SNR-Responsive Communication: Turbo Codes and BCH in a Dynamic HARQ Scheme for Enhanced Efficiency

C. Gnana Jyothi[1], P. Chandana Sai Sri Hasitha[1], N. Yaswanth[1], B. R. Chandavarkar[1*]

[1*]Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangaluru-575025, Karnataka, India.

*Corresponding author(s). E-mail(s): brc@nitk.edu.in;
Contributing authors: chinthagnanajyothi.221cs118@nitk.edu.in; prathapachandanasaisrihasitha.221cs139@nitk.edu.in; namburiyaswanth.221cs232@nitk.edu.in;

Abstract

Wireless communication systems face many challenges due to fluctuating channel conditions, which lead to variable error rates and require robust error management tactics. Normal Automatic Repeat Request (ARQ) methods suffer from high communication overhead and latency and are inefficient in handling retransmissions. Although traditional Hybrid Automatic Repeat Request (HARQ) is effective, its widespread use is limited by high implementation costs. This underscores the need for adaptive systems that dynamically respond to changing channel conditions based on Signal-to-Noise ratio (SNR) values. This paper proposes the Dynamic HARQ (D-HARQ) that switches between ARQ, HARQ with Bose-Chaudhuri-Hocquenghem (BCH) codes, and HARQ with turbo codes based on channel conditions during the transmission. Further, this paper incorporates a selective soft combining technique by estimating the receiver end's signal-to-noise ratio (SNR). The main goal is to achieve the maximum possible throughput even at very low SNR values while maintaining optimal complexity.

Keywords: Adaptive HARQ, SNR estimation, Turbo codes, BCH codes, Selective soft combining, Error correction, Throughput optimization

1 Introduction

Error control mechanisms play a crucial role in reliable data transmission. It ensures the retransmission of data packets found to be erroneous or lost during the transmission. The end-to-end data transfer from one application to another involves several steps, each subject to errors along the way [1]. Due to the necessity of numerous retransmissions, standard Automatic Repeat Request (ARQ) schemes fall short of adequately correcting all errors. Thus, this paper leverages the technology of Hybrid ARQ (HARQ), which integrates ARQ with Forward Error Correction (FEC) [2, 3] for its enhanced efficiency in this context. However, this system may not be necessary under good channel conditions. To address this issue, this paper is developing a multi-functional approach that adjusts to the varying needs of different channel conditions.

1.1 Go-Back-N (GBN) Scheme

GBN scheme [4] is one of the Pure ARQ methods. In this scheme, the sender sends a sequence of frames without waiting for acknowledgment until a specified window size is reached. If errors are detected or the packet is lost, then negative acknowledgment (NACK) [5] is sent, and the sender resends all frames starting from the last acknowledged frame. Figure 1 illustrates the GBN scheme, where the sender (transmitter) transmits a sequence of frames (M1, M2, M3, ...) to the receiver. The receiver acknowledges the correctly received frames with ACK messages (ACK1, ACK2, ...). If a frame is lost or corrupted, the receiver sends a negative acknowledgment (NACK4 in the figure), prompting the sender to retransmit all frames starting from the lost frame (M4 in this case). This scheme does not require a storage buffer at the receiver side. GBN guarantees data integrity and reduces congestion of the network.

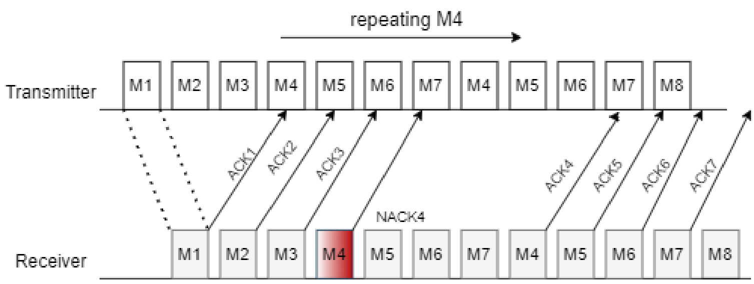

Fig. 1 Go Back-N (GBN) scheme [5]

1.2 Hybrid ARQ

Hybrid ARQ [6, 7] is ARQ concatenated with FEC. This facilitates error correction during transmission, minimizing the likelihood of receiving additional errors. Retransmission occurs when packet errors remain uncorrected. HARQ mainly has two basic schemes: Hybrid ARQ Type I, which has adaptive coding rates and extra error correcting codes to minimize retransmissions while keeping data integrity. Hybrid ARQ Type II uses redundancy to improve error correction capabilities later. These approaches guarantee robust data transmission, optimizing throughput, and reliable communication.

1.3 Turbo Codes

Turbo codes [8] are a part of error-correcting codes that employ parallel concatenated convolutional codes to achieve remarkably efficient error correction. They are most commonly used in 3G/4G mobile communications (e.g., LTE) and in satellite communications, where there is a high demand for reliable information transfer. Turbo codes offer superior performance in noisy channels by iteratively decoding received signals. They take advantage of the redundancy provided by multiple convolutional codes in parallel. This iterative decoding process allows turbo codes to approach the Shannon limit. In Turbo codes, the code rate refers to the ratio of information bits to the total number of transmitted bits. The natural coding rate of a turbo code is R = 1/3 (three output bits for one input bit).

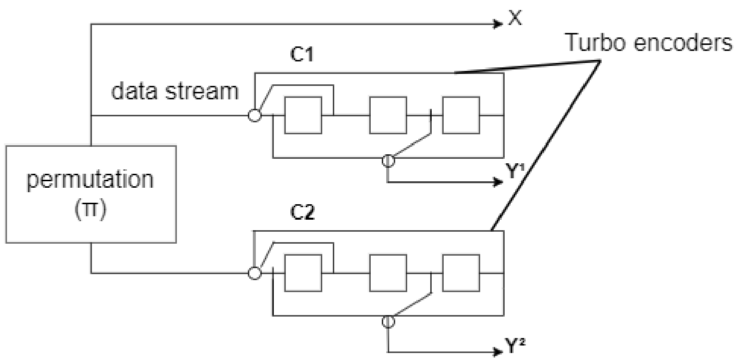

Fig. 2 A classical Turbo code [9]

The Figure 2 depicts a classical Turbo Code structure, which consists of the parallel concatenation of two binary Convolutional encoders, C1 and C2, separated by a permutation (interleaver) block labeled as π. The input data stream X is directly fed into the first encoder C1, producing the systematic output y1. The same input data X is also passed through the permutation block (π), which rearranges or interleaves the order of the data bits before feeding them into the second encoder C2, generating the parity output y2. Therefore, the Turbo Code encoder outputs are y1, which is an exact copy of the input data X (systematic output from C1), and y2, the parity output from C2, after interleaving the input data. The systematic and parity outputs (y1 and y2) are then transmitted. The key idea behind the Turbo Code structure is that the interleaved introduces redundancy and randomizes the input data, allowing the parallel encoders to provide robust error correction capabilities when decoding at the receiver end. Despite high error-correcting capability, they are quite complex to implement.

1.4 Soft Combining [10]

Soft combining is a technique in which a received erroneous packet is stored in a buffer rather than discarded. Two independently uncorrectable erroneous packets may be decoded correctly by utilizing the useful information obtained from their combination. This scheme has two main methods: chase combining and Incremental Redundancy (IR). In chase combining all retransmissions are identical, using maximum-ratio combining the received bits are combined with the error bits of previous transmissions, this adds extra energy for each retransmission by increasing the Eb/N_0 ratio.In incremental redundancy, different redundancy bits are obtained by puncturing the encoder output, thus adding additional information to each retransmission.

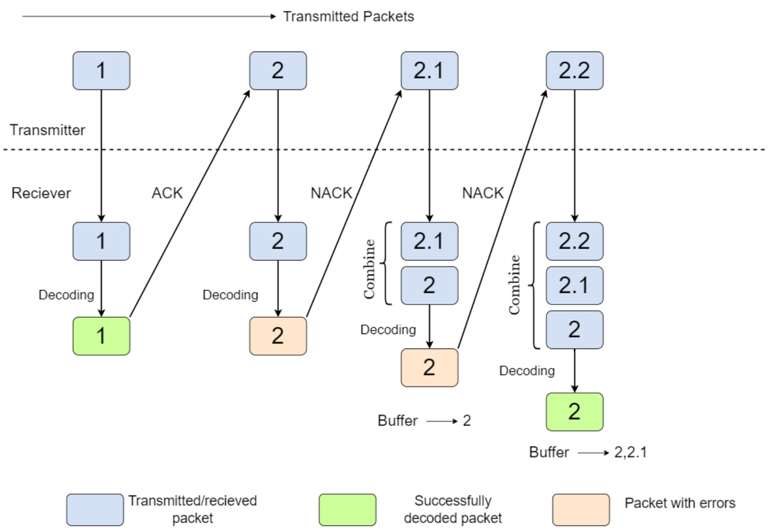

Fig. 3 Illustration of Soft combining with HARQ [11]

This Figure 3 illustrates the concept of soft combining with HARQ in a communication system. The transmitter sends packets, which are received and decoded by the receiver. An ACK is returned to the transmitter if a packet is correctly decoded. A NACK is sent and stored in a buffer if errors occur in the received packet. The transmitter then resends the erroneous packet, and the receiver combines (soft combining) the previously stored erroneous packet with the newly received packet to improve the decoding process. This combining process is repeated for subsequent NACKs and retransmissions, with the buffer storing the combined packets (e.g., 2, 2.1, 2.2.1).

The soft combining technique enhances the reliability of data transmission by utilizing the information from previous erroneous transmissions, leading to improved error correction and overall system performance.

There are several existing error-correcting mechanisms available. Some are easy to implement but do not offer strong error correction capabilities, while others are complex but provide robust error correction. Therefore, the choice of FEC [2] should be based on the channel conditions. Many existing approaches involve repeated retransmissions because they discard corrupted packets rather than utilizing them. These schemes hold good significance in DVB (Digital Video Broadcasting), storage systems (such as CDs, DVDs, and flash memory), and satellite communication systems [12].

This paper enhances system efficiency by dynamically switching among ARQ, HARQ with Turbo Codes, and HARQ with BCH codes based on fluctuating SNR values. It introduces a novel mechanism for selective soft combining packets at the receiver end based on current channel conditions. This approach minimizes retransmissions by utilizing corrupted packets rather than discarding them. Selecting the appropriate schemes at every data transmission stage will undoubtedly result in significant throughput improvements across all scenarios.

The rest of the paper is organized as follows: Section 2 covers the literature Survey, while Section 3 explains the design and implementation of the proposed solution. Section 4 presents simulation results, analysis, and comparisons with existing methods. Finally, Section 5 concludes with future directions and references.

2 Literature survey

This section discusses the various adaptive HARQ schemes followed by their advantages and drawbacks. Firstly, in Subsection 2.1, the Adaptive HARQ with Reed-Solomon (RS) codes utilizes a straightforward approach based on consecutive ACKs and NACKs to dynamically switch between HARQ and ARQ modes. Secondly, in Subsection 2.2, the Adaptive HARQ with two RS codes proposes a three-stage adaptive approach, dividing high channel error rates into HARQ1 and HARQ2 states with different RS code rates to optimize error correction efficiency. Lastly, in Subsection 2.3, the Adaptive HARQ with BCH codes dynamically switches between HARQ1 and HARQ2 schemes based on estimated channel states and SNR thresholds, effectively addressing a wide range of channel conditions using hybrid BCH code schemes.

2.1 Adaptive HARQ with Reed-Solomon codes

J. P. Peter Fidler et al. [13] introduced the adaptation rule that employs a straightforward approach based on counting consecutive acknowledgments (ACKs) and negative acknowledgments (NACKs) to dynamically switch between HARQ and ARQ modes, depending on the estimated channel state. As depicted in Figure 4, the key component of this model is the threshold parameter (T), which sets a limit on consecutive ACKs required to transition from HARQ to ARQ mode. Once the number of consecutive ACKs exceeds this threshold, the system shifts to the less complex ARQ mode, signifying a stable channel condition. Conversely, detecting even a single NACK prompts an immediate switch back to HARQ mode, reflecting the need for enhanced error correction in response to channel degradation.

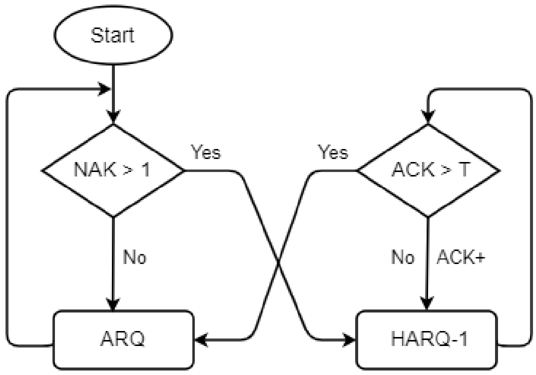

Fig. 4 Adaptive HARQ using RS codes [13]

This schema offers several advantages; firstly, its adaptive system dynamics introduce an asymmetrical strategy swiftly transitioning to HARQ mode upon detecting a noisy channel. This dynamic approach requires a threshold of positive acknowledgments to switch back to ARQ mode, ensuring the system remains robust and responsive to real-time channel conditions. This schema has several disadvantages, such as limited adaptability with only two states, which restricts the system adjustment to high fluctuations in channel quality. This has limited scalability for complex network environments.

2.2 Adaptive HARQ with two RS codes

Michal M. et al. [14] proposed a three-stage adaptive ARQ/HARQ1/HARQ2 scheme. The number of states is increased compared to the previously discussed adaptive HARQ scheme and can achieve relatively higher throughput for various channel bit error rates. There are three operating modes: pure ARQ -Go Back -N [12], HARQ1 with RS code (511, 383, 64), and HARQ2 with RS code (511, 255, 128). The transmitter follows the pure ARQ method for state L of low channel error rate. The previously discussed high channel error rate is now divided into HARQ1 and HARQ2. Both states use RS code but with different code rates.

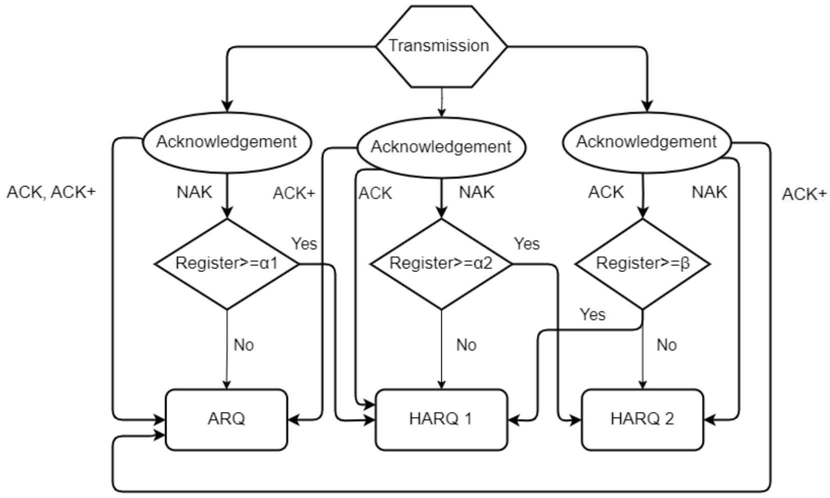

Fig. 5 Dynamic Adaption of different HARQ schemes using two RS codes [14]

Figure 5 explains that the switching logic operates so that the transmitter manages the switch from ARQ to HARQ modes or HARQ1 to HARQ2. The receiver handles the switch from HARQ modes to ARQ modes. These three confirmations are utilized in the proposal [14] as follows:

- ACK when the packet contains no errors (pure ARQ) or when the packet can be corrected by the RS code (HARQ1, HARQ2).
- NAK when the RS code cannot rectify the packet (HARQ1, HARQ2).
- ACK+ When there is no error in the previous W packets.

This paper brings forth several advantages; firstly, it introduces a pioneering approach through a 3-stage proposal with a sliding window mechanism. Unlike conventional ARQ-HARQ methods with fewer stages, this multi-stage scheme offers flexibility and adaptability to diverse channel conditions. However, this approach fails to handle a broader range of SNR values, optimize throughput, and minimize latency. This approach also has drawbacks, such as increased latency for real-time traffic and limited capability in correcting random errors due to its RS error-correcting code.

2.3 Adaptive HARQ with BCH codes

F. C. Kvetoslava Kotuliakova et al. [1] presented an enhanced adaptive ARQ-HARQ method [15] that uses BCH codes and is intended to improve the effectiveness of data transmission over varying channel conditions. This scheme dynamically modifies transmission schemes in response to the channel's measured bit error rate. Figure 6 depicts an adaptive HARQ model that dynamically switches between HARQ1 and HARQ2 schemes, which have different code rates, based on the estimated channel state and SNR threshold. The process starts by estimating the channel state. If the SNR exceeds the threshold, it uses the GBN scheme, which is more efficient for good channel conditions. Otherwise, it checks if the channel is in a bad state (HARQ2). If so, it uses the HARQ2 scheme with a higher code rate, suitable for poor channel conditions to improve reliability. If the channel is not in a bad state, it uses the HARQ1 scheme with a medium code rate. The system smoothly switches to hybrid ARQ schemes as the error rate rises to moderate to high levels.

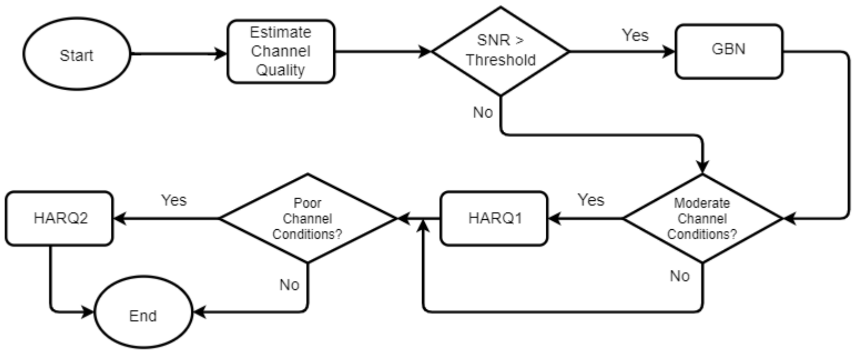

Fig. 6 Adaptive HARQ using BCH codes [1]

These hybrid schemes combine two BCH codes selected to improve error correction performance on various channel conditions. The GBN [12] protocol is used in every mode for better performance, as discussed in Subsection 1.1. This adaptive strategy shows great promise for establishing reliable and effective data communication in various difficult-to-manage channel conditions.

This scheme has various advantages that include dynamically altering parameters to adapt to changing network conditions and effectively controlling retransmissions, the scheme greatly lowers latency and improves system responsiveness. The disadvantages of this schema are having difficulties in maintaining effective data transmission under low SNR conditions. In such conditions it has almost zero throughput.

3 Design and Implementation of Dynamic HARQ (D-HARQ)

This section of the paper discusses the design and implemention of D-HARQ scheme to enhance communication efficiency. Firstly, Subsection 3.1 provides insights into throughput analysis for diverse schemes, including pure ARQ, HARQ with BCH, and turbo codes, elucidating their performance under varying channel conditions. Secondly, Subsection 3.2 outlines a dynamic switching scheme between HARQ modes and pure ARQ based on NACK/ACK counts and SNR thresholds. Thirdly Subsection 3.3 delves into the mechanism of HARQ with Turbo Codes, emphasizing error correction and throughput improvement in low SNR scenarios. Fourthly Subsection 3.4, discussing a robust SNR estimation algorithm based on Error Vector Magnitude (EVM) analysis is crucial for selective oft combining scheme. Lastly Subsection 3.5, discussing about decision making for soft combining at reciever end.

D-HARQ dynamically adapts to different schemes based on real-time fluctuations in SNR values. When SNR is very low, it switches to robust turbo codes for error correction with HARQ. Conversely, when SNR is moderate to high, it employs HARQ with BCH codes of different code rates Notably, when the SNR reaches an extremely high value, the scheme strategically switches back to pure ARQ to maintain simplicity. The scheme operates in four states: Pure ARQ with Go-back-N, two enhanced BCH codes HARQ1(4599, 3447) and HARQ2(4599, 2295), and an optimal state using HARQ with Turbo codes.

3.1 Throughput of different schemes

First stage of implementation is caluculation of throughput of different schemes like Pure ARQ, HARQ with BCH code, HARQ with turbo codes. If the transmitter follows pure ARQ with GBN scheme, then the throughput is given by Eq. 1. Due to error detection bits, this throughput is multiplied by the weight, i.e., the ratio of message symbols to the total number of symbols.

$$\eta_{\mathrm{L}} = \frac{1 - P_e}{1 + S \cdot P_e} \cdot \frac{N - CRC}{N} \tag{1}$$

Here N is the length of the block, P_e is the probability of occurrence of an error in the packet in the pure ARQ method. P_e depends on channel bit error rate, burst errors, and block length. This is expressed in Eq. 2.

$$P_e = 1 - (1 - P_b)^K \tag{2}$$

Where P_b is the Bit error rate. CRC is the number of redundancy symbols added for error detection, and S is defined as the ratio of the time delay of acknowledgment to the time of block transmission

$$S = T_a/T_b \tag{3}$$

T_a is the time delay of acknowledgment, defined as the time delay from terminating block transmission to receiving and processing block acknowledgment, and T_b is the time of block transmission, b is no.of bits per each symbol.

$$b = \log_2(N + 1) \tag{4}$$

If the transmission is in the HARQ scheme, the throughput will be as follows :

$$\eta_H = \frac{K}{N} \times \frac{1 - \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1 - P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1 - P_s)^{N-i}} \tag{5}$$

Where P_s is the probability that the symbol has an error and is calculated as below

$$P_s = 1 - (1 - P_b)^b \tag{6}$$

If the transmitter chooses to use turbo codes with a code rate of 1/3 as the error correction scheme, the throughput is determined by the following calculation.

$$T_{\text{hr}} = \left(\frac{R_c}{T_r}\right) \cdot \left(\frac{k}{k + n_p}\right) \tag{7}$$

Where k/(k + n_p) is the fractional throughput loss because of additional parity bits added for detecting errors, where k is the number of information bits and n_p is the number of parity bits. R_c denotes the code rate (here $R_c = 1/3$), T_r is the average number of transmissions in the HARQ scheme. Where T_r is expressed as

$$T_r = \sum_{i=0}^{\infty} P(D_d)^i = \frac{1}{1 - P(D_d)} \tag{8}$$

Where P(D_d) is the probability that errors are present in the decoded packet.

3.2 Switching scheme

This subsection explains how the D-HARQ model switches between modes and calculation of the thresholds for switching. In this paper, initially, the GBN method is employed. Once the number of transmitted packets reaches the size of the sliding window, a decision is made based on the count of NACK and ACK to switch between states. Switching points are determined by locating the instances where both schemes yield identical throughput at a particular SNR value. The transition is then made so that the scheme with a relatively higher throughput for subsequent SNR values is chosen until another switching point is reached. But the main drawback is under very low SNR values(0-3.47dB), the mentioned schemes produce very negligible throughput (almost 0) as shown in Section 4.

A key enhancement of switching to turbo codes is implemented to overcome this limitation in such scenarios. As discussed in Subsection 1.3, turbo codes include convolutional and block codes in their range of encoding methods, offering high complexity but enhanced accuracy [9]. Turbo codes are utilized only in very low SNR conditions to strike an optimal balance between complexity and accuracy.

When neither a positive nor a NACK is received, it suggests a scenario of extremely low SNR, leading to no throughput, as illustrated in Section 4. The throughput formula further supports this concept, which is inversely proportional to term S. Where S is the ratio of acknowledgment delay to block transmission time, In instances where an acknowledgment for a packet is not received, it implies that the acknowledgment time delay is significantly prolonged, approaching infinity. Since S is the denominator of the throughput equation, a large value for S results in the throughput nearing zero. Consequently, the absence of any acknowledgment can indicate very low SNR, specifically within the range of 0 to 3.47 dB.

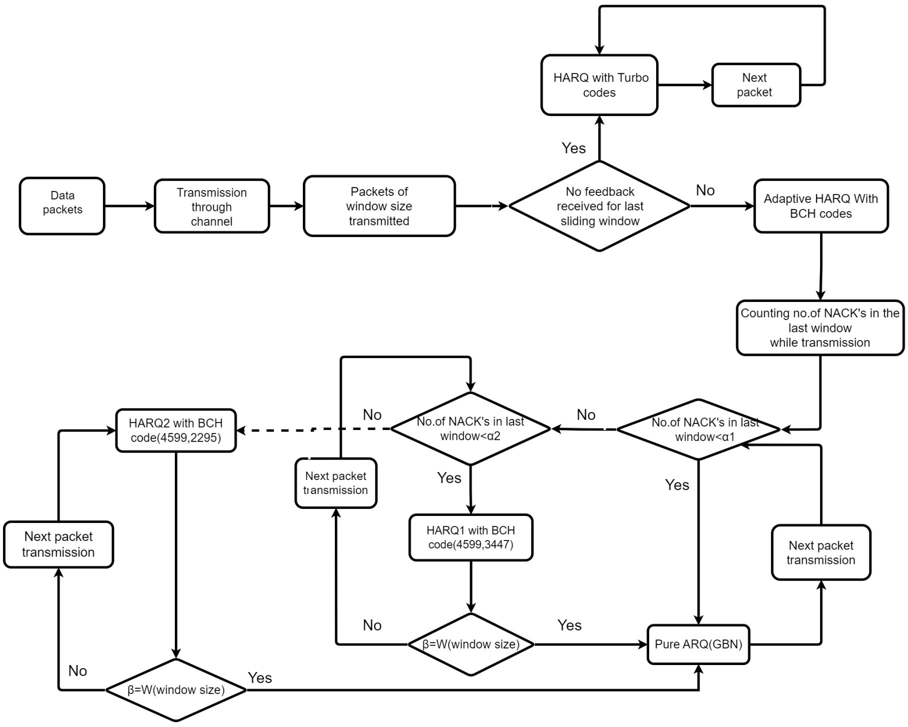

Fig. 7 Illustration of the dynamic switching between different HARQ schemes

The main parameter used to switch between the schemes is the number of NACKs. If the count of NACKs crosses the threshold α_1, the scheme switches from ARQ to HARQ1, as shown in Figure 7. To calculate α_1, first equate the throughput of ARQ with throughput of HARQ1.

$$\frac{1-P_e}{1+S\cdot P_e}\cdot\frac{N\cdot b-CRC}{N\cdot b}=\frac{K}{N}\times\frac{1-\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}}{1+S\cdot\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}} \tag{9}$$

Replace P_e with Eq. 2 and P_s with equation in the Eq. 9.

The unknown variable P_b can be calculated by solving the above equality. By substituting this P_b value in Eq. 2, P_e can be derived and the threshold α_1 is given by

$$\alpha_1 = P_e\cdot W \tag{10}$$

where W denotes window size.

To calculate the threshold α_2 that determines when to switch from HARQ1 to HARQ2, equate the throughput expressions of both HARQ schemes.

$$\frac{K1}{N} \times \frac{1 - \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1-P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1-P_s)^{N-i}} = \frac{K2}{N} \times \frac{1 - \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1-P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1-P_s)^{N-i}}$$

From this equality, the unknown variable P_s can be calculated. Then the threshold α_2 is given by

$$\alpha_2 = P_s \cdot W \tag{11}$$

On the other hand, when there is no error detected in the last sliding window, a new kind of acknowledgment ACK+ is sent to the transmitter. Then the scheme switches back from HARQ1 or HARQ2 to ARQ as the signal quality is exceptionally good, as depicted in Figure 7. When channel conditions are favorable, this transition back to pure ARQ significantly reduces unnecessary costs. This threshold for the number of acknowledgments is denoted by β. Where β=W indicates that W acknowledgments are sent consecutively. The detailed flow of the switching scheme is depicted in Figure 7. In the observed scenario, adaptive HARQ with BCH codes shows negligible throughput at SNR conditions ranging from 0 to 3.47 dB. However, when HARQ with turbo codes is employed, a substantial throughput improvement at very low SNR values can be observed from Figure 11; this suggests that turbo codes could effectively enhance throughput in this specific range.

3.3 Detailed mechanism of HARQ with Turbo Codes

This subsection explains the mechanism HARQ with turbo codes in D-HARQ model. Initially, the packet undergoes a series of encoding phases to enhance error correction capabilities. As shown in Figure 8, the first phase, known as Bit Splitting, prepares the input bitstream for parallel processing. This is followed by Constituent Encoding 1, where the first half of the bitstream is convolutionally encoded, introducing redundancy that serves as the foundation for error correction. To further improve the ability to correct errors, the bitstream undergoes Interleaving, ensuring a more uniform distribution of potential errors across the bitstream. The process continues with Constituent Encoding 2, adding an additional layer of redundancy through a second convolutional encoder. Finally, Rate Matching is applied to adjust the encoded bitstream according to the transmission channel's specific bandwidth and error characteristics. Upon completing the encoding stages, the data is ready for modulation onto a carrier signal and transmitted across the communication channel, Where it is prone to noise and interference. At the receiving end, the Reception and Signal Demodulation stage takes over, where the transmitted signal is captured and demodulated to retrieve the encoded data bitstream.

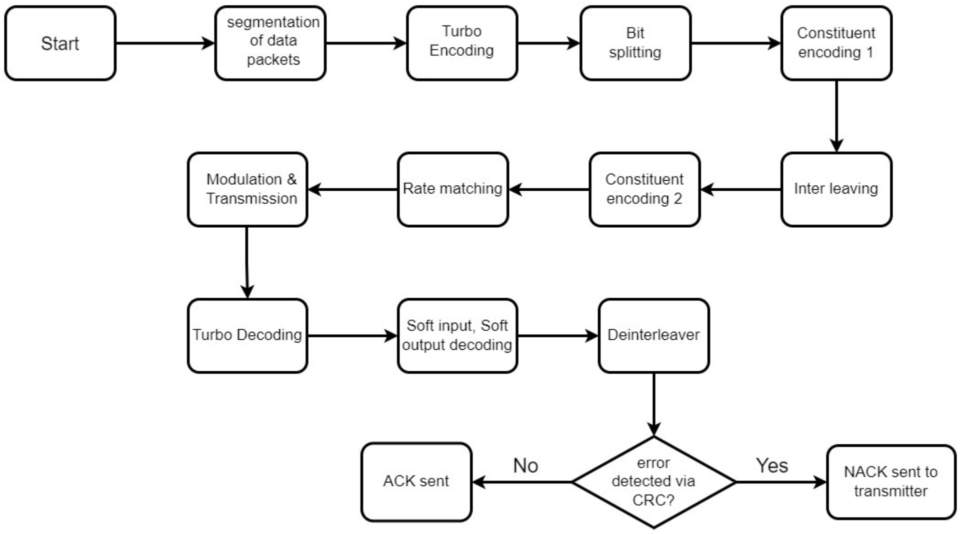

Fig. 8 HARQ with Turbo codes

The turbo decoding process begins with Soft-input Soft-output Decoding Iteration 1 as depicted in Figure 8. This phase marks the initial attempt to decode the received signal, utilizing soft decision inputs to produce soft decision outputs. These outputs include probability metrics that indicate the accuracy of each bit, significantly enhancing the decoding process's effectiveness. De-interleaver re-orders bits between decoding iterations. Subsequent iterations of decoding are required to refine the accuracy of the data. After decoding, the error detection is conducted through Cyclic Redundancy Check (CRC), prompting a NACK for retransmission if errors are detected or an Acknowledgement (ACK) for error-free data, signaling successful reception.

3.4 SNR estimation algorithm

This subsection provides a detailed explanation of the SNR estimation algorithm, which is a crucial requirement for the selective soft combining process. The Error Vector Magnitude (EVM) SNR estimation Algorithm 1 has been employed to enhance the accuracy and the scope of SNR estimation. This advancement comes in response to the limitations of current methodologies, which struggle with accurately estimating low SNRs. Initially, the mean and variance of the received signal samples $|y_n|$ at time instants n = 1, 2, ..., L after downsampling is calculated as below in Eq. 12, Eq. 13.

$$\text{mean} = \frac{1}{L}\sum_{n=1}^{L}|y_n| \tag{12}$$

$$\text{var} = \frac{1}{L}\sum_{n=1}^{L}(|y_n| - \text{mean})^2 \tag{13}$$

SNR is calculated from Eq. 14 using these results.

$$SNR = 10 \times \log\left(\frac{|\text{mean}|^2}{2 \times \text{var}}\right) \tag{14}$$

When the estimated value is less than 10dB, the intermediate z value temporarily stores this SNR.

$$z = SNR \tag{15}$$

And the new SNR' is estimated by using the z value.

$$\text{SNR'} \ = \sqrt{(z - 2.5) \times 39.2} - 7 \tag{16}$$

When the SNR is below 10 dB, SNR' provides an accurate estimate; otherwise, the SNR is considered acceptable. This algorithm has higher estimation accuracy and less deviation in a larger range between -10 to 30 dB. This algorithm's complexity is low as it uses only the mean and variance of data.

Algorithm 1 SNR Estimation Algorithm [15]

Input: Received signal samples y_n, number of samples L
Output: Estimated SNR
Compute mean and variance:
  mean $\leftarrow \frac{1}{L}\sum_{n=1}^{L}|y_n|$
  var $\leftarrow \frac{1}{L}\sum_{n=1}^{L}(|y_n| - \text{mean})^2$
Compute SNR:
  SNR $\leftarrow 10 \times \log_{10}\left(\frac{|\text{mean}|^2}{2\times\text{var}}\right)$
if SNR $<$ 10 dB **then**
  Compute intermediate value:
    $z \leftarrow$ SNR
  Compute new SNR':
    SNR' $\leftarrow \sqrt{(z - 2.5) \times 39.2} - 7$
  return SNR'
else
  return SNR
end if

3.5 Decision Making for Soft Combining at Receiver End

This subsection explains how the receiver decides whether to perform soft combining or not based on the current channel conditions. At the receiver end, the choice of soft combining is made by estimating the SNR value of the received signal using the EVM algorithm as discussed in Section 3.4. This algorithm is advantageous as it can estimate accurate SNR values even at very low SNR conditions. Since calculating SNR for every signal can be complex at the receiver end, calculating at regular intervals reduces the overhead.

Soft combining is a technique that involves combining the re-transmitted data with previously received erroneous blocks to enhance decoding accuracy. But when the channel conditions are exceptionally good, the benefit of soft combining might be limited. It can be observed in Section 4. This is because the individual received signals are already of high quality. Therefore, it might be more efficient to avoid soft combining to save computational resources and reduce complexity. Thus, soft combining should be done selectively by estimating the channel conditions through SNR.

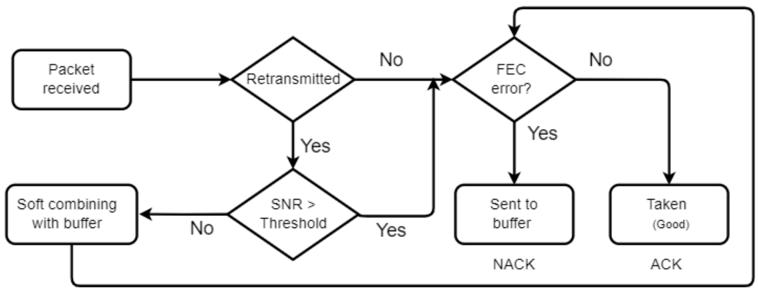

Fig. 9 Illustration of the decision-making for soft combining at the receiver end

Upon receiving a packet, as depicted in Figure 9, the first step is to determine whether it is an original or retransmission transmission. If the packet is not retransmitted, it is directly sent to the error detection module. If no errors are detected, it is accepted directly. However, if errors are identified, it is forwarded to the buffer, and NACK is sent to the transmitter requesting retransmission of the packet.

On the other hand, if the packet is identified as retransmitted, a decision must be made on whether to apply soft combining. This decision is based on the estimated SNR calculated from the EVM algorithm. If the SNR exceeds the predetermined threshold, it signifies a strong signal quality, suggesting that the packet is likely error-free. In such scenarios, merging this packet with previously errored packets in the buffer could degrade performance. Consequently, the packet is directly sent to the error detection module without undergoing soft combining, as shown in Figure 9. Conversely, soft combining is employed if the SNR does not meet the threshold, indicating a weaker signal. This strategy enhances throughput and decreases the required retransmissions by utilizing the cumulative information from multiple receptions of the same packet.

4 Results and Anlaysis

Simulations of D-HARQ were conducted within the Land Mobile and Satellite (LMS) channel using MATLAB and python, focusing on rural and suburban environments. LMS channel facilitates communication between land-based mobile stations and satellites, enabling wide geographic coverage for services such as GPS, satellite phones, and remote sensing in remote or rural areas. Under these conditions, the energy loss in the Line of Sight (LOS) wave is comparatively insignificant. LOS is an unobstructed path between transmitter and receiver, offering reliable communication with minimal signal attenuation, crucial for applications like microwave links and satellite communication. Consequently, signal propagation in this context follows a lognormal distribution with the probability density function (PDF), as detailed below

$$f(x) = \frac{e^{-\left(\frac{(\ln x)^2}{2\sigma^2}\right)}}{x\sigma\sqrt{2\pi}} \tag{17}$$

For x > 0, Where x is the random variable representing the signal strength and σ is the distribution of additive white Gaussian (AWGN) noise.

Table 1 details the number of information bits and the associated redundancy bits, such as CRC for ARQ and error-correcting bits for HARQ1 and HARQ2. This information illustrates the error correction capabilities of each scheme in communication systems. Additionally, Table 2 illustrates the simulation environment.

Table 1 Number of information and redundancy bits for various schemes

Scheme	Information bits	Redundancy bits
ARQ	4567	32 (CRC)
HARQ1 with BCH codes	3447	1152
HARQ2 with BCH codes	2295	2304

Table 2 Simulation environment and parameters

Parameters	Values
Channel model	Land, Mobile, and Satellite
Channel conditions	Rural and Suburban
ARQ Scheme	ARQ, HARQ1, HARQ2
No. of symbols in each block/packet	511
No. of bits per symbol	9
Delay	5 $\times$ No. of information symbols
Noise	White Gaussian (AWGN) noise
Simulation output	Matlab with Python codes
Simulated SNR range	0-18 dB

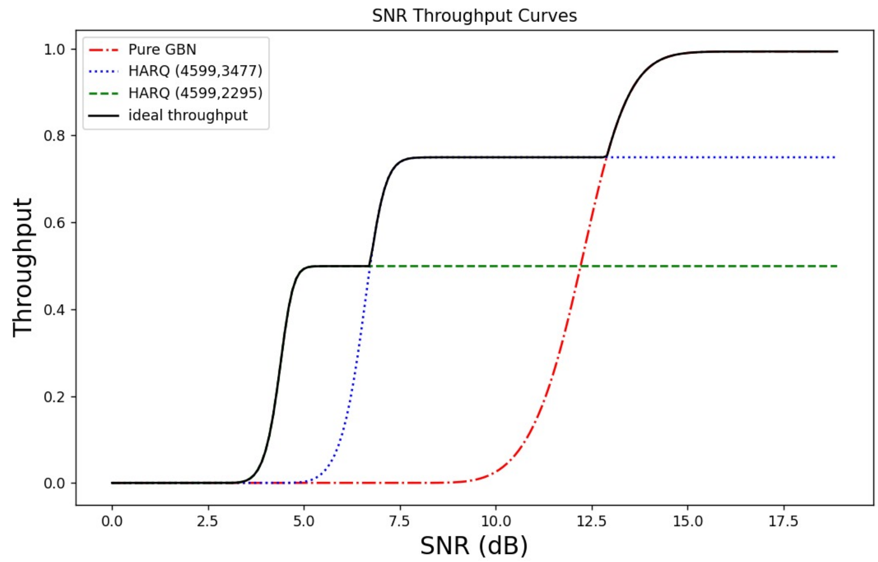

Fig. 10 Transitions between various HARQ and ARQ schemes

Figure 10 illustrates a graph of an D-HARQ model, highlighting the thresholds for transitioning between different modes, which have been established via a trial-and-error methodology. The simulations are done for various sets of thresholds. Ultimately, the graph is almost aligned with the ideal curve for $\alpha 1 = 10$, $\alpha 2 = \beta =$ W$= 64$. Where the throughput of the ideal curve at a given SNR is expressed as

$$\eta_{\text{ideal}} = \max(\eta_{\text{ARQ}}, \eta_{\text{HARQ1}}, \eta_{\text{HARQ2}}) \tag{18}$$

Figure 10 demonstrates the performance benefits of implementing an adaptive HARQ strategy within the SNR range of 3.47 dB to 9 dB, compared to a traditional GBN approach. Adopting the D-HARQ method in this specific SNR interval substantially increases throughput, achieving a gain of approximately 2.16×10^{16} times. Moreover, Figure 10 highlights the enhanced throughput achieved by utilizing two different states of HARQ, each configured with BCH codes of varying code rates. A notable throughput enhancement of 31.67% is observed when transitioning from HARQ state 2 to HARQ state 1 within the SNR range of 6.68 to 12.89 dB. The D-HARQ model recommends transitioning from HARQ back to ARQ at higher SNR values, a strategy convincingly supported by the simulation curve. This curve demonstrates the necessity for such a switch, highlighting a significant improvement of about 28.5% in throughput above 12.89 dB.

The main drawback observed from Fig 10 is negligible throughput nearing zero in the 0 to 3.47dB range. This D-HARQ model successfully avoids this by adapting HARQ with turbo codes in this range.

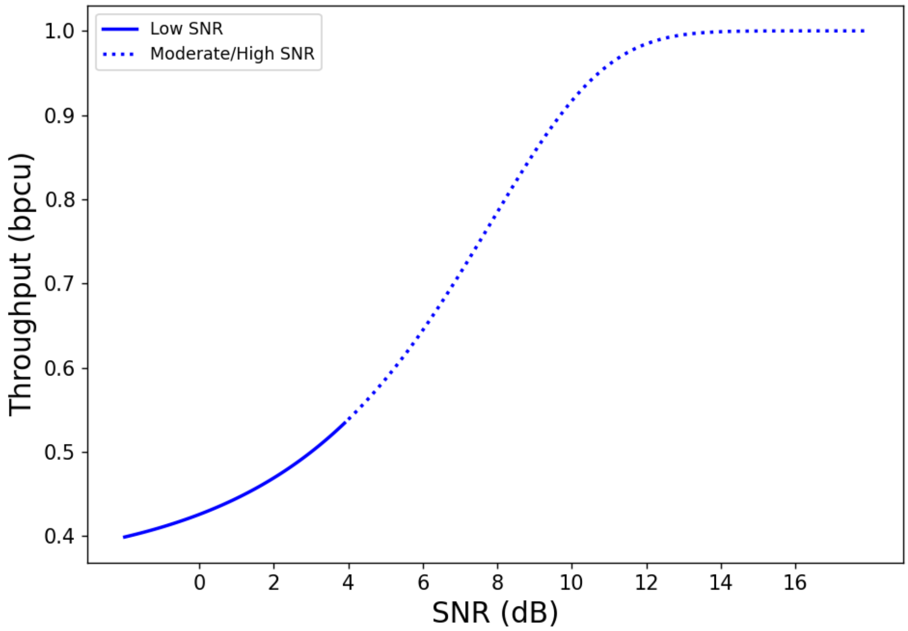

Fig. 11 Estimated Throughput vs SNR for HARQ with Turbo codes (code rate = 1/3)

Figure 11 shows considerable throughput even at low SNR values, Thus an improvement of throughput from 0 to 0.467 bpcu by adapting turbo codes. From Figure 11, one might question the feasibility of utilizing turbo codes across all SNR ranges. Implementing Turbo Codes, while offering superior error correction capabilities, particularly in environments with a high SNR, presents a more complex challenge than BCH codes due to several key factors. First and foremost, the architecture of Turbo Codes involves an intricate iterative decoding process, which relies on passing information back and forth between two or more constituent decoders. This interplay requires a sophisticated algorithm to effectively converge on the correct decoding, significantly complicating the implementation compared to the straightforward algebraic decoding methods used for BCH codes.

As explained in Section 2, Soft combining is beneficial as combining the retransmitted packet with the error packets in the buffer improves its decoding capability, thus reducing the need for retransmissions. But at very high SNR the idea of soft combining doesn't contribute much towards the throughput improvement. Thus, under these scenarios, there is no need to adapt such schemes, thus optimizing the complexity and minimizing the usage of resources.

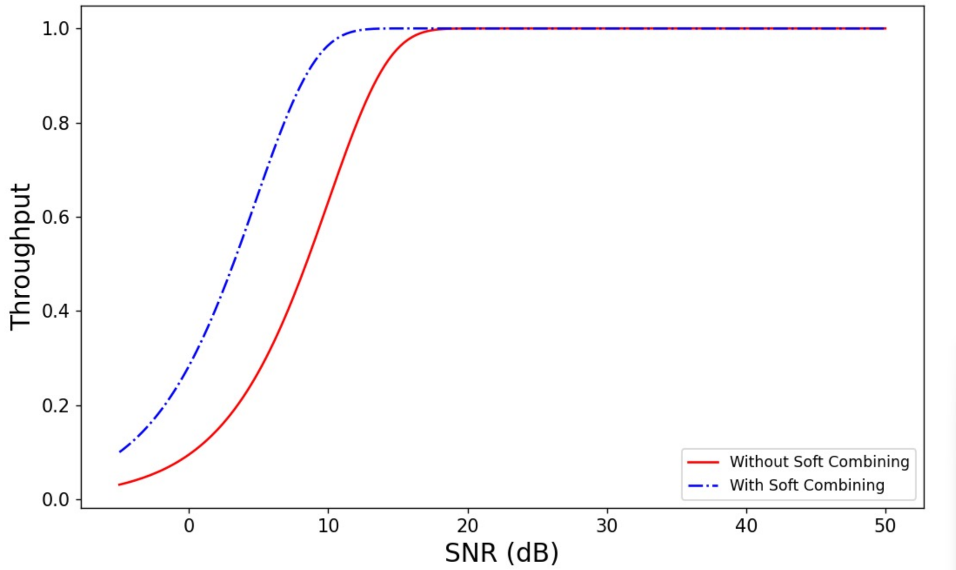

Fig. 12 Performance comparison with and without Soft combing

Figure 12 compares throughput with and without soft combining at all SNR values ranging from 0 to 50 dB. Thus the threshold to selectively soft combine is 17.46 dB, therefore SNR above this threshold indicates that the channel quality is good enough to omit soft combining.

5 Conclusion and Future works

This paper introduced D-HARQ; an SNR responsive scheme that switches between ARQ and HARQ using BCH and turbo coding. Additionally, it is integrated with the selective soft combining method for improved performance. The strategic utilization of turbo codes under very low SNR conditions and BCH codes in other scenarios strikes an optimal balance between cost-effectiveness and reliability. As soft combining is selectively avoided under good channel conditions, the associated complexity overhead is minimized, improving performance and maximizing throughput. The effectiveness of this paper's proposal is validated through simulation results using turbo coding and selective soft combining. These results demonstrate a significant improvement in throughput while reducing complexity at every level of data transmission. Future works could involve initiating simulations and leveraging Machine Learning for accurate threshold determination.

References

[1] K. Kotuliaková, J. Polec, F. Csóka, An adaptive arq-harq method with bch codes, in: 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), IEEE, 2017, pp. 1–5.

[2] R. Kadel, K. Paudel, D. B. Guruge, S. J. Halder, Opportunities and challenges for error control schemes for wireless sensor networks: A review, Electronics 9 (3) (2020) 504.

[3] Q. Li, C.-X. Chen, A hybrid arq protocol for the communication system with multiple channels, in: 2018 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2018, pp. 222–227.

[4] F. A. Nada, Performance analysis of go-back-n arq protocol used in data transmission over noisy channels, Adv. Sci. Technol. Eng. Syst. J 5 (4) (2020) 612–617.

[5] Y.-D. Yao, An effective go-back-n arq scheme for variable-error-rate channels, IEEE Transactions on Communications 43 (1) (1995) 20–23.

[6] A. Ahmed, A. Al-Dweik, Y. Iraqi, H. Mukhtar, M. Naeem, E. Hossain, Hybrid automatic repeat request (harq) in wireless communications systems and standards: A contemporary survey, IEEE Communications Surveys & Tutorials 23 (4) (2021) 2711–2752.

[7] A. Al-Dweik, Y. Iraqi, H. Mukhtar, M. Naeem, E. Hossain, et al., Hybrid automatic repeat request (harq) in wireless communications systems and standards: A contemporary survey, Authorea Preprints (2023).

[8] X.-G. Xia, Understanding turbo codes: A signal processing study, Journal of Information and Intelligence 2 (1) (2024) 1–13.

[9] C. Berrou, R. Pyndiah, P. Adde, C. Douillard, R. Le Bidan, An overview of turbo codes and their applications, in: The European Conference on Wireless Technology, 2005., IEEE, 2005, pp. 1–9.

[10] M. El-Khamy, J. Lee, I. Kang, Soft turbo harq combining, in: 2013 IEEE International Conference on Communications (ICC), IEEE, 2013, pp. 5542–5547.

[11] X.-G. Xia, Understanding turbo codes: A signal processing study, Journal of Information and Intelligence 2 (1) (2024) 1–13.

[12] A. Sastry, Improving automatic repeat-request (arq) performance on satellite channels under high error rate conditions, IEEE Transactions on Communications 23 (4) (1975) 436–439.

[13] P. Fidler, J. Poctavek, K. Kotuliakova, J. Polec, New adaptive arq/harq scheme using rs coding and its throughput analysis, in: IEEE EUROCON 2009, IEEE, 2009, pp. 1662–1667.

[14] M. Martinovič, J. Polec, K. Kotuliaková, An adaptive arq–harq method with two rs codes, International Journal of Electronics and Communication Engineering 7 (4) (2013) 385–389.

[15] X. Qun, Z. Jian, Improved snr estimation algorithm, in: 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), IEEE, 2017, pp. 1458–1461.