

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

C. Gnana Jyothi<sup>1</sup>, P. Chandana Sai Sri Hasitha<sup>1</sup>,  
N. Yaswanth<sup>1</sup>, B. R. Chandavarkar<sup>1\*</sup>

<sup>1\*</sup>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](mailto:brc@nitk.edu.in);  
Contributing authors: [chinthagnanajyothi.221cs118@nitk.edu.in](mailto:chinthagnanajyothi.221cs118@nitk.edu.in);  
[prathapachandanasaisrihasitha.221cs139@nitk.edu.in](mailto:prathapachandanasaisrihasitha.221cs139@nitk.edu.in);  
[namburiyaswanth.221cs232@nitk.edu.in](mailto: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. Standard Automatic Repeat Request (ARQ) methods suffer from high communication overhead and latency and are inefficient in handling retransmissions. Although traditional HARQ is effective, its high implementation costs limit its widespread use. 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 primary objective is to maximize throughput even at low SNR values while ensuring optimal complexity.

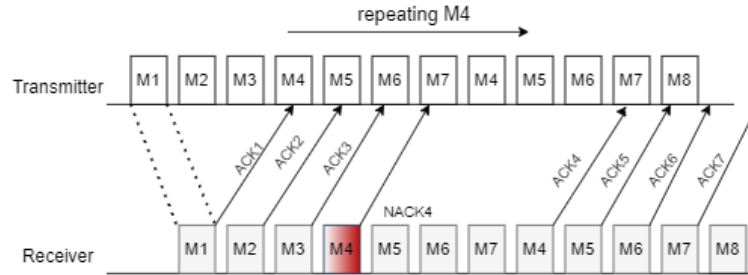
**Keywords:** Adaptive HARQ, SNR estimation, Turbo codes, BCH codes, Selective soft combining, Error correction, Throughput optimization

# 1 Introduction

Error control mechanisms are essential for ensuring reliable data transmission. They help detect errors and trigger retransmissions when packets are lost or corrupted. When data travel from one application to another, it goes through multiple steps and errors can occur at any stage, making error control a crucial part of maintaining communication quality[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. This paper develops a novel approach to address this issue by adjusting 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 transmits a sequence of frames without waiting for acknowledgment until it reaches a specified window size. If errors are detected or the packet is lost, a negative acknowledgment (NACK) [5] is sent, and the sender resends all frames starting from the last acknowledged frame. Figure 1 [5] 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 on the receiver side. GBN guarantees data integrity and reduces network congestion.



**Fig. 1** Go Back-N (GBN) scheme

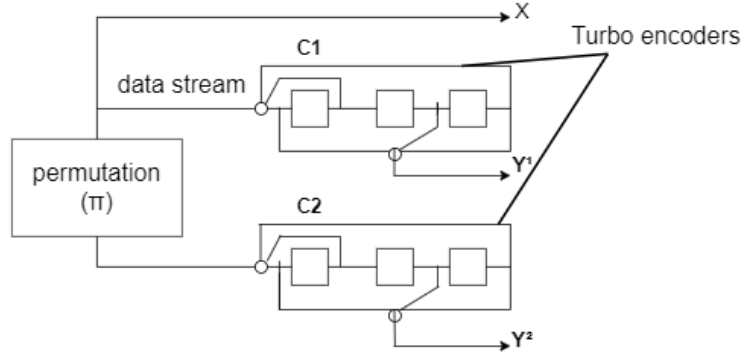
## 1.2 Hybrid ARQ

Hybrid ARQ [6, 7] combines ARQ with FEC, enabling error correction during transmission and reducing the likelihood of additional errors. If packet errors remain uncorrected, the system initiates retransmission. HARQ mainly has two basic schemes:

firstly, Hybrid ARQ Type I, which has adaptive coding rates and additional error-correcting codes to minimize retransmissions while maintaining 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).



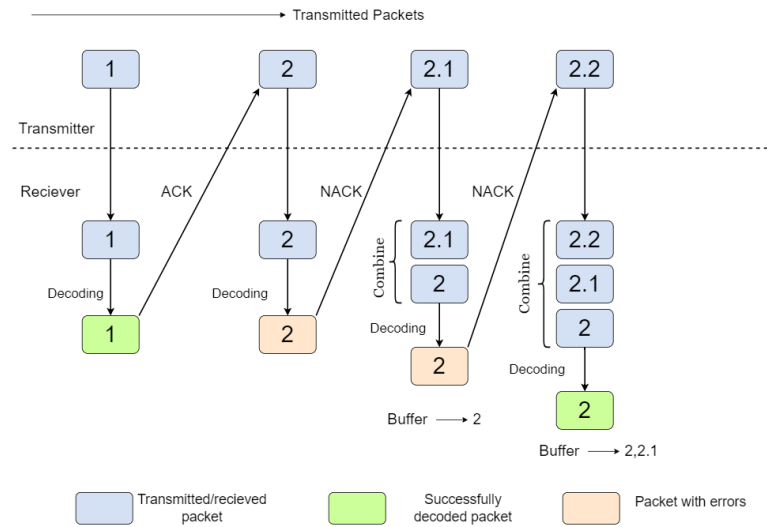
**Fig. 2** A classical Turbo code

Figure 2 [9] depicts a classical Turbo Code structure, which consists of the parallel concatenation of two binary Convolutional encoders, C1 and C2, separated by a permutation block labeled  $\pi$ . 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 ( $\pi$ ), 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 stores a received erroneous packet in a buffer rather than discarding it. Two independently uncorrectable erroneous packets may be decoded correctly using 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 with the error bits of previous transmissions adds extra energy for each retransmission by increasing the  $E_b/N_0$  ratio. In incremental redundancy, the encoder generates distinct redundancy bits by puncturing its output, adding new information to each retransmission.



**Fig. 3** Illustration of Soft combining with HARQ

Figure 3 [11] illustrates soft combining with HARQ in a communication system. The transmitter sends packets, and the receiver decodes them. If it correctly decodes a packet, it returns an ACK to the transmitter. If errors occur in the received packet, the receiver sends a NACK and stores it in a buffer. The transmitter then resends the new copy of the same packet, and the receiver combines (soft combining) the previously stored erroneous packet with the newly received packet to improve the decoding process. The receiver repeats the combining process for subsequent NACKs and retransmissions, storing the combined packets in the buffer (e.g., 2, 2.1, 2.2.1). The soft combining techniques enhance 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, FEC [2] should be chosen based on the channel conditions. Many existing approaches involve repeated retransmissions because they discard corrupted packets rather than utilizing them. These schemes are significant in DVB (Digital Video Broadcasting) 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 result in significant throughput improvements across all scenarios.

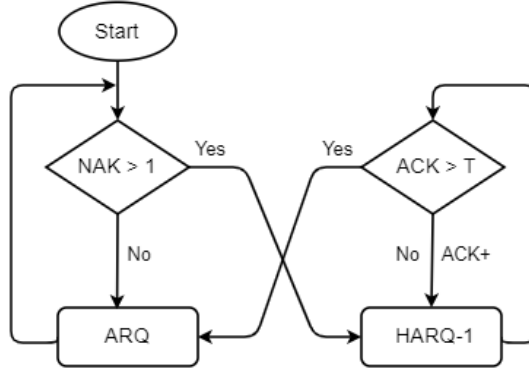
The paper is structured as follows: Section 2 covers the literature Survey, while Section 3 discusses the design and implementation of the proposed approach. Section 4 presents simulation results, analysis, and comparisons with existing methods. Finally, Section 5 concludes with future directions and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

## 2 Literature survey

Subsections 2.1 to 2.7 outline various techniques, along with their strengths and limitations, from different papers focused on enhancing data transmission efficiency and reliability using HARQ methods. Subsection 2.1 discusses Adaptive HARQ with Reed-Solomon (RS) Codes. Subsection 2.2 covers Adaptive HARQ with Two RS Codes. Subsection 2.3 focuses on Adaptive HARQ with BCH Codes. Subsection 2.4 highlights Non-orthogonal HARQ for URLLC. Subsection 2.5 is about Energy-efficient optimization over fading channels. Subsection 2.6 covers Adaptive HARQ with reinforcement learning technique. Finally, Subsection 2.7 discusses data recovery loops in 6G networks.

### 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 predicted channel state. As depicted in Figure 4 [13], 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. If a NACK is detected, the system immediately switches back to HARQ mode to improve error correction as the channel degrades.



**Fig. 4** Adaptive HARQ using RS codes

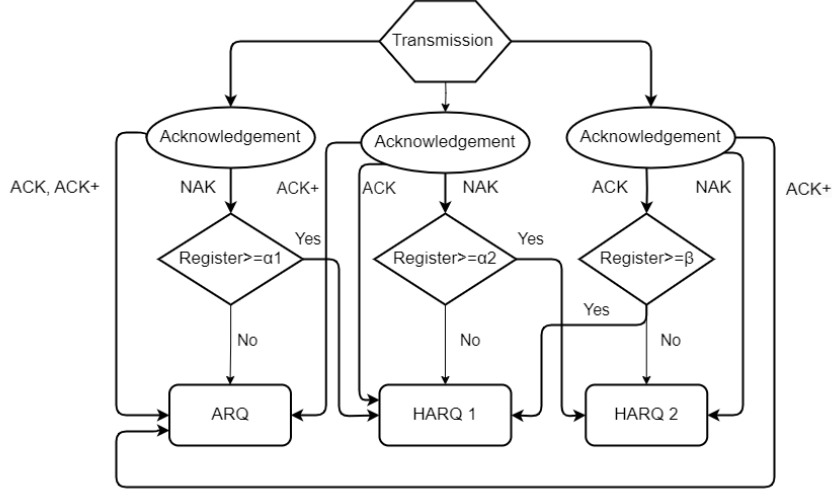
This approach has several benefits. First, its adaptive nature allows a quick switch to HARQ mode when a noisy channel is detected. This dynamic approach requires a threshold of positive acknowledgments to switch back to ARQ mode, ensuring that the system remains robust and responsive to real-time channel conditions. On the other hand, this schema has a few limitations, such as limited adaptability with only two states, which restricts the system adjustment to high fluctuations in channel quality and 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. Increasing the number of states compared to the adaptive HARQ scheme discussed in 2.1 allows for 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 with a low channel error rate. The high channel error rate is further divided into HARQ1 and HARQ2. Both states use RS codes with different code rates.

Figure 5 [14] illustrates the switching logic, where the transmitter controls the transition from ARQ to HARQ modes or between HARQ1 and HARQ2. Meanwhile, the receiver manages the switch from HARQ modes back to ARQ. In the proposed approach [14], these three confirmations are utilized as follows:

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



**Fig. 5** Dynamic Adaption of different HARQ schemes using two RS codes

This approach brings 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 optimize throughput for a broader range of SNR values -also, 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 that uses BCH codes 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. This paper proposes an adaptive HARQ model that dynamically switches between HARQ1 and HARQ2 schemes, which have different code rates based on the estimated channel state. If the error rate is low, the GBN scheme is used, which is more efficient for good channel conditions. Otherwise, it evaluates the channel condition based on the number of NACKs and a predefined threshold. If the condition indicates significant transmission errors, the system follows the HARQ2 scheme with a higher code rate, enhancing reliability in challenging environments. The HARQ1 scheme with a medium code rate is used if the error level remains within an acceptable range.

These hybrid schemes combine two BCH codes selected to improve error correction performance in various channel conditions. The GBN [12] protocol is used for retransmissions, ensuring better performance, as discussed in Subsection 1.1. This scheme has various advantages, including dynamically altering parameters to adapt to changing network conditions and effectively controlling retransmissions, significantly lowering

latency and improving responsiveness. The disadvantage of this schema is the difficulty of maintaining effective data transmission under low SNR conditions. In such conditions, there is almost zero throughput.

## 2.4 Non-orthogonal HARQ for URLLC

Faisal Nadeem et al. [15] proposed the non-orthogonal HARQ for URLLC, a scheme tailored for ultra-reliable and low-latency communication systems. It introduces non-orthogonal retransmissions, allowing multiple packets to share time slots during retransmission, significantly reducing latency. This scheme is particularly successful in high-speed applications, such as industrial automation or driverless cars. While it achieves impressive reliability and delay reductions, its reliance on non-orthogonal resource sharing can lead to resource allocation complexities in highly dynamic network environments. Additionally, its application is primarily limited to latency-critical scenarios.

## 2.5 Energy-efficient optimization for HARQ schemes over time-correlated fading channels

Zheng Shi et al. [16] proposed an energy-efficient optimization for HARQ schemes in the context of time-correlated fading channels, addressing the critical trade-off between energy consumption and communication reliability. The study improves energy efficiency by optimizing transmission powers and rates while maintaining acceptable performance in terms of error rates. The closed-form solutions derived for HARQ schemes, such as Type I, Chase Combining (CC), and Incremental Redundancy (IR), demonstrate a practical approach to reducing energy usage. However, the model assumes relatively stable time-correlated fading channels, which might not generalize well to rapidly changing SNR conditions. Moreover, the focus on energy efficiency sometimes comes at the cost of reduced throughput in highly challenging channel conditions.

## 2.6 Adaptive HARQ with reinforcement learning

Chia-Hung Yeh et al. [17] proposed the adaptive HARQ with reinforcement learning that employs machine learning to dynamically select optimal transmission strategies based on channel quality indicators (CQI). The approach uses reinforcement learning algorithms, to adapt to real-time changes in channel conditions by exploring an extensive action space, including varying retransmission schemes. This adaptability reduces block error rates and lower transmission delays compared to static HARQ mechanisms. Despite its strengths, this scheme introduces significant computational overhead, making it less feasible for real-time applications with resource-constrained devices. Additionally, the training phase for reinforcement learning models requires extensive data and computational resources, which might not always be practical.



## 2.7 Data recovery loops in 6G networks

Uyoata E. Uyoata et al. [18] presented the data recovery loops in a 6G networks paper that focuses on the interplay between HARQ schemes in the physical and medium access control layers and higher-layer ARQ mechanisms. This study emphasizes the need for enhanced coordination between layers to achieve efficient data recovery in 6G networks. The proposed enhancements, such as fast HARQ schemes for challenging radio conditions, show substantial throughput gains, especially for users at cell edges or with poor signal quality. However, the study narrows its scope to 6G scenarios, offering limited insights into how these mechanisms perform in legacy or non-6G network environments. The high complexity of integrating these multi-layer recovery loops also raises concerns about scalability and implementation in real-world systems.

**Table 1:** Comparison of different HARQ techniques

Technique	Description	Advantages	Drawbacks
Adaptive HARQ with Reed-Solomon (RS) Codes [13]	Dynamically switches between HARQ and ARQ modes based on ACKs and NACKs.	<ul style="list-style-type: none"> <li>• Robust and straightforward to channel fluctuations.</li> <li>• Adapts quickly to noisy channels.</li> </ul>	<ul style="list-style-type: none"> <li>• Limited adaptability with only two states.</li> <li>• Poor scalability for complex networks.</li> </ul>
Adaptive HARQ with Two RS Codes [14]	Employs a three-stage ARQ/HARQ1/HARQ2 scheme with different RS code rates.	<ul style="list-style-type: none"> <li>• Flexible and adaptable to diverse channel conditions.</li> <li>• Higher throughput for various error rates.</li> </ul>	<ul style="list-style-type: none"> <li>• Increased latency for real-time traffic.</li> <li>• Limited error correction for random errors due to RS code design.</li> </ul>
Adaptive HARQ with BCH Codes [1]	Dynamically switches between HARQ1 and HARQ2 schemes with varying code rates.	<ul style="list-style-type: none"> <li>• Adapts to moderate to high error rates effectively.</li> <li>• Reduces latency by dynamically altering retransmission parameters.</li> </ul>	<ul style="list-style-type: none"> <li>• Ineffective under low SNR conditions.</li> <li>• Throughput approaches zero in poor signal environments.</li> </ul>
Non-orthogonal HARQ for URLLC [15]	Introduces non-orthogonal retransmissions to reduce latency, enabling multiple packets to share time slots during retransmission.	<ul style="list-style-type: none"> <li>• Reduces latency.</li> <li>• Effective in high-speed applications (e.g., autonomous vehicles, industrial automation).</li> </ul>	<ul style="list-style-type: none"> <li>• Resource allocation complexities in dynamic networks.</li> <li>• Limited versatility for throughput optimization.</li> </ul>

Technique	Description	Advantages	Drawbacks
Energy-efficient optimization over fading channels [16]	Optimizes transmission powers and rates for energy efficiency while maintaining reliability in time-correlated fading channels.	<ul style="list-style-type: none"> <li>Enhances energy efficiency.</li> <li>Provides practical closed-form solutions for various HARQ schemes (e.g., Type I, CC, IR).</li> </ul>	<ul style="list-style-type: none"> <li>Assumes stable fading conditions.</li> <li>May reduce throughput in challenging channel environments.</li> </ul>
Adaptive HARQ with reinforcement learning [17]	It uses machine learning to select optimal retransmission strategies based on channel quality indicators dynamically.	<ul style="list-style-type: none"> <li>Adapts to real-time channel changes.</li> <li>Reduces block error rates and delays compared to static HARQ mechanisms.</li> </ul>	<ul style="list-style-type: none"> <li>High computational overhead.</li> <li>Requires extensive data and resources for model training.</li> </ul>
Data recovery loops in 6G networks [18]	Enhances coordination between physical, MAC, and higher-layer ARQ mechanisms to improve data recovery in 6G networks.	<ul style="list-style-type: none"> <li>Substantial throughput gains for edge users or poor signal quality.</li> <li>Effective in challenging radio conditions.</li> </ul>	<ul style="list-style-type: none"> <li>Limited applicability to non-6G systems.</li> <li>High complexity and scalability challenges in real-world implementations.</li> </ul>

However, most solutions involve trade-offs between complexity, generalizability, and resource constraints, as mentioned in Table 1, which should be carefully considered based on the intended application. The proposed D-HARQ perfectly balances complexity and throughput as it adaptively switches between schemes according to channel conditions, requiring no heavy computations.

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

This section of the paper discusses the design and implementation of the D-HARQ scheme to improve communication efficiency. Firstly, Subsection 3.4 provides insights into throughput analysis for schemes, including pure ARQ, HARQ with BCH, and turbo codes, explaining their performance under varying channel conditions. Secondly, Subsection 3.1 outlines a dynamic switching scheme between HARQ modes and pure ARQ based on NACK/ACK counts and SNR thresholds. Thirdly, Subsection 3.2 delves into the mechanism of HARQ with Turbo Codes, emphasizing error correction and throughput improvement in low SNR scenarios. Fourthly, in Subsection 3.3, the SNR estimation algorithm based on Error Vector Magnitude (EVM) analysis is discussed,

which is crucial for a selective soft combining scheme. Lastly, Subsection 3.5 discusses decision-making for soft combining at the receiver 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 with HARQ for error correction. 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 reduce unnecessary complexity. 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

The first stage of implementation is a calculation of the throughput of different schemes like Pure ARQ, HARQ with BCH code, and HARQ with turbo codes. If the transmitter follows pure ARQ with the GBN scheme, then the throughput is given by Eq. 1 [14]. Error detection bits multiply this throughput by the weight, i.e., the ratio of message symbols to the total number of symbols.

$$\eta_L = \frac{1 - P_e}{1 + S \cdot P_e} \cdot \frac{N - CRC}{N} \quad (1)$$

Here,  $N$  is the length of the block,  $CRC$  represents the number of redundancy symbols added for error detection,  $P_e$  is the probability of occurrence of an error in the packet in the pure ARQ method.  $P_e$  depends on the channel bit error rate, burst errors, and block length, as expressed in Eq. 2 [1].

$$P_e = 1 - (1 - P_b)^K \quad (2)$$

Here,  $P_b$  is the bit error rate,  $K$  represents the number of information symbols in the packet, and  $S$  is defined as the ratio of the acknowledgment time delay to the block transmission time.

$$S = T_a / T_b \quad (3)$$

$T_a$  is the time delay of acknowledgment, defined as the delay from terminating block transmission to receiving and processing block acknowledgment.  $T_b$  is the time of block transmission [19].

If the transmission is in the HARQ scheme, the throughput [19] will be as follows :

$$\eta_H = \frac{K}{N} \times \frac{1 - \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}} \quad (4)$$

Here,  $t$  represents the count of repairable symbols ( $2t$  = redundancy bits), and  $P_s$  is the probability that the symbol has an error, calculated as follows.

$$P_s = 1 - (1 - P_b)^b \quad (5)$$

Here,  $b$  is no. of bits per symbol given by

$$b = \log_2(N + 1) \quad (6)$$

If the scheme uses turbo codes with a code rate of 1/3 for error correction, then the throughput is calculated as follows:

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

Where  $k/(k + n_p)$  is the fractional throughput loss because of additional parity bits added for detecting errors, where  $k$  represents the number of information bits, while  $n_p$  denotes the number of parity bits,  $R_c$  denotes the code rate (here,  $R_c = 1/3$ ).  $T_r$  represents the average number of transmissions in the HARQ scheme and is expressed in [20].

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

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

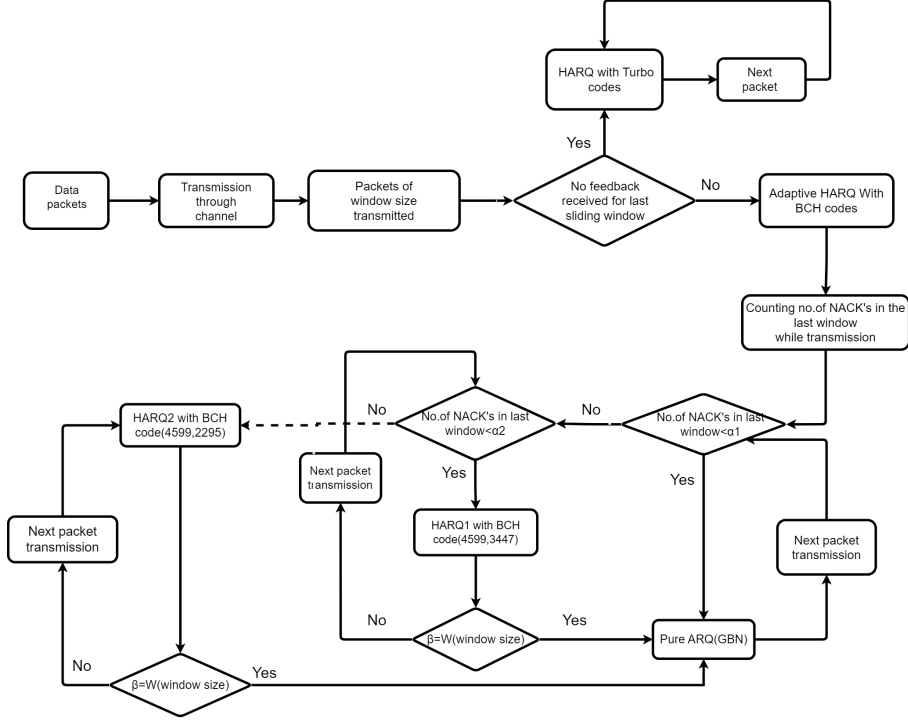
### 3.2 Switching Scheme

This subsection explains the dynamic switching mechanism of the D-HARQ model between different modes and the calculation of thresholds for switching. Initially, the pure ARQ with GBN technique serves as the starting point. Once the number of transmitted packets reaches the size of the sliding window, decisions are made based on the counts of NACKs (Negative Acknowledgments) and ACKs (Acknowledgments) to determine whether a state transition is necessary.

The switching points are determined by identifying the SNR (signal-to-noise ratio) values at which both schemes yield identical throughput. The system transitions to the scheme with the higher throughput for subsequent SNR values until it reaches another switching point. However, under very low SNR conditions (0-3.47 dB), these schemes produce negligible throughput (nearly zero), as demonstrated in Section 4. This paper addresses this limitation by introducing a key enhancement that involves the use of turbo codes.

#### 3.2.1 Turbo Codes for Low SNR Conditions

When neither a positive acknowledgment (ACK) nor a NACK is received, it indicates a scenario of extremely low SNR, resulting in negligible throughput. This behavior is supported by the throughput formula, which is inversely proportional to parameter  $S$ , defined as the ratio of acknowledgment delay to block transmission time. Without acknowledgments, acknowledgment latency increases dramatically, pushing  $S$  closer to infinity. As a result, throughput approaches zero, emphasizing the need for turbo codes in such situations. Using turbo codes in these circumstances significantly increases throughput.



**Fig. 6** Illustration of the dynamic switching between different HARQ schemes

### 3.2.2 Switching Criteria

The number of NACKs governs the switching mechanism received within a sliding window. If the count of NACKs exceeds a threshold,  $\alpha_1$ , the scheme transitions from ARQ to HARQ1, as illustrated in Figure 6. To compute  $\alpha_1$ , the throughput of ARQ is equated to that of HARQ1:

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

The parameters used in this Eq. 9 are defined as follows.  $P_e$  represents the probability of error in the packet in the pure ARQ, while  $P_s$  is the probability that the symbol has an error.  $N$  is the block length, representing the number of symbols in each packet, and  $b$  is the number of bits per symbol in each block.  $CRC$  represents the number of redundancy bits added for error detection.  $S$  is the ratio of acknowledgment delay to block transmission time, defined as  $S = \frac{T_a}{T_b}$ , where  $T_a$  is the acknowledgment delay, and  $T_b$  is the time required to transmit a block. Moreover,  $K$  represents the number of information symbols in the packet.

The unknown variable  $P_b$  (bit error probability) is determined by substituting the relevant formulas with  $P_e$  and  $P_s$ . After determining  $P_b$ , we derive  $P_e$  and determine the threshold  $\alpha_1$  [14].

$$\alpha_1 = P_e \cdot W \quad (10)$$

Where  $W$  represents the sliding window size.

To compute the threshold  $\alpha_2$  for switching from HARQ1 to HARQ2, we equate the throughput expressions of both schemes.

$$\frac{K_1}{N} \times \frac{1 - \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}} = \frac{K_2}{N} \times \frac{1 - \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^N \binom{N}{i} P_s^i (1 - P_s)^{N-i}} \quad (11)$$

Additional parameters introduced here include  $K_1$ , representing the number of information symbols in HARQ1, and  $K_2$ , denoting the number of information symbols in HARQ2.

This equation determines  $P_s$ , and the threshold  $\alpha_2$  [14] is given by:

$$\alpha_2 = P_s \cdot W \quad (12)$$

### 3.2.3 Transition Back to ARQ

When the last sliding window detects no errors, the receiver sends a new acknowledgment type, ACK+, to the transmitter. This triggers a transition from HARQ1 or HARQ2 to ARQ, as shown in Figure 6. The transition depends on the number of consecutive ACKs received. The threshold for this scenario,  $\beta$  [14], is defined as:

$$\beta = W \quad (13)$$

Where  $\beta$  denotes the successful and error-free acknowledgment of every packet in the sliding window, this change improves system efficiency by reducing unnecessary computing overhead when channel conditions are favorable.

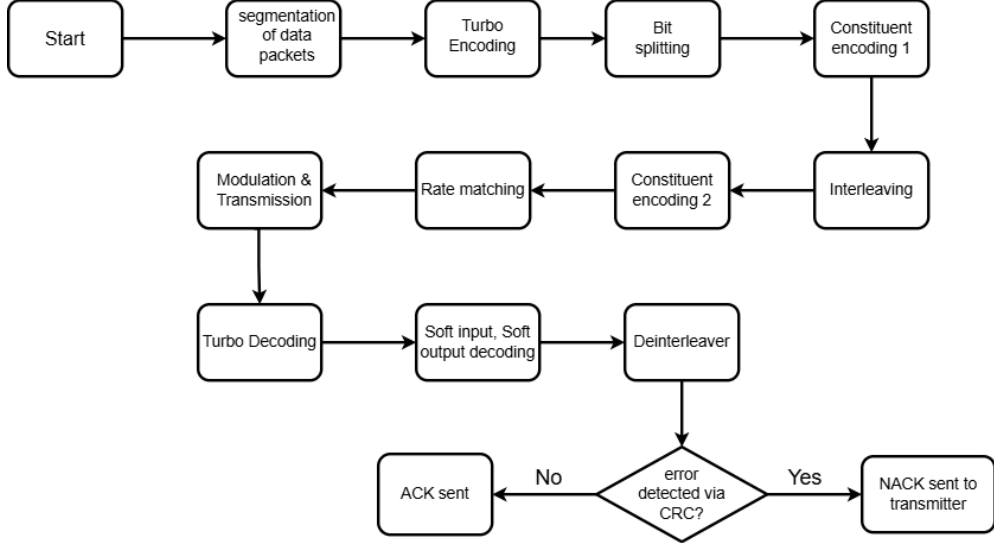
### Summary of Switching Scheme

The adaptive switching process ensures optimal throughput across varying SNR conditions. Turbo codes significantly improve performance under extremely low SNR (0-3.47 dB), as shown in Figure 10. The proposed model balances complexity and efficiency by dynamically selecting the appropriate HARQ scheme based on channel conditions.

## 3.3 Detailed mechanism of HARQ with Turbo Codes

This subsection explains the mechanism of HARQ with turbo codes in the D-HARQ model. Initially, the packet goes through a series of encoding phases to improve error correction capabilities. As shown in Figure 7, the first phase, known as Bit Splitting, prepares the input bitstream for parallel processing. Constituent follows this 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. The process continues with Constituent Encoding 2, adding a layer of redundancy through a second convolutional encoder. Finally, Rate Matching is applied to adjust the encoded bitstream according to the transmission channel's 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, which 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. 7** HARQ with Turbo codes

Soft-input Soft-output Decoding Iteration 1 is the first step in the turbo decoding process, as shown in Figure 7. This phase represents the first attempt to decode the received signal by using soft decision inputs to generate soft decision outputs. These outputs, which contain probability metrics indicating the precision of each bit, significantly increase the effectiveness of the decoding process. The De-interleaver reorders bits between decoding iterations. Subsequent iterations of decoding are required to refine the accuracy of the data. After decoding, the Cyclic Redundancy Check (CRC) finds problems. If errors are discovered, it sends a NACK requesting retransmission; otherwise, it sends an Acknowledgement (ACK) indicating successful reception.

### 3.4 SNR estimation algorithm

This subsection presents a detailed explanation of the SNR estimation algorithm, a key part of the selective soft combining process. The Error Vector Magnitude (EVM) SNR estimation algorithm 1 enhances the accuracy and scope of SNR estimation. This advancement comes in response to the limitations of current methodologies, which struggle with accurately estimating low SNRs. Eq. 14 and Eq. 15 initially calculate the mean and variance of the received signal samples  $|y_n|$  at time instants  $n = 1, 2, \dots, L$  after downsampling.

$$\text{mean} = \frac{1}{L} \sum_{n=1}^L |y_n| \quad (14)$$

$$\text{var} = \frac{1}{L} \sum_{n=1}^L (|y_n| - \text{mean})^2 \quad (15)$$

SNR is calculated from Eq. 16 [21] using these results.

$$SNR = 10 \times \log \left( \frac{|\text{mean}|^2}{2 \times \text{var}} \right) \quad (16)$$

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

$$z = SNR \quad (17)$$

The new SNR' [21] is estimated using the  $z$  value.

$$SNR' = \sqrt{(z - 2.5) \times 39.2} - 7 \quad (18)$$

SNR' gives a precise estimate when the SNR is less than 10 dB; if it is higher, the SNR is deemed acceptable. This approach exhibits reduced deviation and increased estimation accuracy within a broader range of -10 to 30 dB. The complexity of this approach is modest because it simply needs the data's mean and variance.

### 3.5 Decision Making for Soft Combining at Receiver End

This subsection details the decision-making process for soft combining at the receiver, focusing on its selective application based on channel conditions. The receiver estimates the SNR value of the received signal using the EVM algorithm, as discussed in Section 3.4. The EVM algorithm is accurate even at low SNR levels, making it suited for adjusting to changing channel circumstances. Since evaluating the SNR for each received packet imposes significant computing costs, it is conducted on a regular basis to optimize resource utilization.

Soft combining is a technique used to increase decoding accuracy by mixing previously buffered packets with retransmitted data. However, its use is contingent upon the channel conditions. For example, the benefits of soft combining are negligible when the SNR is large and the individual signals are of adequate quality. In these situations, avoiding soft combining minimizes processing delays and conserves computer resources. Soft combining, on the other hand, is necessary to increase decoding



---

**Algorithm 1** SNR Estimation Algorithm

---

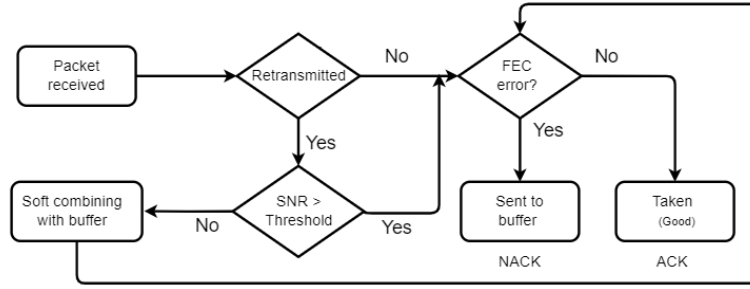
```

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

```

---

performance and reduce retransmissions when the SNR is low due to poor signal quality.



**Fig. 8** Decision-making process for soft combining at the receiver.

Figure 8 shows the decision-making process for soft combining. After receiving a packet, the receiver performs the following three steps:

### 3.5.1 Packet Classification

The receiver first determines whether the packet is an original transmission or a retransmission. This is critical for subsequent processing decisions.

### 3.5.2 SNR Estimation

The EVM algorithm calculates the SNR of the received signal. Based on this estimation:

- *High SNR*: If the SNR exceeds a predetermined threshold, it indicates strong signal quality. Since the signal is probably error-free, the packet is routed straight to the error detection module without soft combining. Unnecessarily combining such high-quality signals with buffered data may reduce decoding accuracy.
- *Low SNR*: The signal quality is poor if the SNR drops below the threshold. In this case, the retransmitted packet is integrated with previously buffered data using soft combining, which increases decoding reliability by utilizing cumulative information from numerous receptions.

### 3.5.3 Error Detection and Feedback

- If the receiver correctly decodes the combined packet and finds it error-free, it accepts it.
- If errors continue to occur, the receiver sends a NACK to request more retransmissions and keeps the packet buffered.

The selective application of soft combining balances computational efficiency and decoding performance. Dynamically adapting to channel conditions enhances throughput and minimizes overall retransmission overhead.

## 4 Results and Analysis

Simulations of D-HARQ were conducted within the Land Mobile and Satellite (LMS) channel using MATLAB (version R2023a) and Python (version 3.10), 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, which is crucial for applications like microwave links and satellite communication.

**Table 2** 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 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 3 illustrates the simulation environment. Consequently, signal propagation in this context follows a lognormal distribution, characterized by its probability density function (PDF) [22], as detailed below:

$$f(x) = \frac{e^{-\left(\frac{(\ln x)^2}{2\sigma^2}\right)}}{x\sigma\sqrt{2\pi}} \quad (19)$$

For  $x > 0$ , Where  $x$  represents the random variable for signal strength and  $\sigma$  denotes the distribution of additive white Gaussian noise (AWGN).

**Table 3** Simulation environment and parameters

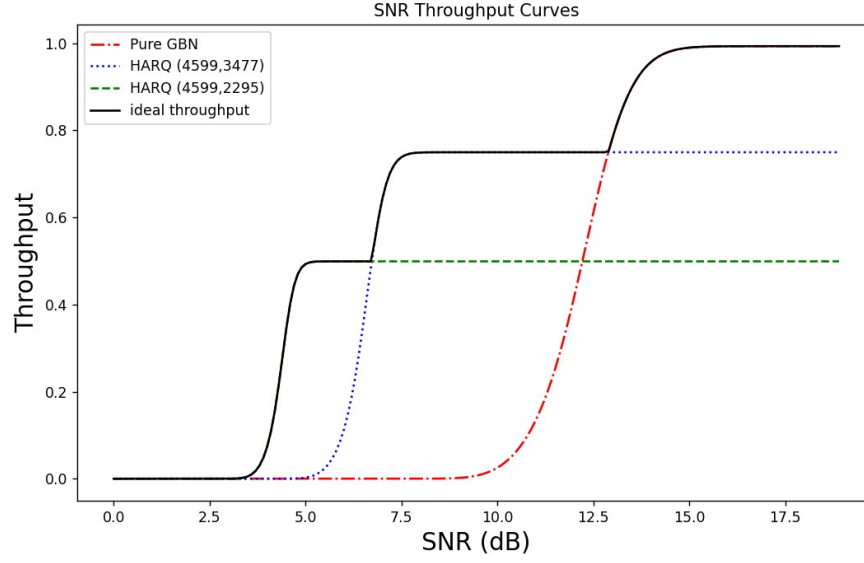
Parameters	Values
Channel model	Land, Mobile, and Satellite
Channel conditions	Rural and Suburban
ARQ Scheme	ARQ, HARQ1, HARQ2
No. of symbols in each block/packet	511
No. of bits per symbol	9
Delay	$5 \times$ No. of information symbols
Noise	White Gaussian (AWGN) noise
Simulation output	MATLAB (version R2023a) and Python (version 3.10)
Simulated SNR range	0-18 dB

Figure 9 illustrates a graph of a D-HARQ model, highlighting the thresholds for transitioning between different modes, established through a trial-and-error methodology. The simulations use various sets of thresholds. Ultimately, the graph aligns almost perfectly with the ideal curve for  $\alpha_1 = 10$ ,  $\alpha_2 = \beta = W = 64$ , where the ideal curve's throughput at a given SNR is expressed as:

$$\eta_{\text{ideal}} = \max(\eta_{\text{ARQ}}, \eta_{\text{HARQ1}}, \eta_{\text{HARQ2}}) \quad (20)$$

Figure 9 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  $2.16 \times 10^{16}$  times. Moreover, Figure 9 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.

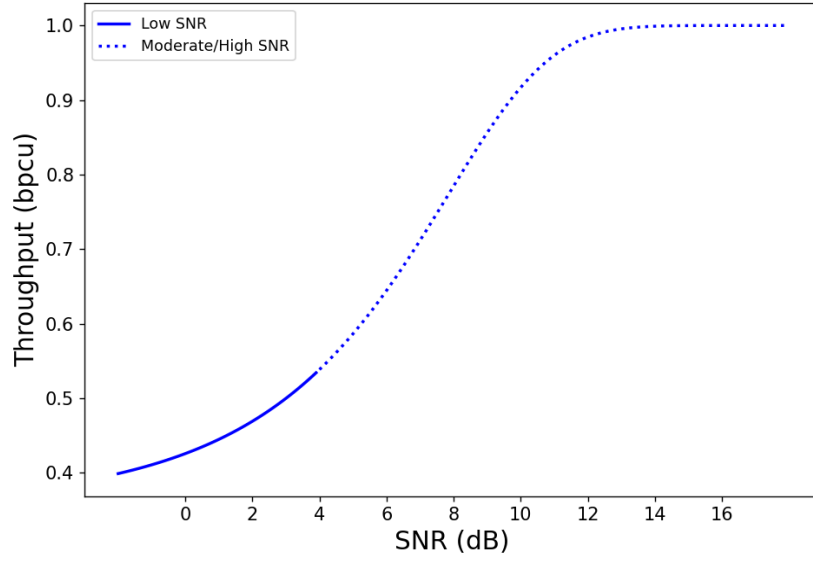


**Fig. 9** Transitions between various HARQ and ARQ schemes

The main drawback observed from Fig 9 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.

Figure 10 shows considerable throughput even at low SNR values, thus an improvement in throughput from 0 to 0.467 bpcu (bits per channel use) by adapting turbo codes. In Figure 10, 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. The design of Turbo Codes especially includes a complex iterative decoding process that relies on information being passed back and forth between two or more constituent decoders. This interaction necessitates a complicated algorithm to properly converge on the correct decoding, greatly complicating implementation compared to the simple algebraic decoding methods employed for BCH codes.

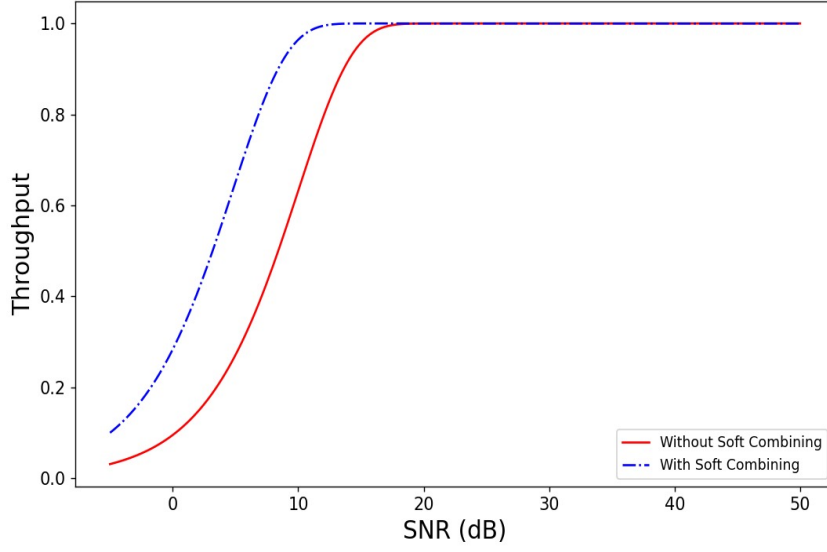
As explained in Section 2, Soft combining is beneficial as combining the retransmitted packet with the error packets in the buffer improves its decoding capability, thus reducing the need for retransmissions. However, at a very high SNR, soft combining does not contribute much towards the throughput improvement. Since these schemes are unnecessary in such scenarios, they are not employed, optimizing the complexity and minimizing the usage of resources.



**Fig. 10** Estimated Throughput vs SNR for HARQ with Turbo codes (code rate = 1/3)

Figure 11 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, an SNR above this threshold indicates that the channel quality is good enough to omit soft combining.

Although D-HARQ introduces added complexity through dynamic switching, turbo code integration, and selective soft combining, these features are essential for significantly improving performance, particularly in low SNR conditions (0–3.47 dB). The system’s ability to adapt to changing channel conditions and minimize retransmissions boosts throughput and resource efficiency. For instance, turbo codes improve throughput from 0 to 0.467 bpcu, as shown in Figure 10, and adaptive HARQ outperforms traditional GBN in the 3.47 dB to 9 dB SNR range, as seen in Figure 9. Furthermore, selective soft combining is only applied when SNR is below 17.46 dB, ensuring optimal use of resources.



**Fig. 11** Performance comparison with and without 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, the selective soft combining method integrates with it to improve performance. The strategic utilization of turbo codes under very low SNR conditions and BCH codes in other scenarios strike an optimal balance between cost-effectiveness and reliability. Soft combining is purposely avoided during favorable channel conditions, reducing the related complexity overhead. Moreover, enhancing performance and throughput. Simulation results that employ turbo coding and selective soft combining confirm the usefulness of this paper's idea. These findings show a significant gain in throughput while lowering complexity at all levels of data transmission. Future work could include simulations using machine learning approaches to determine proper thresholds. Future research can focus on finding ways to deliver high throughput while maintaining low latency.

## Statements & Declarations

### Competing Interests

I have no conflicts of interest to disclose. No financial conflicts with any author or agencies.

### Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

### Author Contributions

Material preparation and analysis were performed by us only.

### Data availability statements (DAS)

As our own

### Research Involving Human and /or Animals

As our own

### Informed Consent

Not applicable

## 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] F. Nadeem, M. Shirvanimoghaddam, Y. Li, B. Vucetic, Non-orthogonal harq for urllc: Design and analysis, *IEEE Internet of Things Journal* 8 (24) (2021) 17596–17610.
- [16] Z. Shi, S. Ma, G. Yang, M.-S. Alouini, Energy-efficient optimization for harq schemes over time-correlated fading channels (2022).
- [17] S.-Y. Lin, M.-H. Yang, S. Jia, An adaptive hybrid automatic repeat request (a-harq) scheme based on reinforcement learning, *Electronics* 12 (19) (2023) 4127.
- [18] U. E. Uyoata, A. Amiri, E. Juan, G. Pocovi, P. Andres-Maldonado, K. I. Pedersen, T. Kolding, Revisiting data recovery loops in 6g networks (2024) 1145–1150.
- [19] K. Kotuliaková, D. Šimlaštíková, J. Polec, Analysis of arq schemes, *Telecommunication Systems* 52 (2013) 1677–1682.
- [20] M. W. El Bahri, F. Raouefi, H. Boujemâa, M. Siala, Performance of harq i schemes using turbo codes, in: *2005 12th IEEE International Conference on Electronics, Circuits and Systems*, IEEE, 2005, pp. 1–4.
- [21] X. Qun, Z. Jian, Improved snr estimation algorithm, in: *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, IEEE, 2017, pp. 1458–1461.
- [22] J. Poctavek, K. Kotuliaková, J. Polec, M. Osadský, S. Ondrusova, Throughput parameter optimization of adaptive arq/harq scheme, *International Journal of Communication Networks and Information Security* 3 (1) (2011) 89.