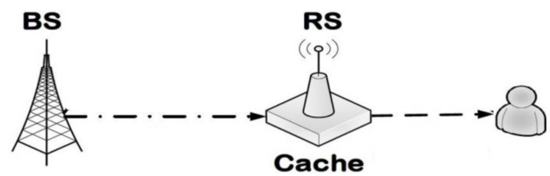

Fig. 1. Relay system model

Previous literature has shown that DF generally has better performance than AF, but this was based on channel capacity and it did not consider some of the practical benefits of AF, such as simpler hardware and lower latency [5]. Moreover, [6] analyzed several HARQ protocols suitable for a relay system, but the analysis of the performance was based on frame error rate. We will propose a method that considers the practical complexity of decoding and retransmissions by associating tunable costs to them, and we will present an optimization framework to minimize the average costs per information bit sent. By adjusting the relative costs, this method can be used to model practical constraints such as latency.

The rest of the paper will be organized as follows. Section II defines the system model and introduces the parameters used. Section III builds the decision engine for a single link scenario. Section IV derives the decision engine for the relay. Finally, section V will show numerical results to illustrate our proposed policy, and section VI concludes the paper.

II. System Model

We consider a multi-hop relay system as depicted in Fig. 1. There are data channels from base station (BS) to relay station (RS), and from relay to the end user. The end user cannot hear the base station so they must communicate via the relay in the middle. Both data channels are equipped with an independent error-free side channel that allows the receiver to acknowledge and provide limited feedback to the transmitter.

A. Channel and Error Correction

There exists a certain amount of correlation in the channel experienced by adjacent codewords (adjacent in time or frequency). For simplicity, the channel will be modeled as an interference-free AWGN with variable SNR. It will be assumed that all the codewords in a given bundle experience the same SNR, but it is possible that subsequent transmissions of incremental redundancy are received with different quality.

Most standards and prototype systems have adopted binary QC-LDPC codes due to their outstanding performance and parallel architecture [7], [8]. Furthermore, they can be easily punctured or extended to adapt their coding rate.

LDPC codes are linear block codes characterized by a sparse parity check matrix $H \in \{0,1\}^{(n-k)\times n}$. Received symbols are processed to obtain a log-likelihood ratio (LLR) for each individual bit b as

$$\ell = \text{LLR}(b|r) = \log \frac{p(0|r)}{p_i(1|r)}, \quad (1)$$

where $p(0|r)$ and $p(1|r)$ respectively represent the probability of b being 0 or 1, conditioned on the received value r. These LLRs are then progressively refined by iterative message passing over the Tanner graph of H, until they converge to a feasible codeword or the algorithm reaches a maximum number of iterations. LDPC decoders very rarely converge to a wrong codeword; it is much more likely that they simply fail to converge by the maximum number of iterations.

Our optimization and simulations will focus on the QC-LDPC code of length $n = 648$ and $k = 432$ (rate 2/3) proposed in the 3GPP standard for 802.11n [9], but the techniques proposed here could be applied to any other code by adjusting the FER characteristics. As shown in [3], the FER for this code (and extensions) can be well approximated by

$$P_e(SNR, R) = Q\left(\frac{\mu - R}{\sigma}\right), \quad (2)$$

where $\mu = -0.2\cdot SNR^{-1.74}+0.86$, $\sigma = 0.12\cdot SNR^{-0.42}-0.08$, SNR is in linear scale, and R represents the code rate.

B. Single Link System: Hybrid ARQ

It will be assumed that the feedback channel offers at most one bit of feedback for each codeword. This gives 16 possible feedback messages for a bundle of 4 codewords, for instance. However, it may turn out that our HARQ strategy does not require that many feedback messages per bundle, in which case the required number of feedback bits can be lower. Furthermore, we assume that the receiver can request an unlimited number of rounds of incremental redundancy, until the whole bundle is successfully decoded. Each round incurs a constant overhead cost of c_R to account for additional complexity and latency.

C. Relay Decision: Amplify or Decode?

The relay can decide between AF and DF, based on its SNR estimates and the code rate. If it chooses DF, the system operates as two independent links using the same single link HARQ protocol. The base station first transmits to the relay, with retransmissions if necessary until the relay can successfully decode the entire bundle. The relay then transmits to the end user in the same way, and it has all the required knowledge to compute the IRs should the end user request any.

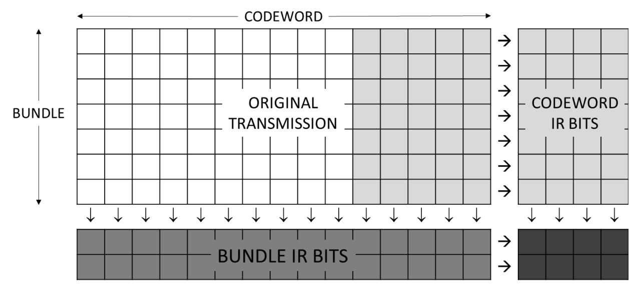

Fig. 2. Types of incremental redundancy.

If the relay chooses AF instead, we assume that it caches the LLRs from the demodulator output before forwarding them to the end user. If the end user successfully decodes the bundle of codewords, we have saved time and computation at the relay by skipping decoding, and the cached LLRs can be discarded. However, if the end user fails to decode any of the codewords, the relay will decode the bundle based on its cached LLRs (with the help of HARQ if needed) before sending IRs to the end user. Note that in this case, the subsequent IRs received at the end user will most likely have a higher SNR than the original bundle.

Theoretically it is better to amplify the signal as much as possible at the relay, so that the perceived SNR at the end user approaches the SNR on the first link. However, due to practical power constraints, we assume that the base station transmits at unit power, and the relay will amplify its received signal such that the original signal component has unit average power too. In other words, suppose the base station sends the signal x with unit power $E[x^2] = 1$, and the relay receives

$$y_1 = g_1 x + n_1,$$

where g_1 is the channel gain and n_1 is the Gaussian noise with variance σ_1^2. The relay amplifies y_1 by a factor of $1/g_1$, and sends it over the second channel, so the end user receives

$$y_2 = g_2 \frac{1}{g_1} y_1 + n_2 = g_2\Big(x + \frac{n_1}{g_1} + \frac{n_2}{g_2}\Big).$$

Using the definition $SNR_j = g_j^2/\sigma_j^2$ ($j = 1, 2$) and the fact that the noise components on the two links are independent, we see that the SNR of y_2, *i.e.* the SNR at the end user due to AF, can be found as

$$SNR_{AF} = \frac{E[x^2]}{\text{Var}[\frac{n_1}{g_1} + \frac{n_2}{g_2}]} = (SNR_1^{-1} + SNR_2^{-1})^{-1}, \quad (3)$$

which is always lower than the SNR on either link.

III. Decision Engine for Single Link

A. Incremental Redundancy

The term "Incremental Redundancy" includes any additional bits requested by a receiver so as to attempt the decoding of a codeword (or bundle of codewords) that had previously failed. As shown in Fig. 2, these bits can be of different types, depending on how they are constructed:

1) Codeword parity (or extension) bits: generate new parity for each codeword by extending the matrix H with new rows and columns representing combinations of bits not previously used.
2) Bundle parity bits: construct a bitwise erasure code across multiple bundled codewords [10]. For simplicity we use the bitwise XOR across all the codewords.

Practical LDPC decoders have limited memory, often insufficient to handle a joint decoding of all the codewords and IR in a bundle. Therefore, we assume that each codeword is decoded independently, after using the bundle parity bits to refine the LLR values. Codeword parity bits, however, can be directly fed to the LDPC decoder. Our previous work [3] studied the effect of IR on the bundle in detail.

Bundle parity bits are used to refine the LLRs in each column as follows. Let ℓ_i denote the LLR corresponding to the i-th codeword in the bundle, and let ℓ_{n+1} denote the LLR of their XOR (*i.e.* bundle parity). The updated LLRs are calculated as

$$\ell_k^{\text{new}} = \ell_k + \left(\prod_{i=1, i\neq k}^{n+1} \operatorname{sign} \ell_i\right) \min_{\substack{i=1\dots n+1\\ i\neq k}} |\ell_i|, \tag{4}$$

for $k = 1 \dots n+1$. The SNR of the bits will be improved after such an update, as shown in [3].

Codeword extension bits reduce the rate of the code. This in turn reduces the probability of decoding error according to Eq. (2) with

$$\mathit{SNR}_{\text{eff}} = (E[\mathit{SNR}^{-1}])^{-1}. \tag{5}$$

This effective SNR captures the effect of different SNR qualities in the codeword and was justified in [3].

B. Optimization

Our previous work [3] derived a protocol to optimize the HARQ strategy for the single link scenario. The protocol tells the receiver the type and number of IR bits that it should request, given the SNR and code rate of the codeword. That work assumed that the IRs will have the same SNR as the original codeword. We now extend the method to include an additional variable SNR_{IR}, i.e. the expected SNR of the upcoming IRs.

In order to make the problem manageable, the SNR and rate R are quantized to take a finite number of values. Furthermore, the number of IR bits requested is also restricted to a small pre-defined discrete set, so as to limit the number of feedback bits required to make such a request. The HARQ protocol for a bundle of codewords can then be modeled as a Markov Decision Process (MDP) with a finite set of actions and states:

- State: $s = (f, \mathit{SNR}, R)$, where f represents the number of codewords in the bundle whose decoding failed, SNR their effective SNR, and R their coding rate.
- Action: $A(s) = (\alpha, \beta)$, where α represents the number of extension bits requested (per codeword in the bundle) and β represents the number of bundle parity bits requested.
- Cost: $C = b\alpha + \beta + fc_D + c_R$, where b denotes the number of codewords per bundle (*i.e.* bundle size). We assume that it costs 1 unit to transmit 1 bit; it costs c_D to decode a single codeword (we only need to decode the codewords which failed in the previous round). Finally c_R represents the overhead associated to each round of retransmission, which accounts for feedback bits, increased latency, hardware complexity etc.

The objective is to minimize the total cost until successfully decoding the bundle, *i.e.* we wish to find

$$A(s) = \arg\min_{\alpha,\beta} E\{\text{Total cost}|s, \alpha, \beta\} \tag{6}$$

for all s. The IR bits (α, β) will reduce the code rate and increase the SNR, transitioning $s_1 = (f_1, \mathit{SNR}_1, R_1)$ to a new state $s_2 = (f_2, \mathit{SNR}_2, R_2)$ where SNR_2 and R_2 are deterministic, and $f_2 \leq f_1$ follows the binomial distribution. They are characterized by the following equations:

$$\begin{cases} \mathit{SNR}_2 = \left[\left(\frac{\alpha}{\mathit{SNR}_{IR}} + \frac{\beta}{\Delta(\mathit{SNR}_{1;IR})} + \frac{k/R_1 - \beta}{\mathit{SNR}_1}\right)\frac{1}{k/R_1+\alpha}\right]^{-1} \\ R_2 = \frac{k}{k/R_1+\alpha} \\ P(f_2|s_1,\alpha,\beta) = \binom{f_1}{f_2} p^{f_2}(1-p)^{f_1-f_2} \end{cases} \tag{7}$$

where $\Delta(\mathit{SNR}_{1;IR})$ denotes the effective SNR of the bundle after we update the LLRs based on the bundle IR bits with SNR_{IR}. The formula for SNR_2 is obtained by applying Eq. (5), and using the fact that each codeword can be divided into three sections of different SNR qualities: α bits of IRs have SNR_{IR}; the first β bits in the codeword have SNR of $\Delta(\mathit{SNR}_{1;IR})$ after their LLRs are updated; finally the remaining $k/R_1 - \beta$ bits have the same SNR_1 as before.

The probability p in Eq. (7) represents the conditional probability that a codeword fails in state s_2 conditioned on its failure in s_1, and can be evaluated based on the frame error rate function given in Eq. (2) as $p = \frac{P_e(\mathit{SNR}_2, R_2)}{P_e(\mathit{SNR}_1, R_1)}$.

We now wish to find the expected future cost V and the optimal action A to take given a current state s and SNR_{IR}:

$$\begin{aligned} V(s, \mathit{SNR}_{IR}) &= E[\text{Total cost until success}|s,\alpha,\beta] \\ &= c_R + b\alpha + \beta + fc_D + \sum_{s\to s'} P(s'|s,\alpha,\beta)V(s') \end{aligned} \tag{8}$$

$$A(s, \mathit{SNR}_{IR}) = \arg\min_{(\alpha,\beta)} V(s, \mathit{SNR}_{IR}). \tag{9}$$

With the states discretized to a finite number of values, we can use the value iteration algorithm [11] to optimize V and A for all s and SNR_{IR}. Essentially, it starts with a random value function and alternates between updating the policy A according to Eq. (9) and the value V according to Eq. (8), until they converge. At that point $A(s, \mathit{SNR}_{IR})$ stores the policy to be followed whenever the system is in state s and the IRs expect SNR_{IR}; $V(s, \mathit{SNR}_{IR})$ stores the total expected future cost until the receiver can decode the entire bundle successfully.

The action A describes the decision engine for the single link model. This decision engine can be applied to the end user when it asks for IRs from the relay, or to the relay when it wishes to get the information bits from the base station. In any case, when the receiving end suffers decoding failures, it first determines its state by estimating the SNR of the bundle and the subsequent IRs, and then follows A to ask the transmitter for a combination of IR bits (α, β).

IV. Decision Engine for Relay

Now that we have a decision engine for a single link, we may derive the decision engine for the relay. In order to compare AF and DF, we build a cost model for each of them, and program the relay to adopt the strategy which minimizes the total expected cost per information bit delivered to the destination. Specifically, we find the cost of AF and DF (c_{AF} and c_{DF}) as functions of SNR_1, SNR_2 and R_1, which respectively denote the SNR on the first and second link, and the code rate used on the first link.

Likewise to the single link decision engine, it costs 1 unit to transmit 1 bit of information per link; c_D and c_R denote the overhead costs of decoding one codeword and each retransmission request. We will calculate the total expected cost of transmitting one bundle from the base station to the end user, accounting for all the costs in the system.

A. Cost of DF

When the relay chooses DF, the system is equivalent to having two independent links. Thus we write the cost of DF as

$$c_{DF} = c_1 + c_2, \tag{10}$$

where $c_j := E[\text{cost on } j\text{-th link}]$ $(j = 1, 2)$. We express c_j as the sum of three contributing terms: the number of bits sent on the j-th link is equal to bk/R_j; the cost of decoding b codewords in the bundle is bc_D; finally the expected future cost if there are decoding failures.

$$c_j = \frac{bk}{R_j} + bc_D + \sum_{i=1}^{b} P_B(b, p_j, i)\delta_j(i), \tag{11}$$

where $\delta_j(i)$ represents the expected future cost on the j-th link given that there are i failures in the bundle, $p_j = P_e(SNR_j, R_j)$ is obtained from Eq. (2), and $P_B(b, p_j, i) := \binom{b}{i} p_j^i (1 - p_j)^{b-i}$. The code rate on the second link R_2 is chosen such that c_2 is minimized. Finally, we express the expected future cost as

$$\delta_j(i) = V((i, SNR_j, R_j), SNR_j), \tag{12}$$

where we utilized Eq. (8) from the single link scenario. Here we assumed that the IRs can expect the same SNR as the original codewords, hence the last argument is SNR_j.

B. Cost of AF

Suppose the relay forwards the bundle without changing the code rate, *i.e.* $R_2 = R_1$, then the number of bundle bits sent over the two channels are the same. The end user spends bc_D to decode the bundle, and if it finds that $i \geq 1$ codewords fail to decode, it needs to ask for retransmission. Thus

$$c_{AF} = 2 \cdot \frac{bk}{R_1} + bc_D + \sum_{i=1}^{b} P_B(b, p_{AF}, i)\delta_{AF}(i), \tag{13}$$

where $p_{AF} = P_e(SNR_{AF}, R_1)$, SNR_{AF} is obtained from Eq. (3), and $\delta_{AF}(i)$ denotes the expected future cost given that there are i failures at the end user. In the case that $i \geq 1$, the relay will try to decode the bundle that it cached, and ask the base station for IRs if necessary until the relay decodes the entire bundle. If there are j failed codewords at the relay, getting the whole bundle correct at the relay will cost $V((j, SNR_1, R_1), SNR_1)$. The relay will then be able to transmit any IRs that the end user requests. This step costs another $V((i, SNR_{AF}, R_2), SNR_2)$, since the end user is in state (i, SNR_{AF}, R_2), and the additional IRs from the relay to the end user will have SNR_2.

We assume that any codeword that fails at the relay will also fail at the end user because it is extremely unlikely that the random Gaussian noise on the second link will coincidentally cancel out the errors in the codeword. In other words, if a codeword decodes successfully at the end user, it must also be successful at the relay. As a result, given that there are i failures at the end user, the number of failures at the relay j follows the binomial distribution $B(i, p_R)$ where p_R represents the conditional probability that a codeword fails at the relay given that it failed at the end user. Thus we see that $\delta_{AF}(i)$ can be expressed as

$$\delta_{AF}(i) = bc_D + V((i, SNR_{AF}, R_2), SNR_2) + \sum_{j=1}^{i} P_B(i, p_R, j) V((j, SNR_1, R_1), SNR_1), \tag{14}$$

where $p_R = \frac{P_e(SNR_1, R_1)}{P_e(SNR_{AF}, R_1)}$ follows from Bayes's rule.

We can now compute c_{DF} and c_{AF} for all discretized values of SNR_1, SNR_2, and R_1 using Eqs. (10) and (13). A decision map is then generated depending on which of the two forwarding method gives a smaller expected cost. In a practical situation, the relay can estimate the SNR on the two links and find the rate of the code that it receives, and make its decision accordingly.

V. Numerical Results

We now focus on the QC-LDPC code of length $n = 648$ and $k = 432$ (rate 2/3) [9] and show the numerical results from our analysis. We use a bundle size of $b = 4$; decoding and retransmission overheads are $c_D = 300$ and $c_R = 100$.

A. Single Link

Fig. 3 shows the policies obtained for α and β at rate $R = 0.6$ and $SNR_{IR} = -1.5$ dB, *i.e.* the number of extension bits (α) and bundle parity bits (β) to be requested as a function of the number of failed codewords remaining in the bundle and the effective SNR of those codewords for the known SNR_{IR} and R. It can be observed that, as the SNR decreases, the total number of IR bits to be requested increases. This makes sense, since highly corrupted bundles will require more IR for successful recovery. Also, when the number of failures is small our policy suggests requesting bundle parity bits instead of extension bits. This is worth noticing, since when there is a single failure, bundle parity is equivalent to Chase combining,

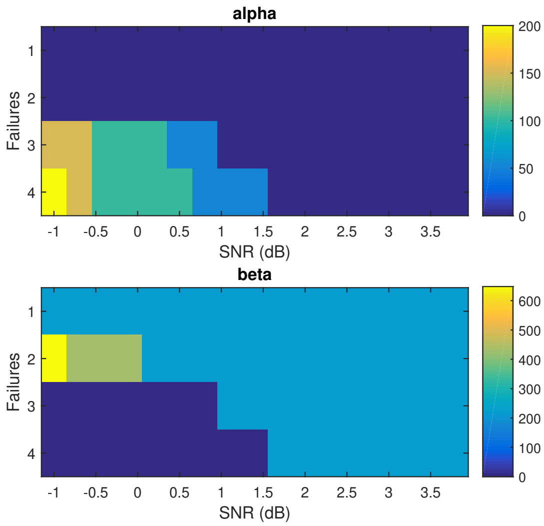

Fig. 3. α and β decision for $R = 0.6$ and $SNR_{IR} = -1.5$ dB

which is usually inferior to extension bits [12]. However, this can be explained in our system because the receiver cannot convey to the transmitter which codewords failed specifically; the transmitter will have to send extension bits for every codeword in the bundle, even for those that have already been successfully decoded.

The policies in Fig. 3 have less than 16 possible combinations of (α, β), so 4 bits of feedback are sufficient to specify the retransmission strategy. This requires 1 feedback bit per codeword, which is comparable to traditional fixed IR schemes with individual acknowledgments.

B. Relay

The numerical values of c_{AF} and c_{DF} are found and compared to obtain the decision map. The relay estimates the SNR of the two channels, finds the code rate used on the first channel, and refers to the decision map to decide whether to use AF or DF. Fig. 4 shows the decision map for $R_1 = 0.5$.

It can be seen that when both SNR_1 and SNR_2 are high enough, the relay prefers AF. This is because when the resultant SNR_{AF} is high, AF has little risk of decoding failure at the end user while alleviating the decoding cost at the relay. Our simulation also showed that as we increase the code rate R_1, the region where AF is better moves to the right. This makes sense, since increasing the code rate requires higher SNR to maintain the aforementioned low risk of decoding failure.

VI. Conclusion

This paper proposes HARQ techniques suitable for a multi-hop relay system. It first models the single-link HARQ protocol as a Markov Decision Process aiming at minimizing a cost function, which results in a set of incremental redundancy policies parameterized by the number of failures in the bundle, average SNR, and the coding rate. It then derives the decision engine for the relay, *i.e.* a map the relay follows to decide whether to amplify and forward or decode and forward, based on the expected SNRs and the coding rate. The deivations are verified through numerical resutls.

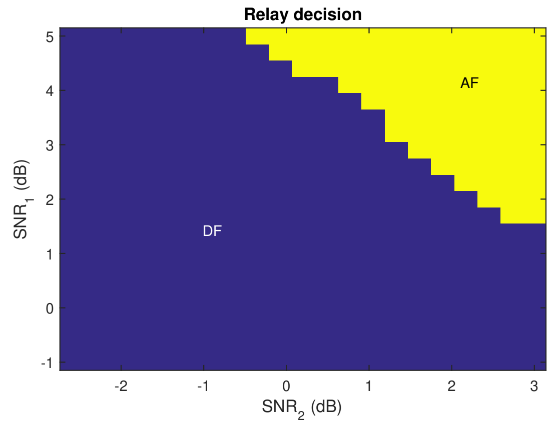

Fig. 4. Relay decision map, shown for $R_1 = 0.5$

References

[1] R. D. Wesel, K. Vakilinia, S. V. Ranganathan, and D. Divsalar, "Resource-aware incremental redundancy in feedback and broadcast," in *International Zurich Seminar on Communications*, 2016, p. 63.

[2] K. Vakilinia, S. V. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary LDPC examples," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2245–2257, 2016.

[3] M. Zhang, A. Castillo, and B. Peleato, "Optimizing HARQ feedback and incremental redundancy in wireless communications," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[4] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Communications magazine*, vol. 22, no. 12, pp. 5–17, 1984.

[5] G. Levin and S. Loyka, "Amplify-and-forward versus decode-and-forward relaying: Which is better?" in *22th International Zurich seminar on communications (IZS)*. Eidgenössische Technische Hochschule Zürich, 2012.

[6] K. Pang, Y. Li, and B. Vucetic, "An improved hybrid arq scheme in cooperative wireless networks," in *2008 IEEE 68th Vehicular Technology Conference*. IEEE, 2008, pp. 1–5.

[7] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 71–81, 2006.

[8] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 1, pp. 104–114, 2007.

[9] IEEE, *IEEE 802.11n Wireless LAN Medium Access Control MAC and Physical Layer PHY specifications.*, IEEE 802.11n-D1.0 Std., 2006.

[10] T. A. Courtade and R. D. Wesel, "Optimal allocation of redundancy between packet-level erasure coding and physical-layer channel coding in fading channels," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2101–2109, 2011.

[11] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[12] P. Frenger, S. Parkvall, and E. Dahlman, "Performance comparison of HARQ with Chase combining and incremental redundancy for HSDPA," in *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, vol. 3. IEEE, 2001, pp. 1829–1833.