SNR-Responsive Communication: Turbo Codes and BCH in a Adaptive HARQ scheme for Enhanced Efficiency.

C.Gnana Jyothi, P.Chandana Sai Sri Hasitha, N.Yaswanth
Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India
9346796854, 9030495457, 9676794131
chinthagnanajyothi.221cs118@nitk.edu.in, prathapachandanasaisrihasitha.221cs139@nitk.edu.in, namburiyaswanth.221cs232@nitk.edu.in

April 25, 2024

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) [1]methods suffer from high communication overhead and latency and are very inefficient in handling retransmissions. Although traditional HARQ [2](Hybrid Automatic Repeat Request) is effective, its widespread use is limited by high implementation costs. This underscores the need for adaptive systems that can dynamically respond to changing channel conditions based on Signal-to-Noise ratio (SNR) values [3]. The proposed scheme switches between ARQ, HARQ with Bose-Chaudhuri-Hocquenghem (BCH) codes [4], and HARQ with turbo codes [5]based on channel conditions during the transmission. Further, this scheme incorporates a selective soft combining technique by estimating SNR at the receiver end. The main goal is to achieve the maximum possible throughput even at very low SNR values while maintaining optimal complexity.

I. Introduction

Error control mechanisms play a crucial role in reliable data transmission. It ensures the retransmission of data packets that are found to be erroneous or lost during the transmission. The end-to-end transfer of data from one application to another application involves several steps, each subject to errors along the way [6]. Due to the necessity of numerous retransmissions, standard Automatic Repeat Request (ARQ) schemes fall short of adequately correcting all errors. Thus, we are leveraging the technology of hybrid ARQ, which integrates ARQ with Forward Error Correction (FEC) [7], [8] for its enhanced efficiency in this context. However, this system may not be necessary under good channel conditions. To address this, we are developing an adaptive approach that adjusts to the varying needs of different channel conditions.

A. Go-Back-N (GBN) scheme [9] :

GBN scheme is one of the Pure ARQ methods. [9] In this scheme, the sender continues to send 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) [10] is sent and the sender resends all frames starting from the last acknowledged frame as shown in Fig.1. This scheme does not require a storage buffer at the receiver side. GBN guarantees data integrity and reduces congestion of the network.

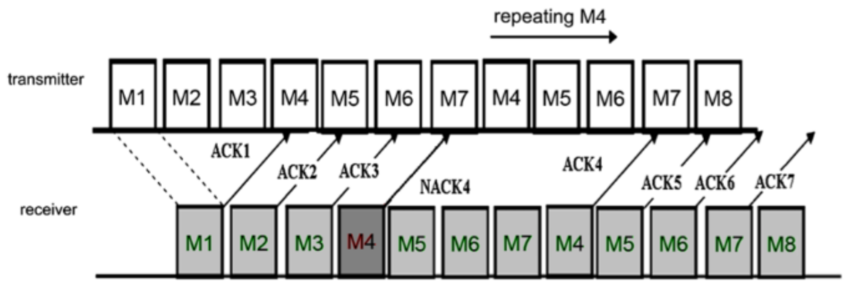

Figure 1: Go Back-N (GBN) scheme [10]

B. Hybrid ARQ [2]:

Hybrid ARQ is ARQ concatenated with FEC [2].This facilitates error correction during transmission, minimizing the likelihood of receiving additional errors. Retransmission occurs when packet errors remain uncorrected. HARQ has mainly two basic schemes: Hybrid ARQ Type I, having adaptive coding rates and extra additional error correcting codes to minimize retransmissions while keeping the data integrity [2]. And Hybrid ARQ Type II uses redundancy to improve error correction capabilities later [2]. These approaches guarantee robust data transmission, optimizing throughput and reliable data communication .

C. Turbo codes [11] :

Turbo codes are a part of error-correcting codes that employ parallel concatenated convolutional codes to achieve remarkably efficient error correction [11]. 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 [11]. This iterative decoding process allows turbo codes to approach the Shannon limit [11], providing robust error correction even in challenging transmission environments. Despite high error-correcting capability, they are quite complex to implement.

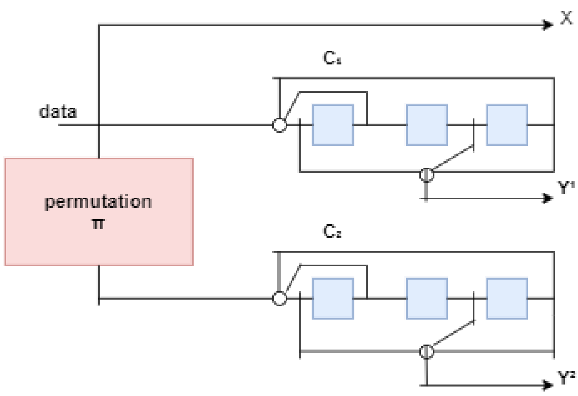

Figure 2: The classical Turbo code [5]

D. Soft combining [12] :

Soft combining is a scheme in which a received erroneous packet is stored in a buffer rather than discarding, It may happen that two independently uncorrectable erroneous packets can be decoded correctly by utilizing the useful information obtained from their combination [12]. This scheme has two main methods: *chase combing* and *Incremental Redundancy (IR)* [12]. 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 [13].In Incremental redundancy, different versions of redundancy bits are obtained by puncturing the encoder output thus adding additional information in each retransmission [13].

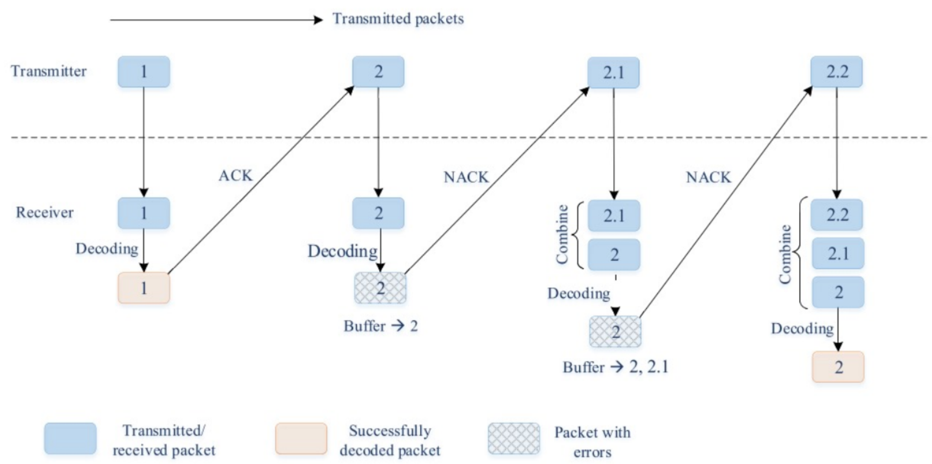

Figure 3: Illustration of Soft combining with HARQ [13]

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 [7] 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 [14].

The proposed scheme 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 of 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 stage of data transmission will undoubtedly result in significant throughput improvements across all scenarios.

The rest of the paper is organized as follows, section II covers the literature Survey, while Section III explains the design and implementation of the proposed solution. Section IV presents simulation results, analysis, and comparisons with existing methods. Finally, section V concludes with future directions and references.

II. Literature survey

This section of the paper discusses the previously proposed approaches on adaptive HARQ with their advantages and drawbacks.

A. *Adaptive HARQ with RS codes* [15]

J. P. Peter Fidler and Kvetoslava Kotuliakova [15] 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 [15]. A 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 [15]. Once the number of consecutive ACKs exceeds this threshold, the system shifts to the less complex ARQ mode, signifying a stable channel condition. Conversely, the detection of even a single NACK prompts an immediate switch back to HARQ mode, reflecting the need for enhanced error correction in response to channel degradation [15].

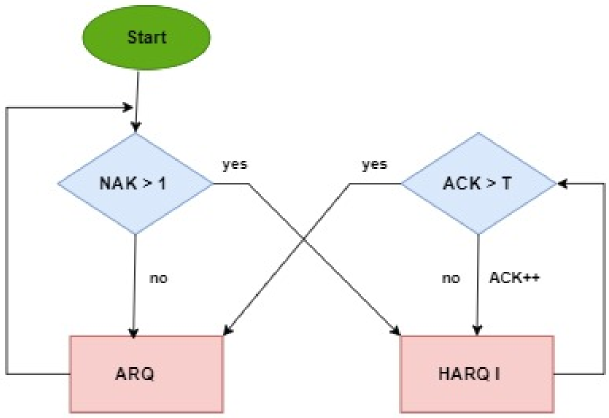

Figure 4: Flowchart of Adaptive HARQ using RS codes [15]

This schema offers several advantages. Firstly, its adaptive system dynamics introduce an asymmetrical strategy as shown in Fig.4, 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, 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.

B. Adaptive HARQ with two RS codes [16]

IK. K. Michal Martinovi˘c and Jaroslav Polec [16] 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 [14], HARQ1 with RS code (511, 383, 64) and HARQ2 with RS code (511, 255, 128) [16]. For state L of low channel error rate transmitter follows the pure ARQ method. The previously discussed high channel error rate is now divided into HARQ1 and HARQ2. Both states use RS code but with different code rates. The switching logic operates in a way that the transmitter manages the switch from ARQ to HARQ modes or HARQ1 to HARQ2. And receiver handles the switch from HARQ modes to ARQ modes [16]. The ideology of this schema is explained in Fig.5

These three confirmations are utilized in the proposal:

- ACK when the packet contains no errors (pure ARQ) or when the packet can be corrected by the RS code (HARQ1, HARQ2) [16].
- NAK when the RS code is unable to rectify the packet (HARQ1, HARQ2) [16].
- ACK + When there is no error in the previous W packets [16].

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. but even this approach fails to handle a broader range of SNR values, optimizing throughput and minimizing latency. This approach also comes with drawbacks such as increased latency for real-time traffic and limited capability in correcting random errors due to its Reed-Solomon error correcting code.

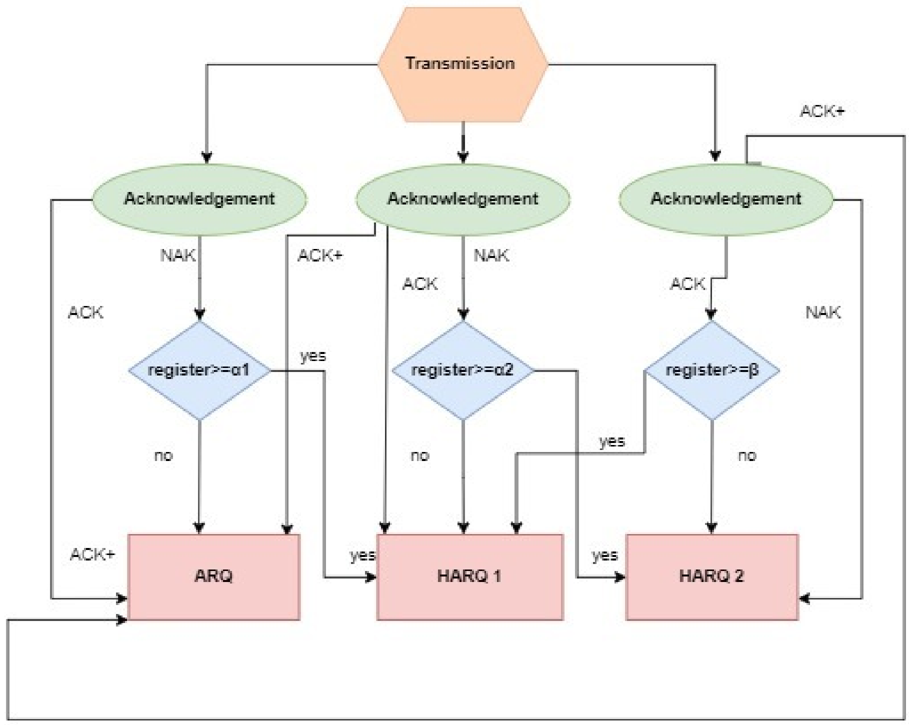

Figure 5: Flowchart of Dynamic Adaption of different HARQ schemes using two RS codes [16]

C. Adaptive HARQ with BCH codes [6]:

F. C. Kvetoslava Kotuliakova and Jaroslav Polec [6] presented an enhanced adaptive ARQ-HARQ method that makes use of BCH codes and is intended to improve the effectiveness of data transmission over varying channel conditions [6]. This scheme dynamically modifies transmission schemes in response to the channel's measured bit error rate. When there are low error rates, it first uses a pure ARQ mode to provide data reliably and redundantly. But the system smoothly switches to hybrid ARQ schemes as the error rate rises to moderate to high levels [6].

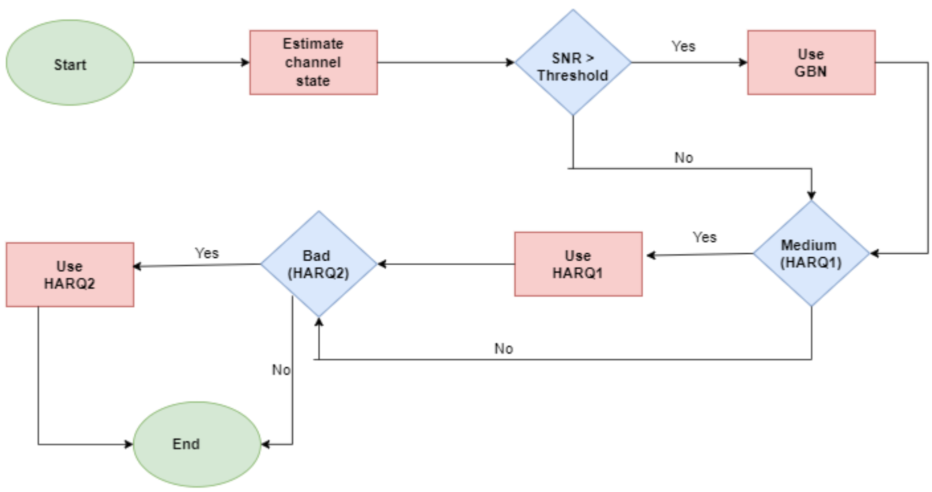

Figure 6: Flowchart of Adaptive HARQ using BCH codes [6]

These hybrid schemes combine two different BCH codes that were specifically selected to improve error correction performance on a range of channel conditions. In every mode, the Go-Back-N [14] protocol is used for better performance as discussed in section IA. This adaptive strategy shows great promise for establishing reliable and effective data communication in a variety of 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.

It is clear from the literature survey that throughput tends to zero at low SNR levels. So this became the motivation to propose a new kind of adaptive HARQ that increases throughput at all channel conditions.

III. Design and Implementation

The proposed method focuses on improving communication efficiency by developing an adaptive Hybrid Automatic Repeat Request (HARQ) scheme. This scheme dynamically adapts to different schemes based on the real-time fluctuations in signal-to-noise ratio (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) [6] and HARQ2(4599,2295) [6], and an optimal state using HARQ with Turbo codes.
For the estimation of SNR, the following algorithm is used.

A. SNR estimation algorithm [3]:

We are using The EVM SNR estimation algorithm, aiming to improve the accuracy and range of SNR estimation compared to existing methods because, in the existing algorithms, the low SNR can't be estimated accurately [3] . The algorithm proposed in this paper even estimates very low SNR values [3]. In this algorithm error vector magnitude (EVM) approach with modifications, initially computes the mean and variance of received signals . They are calculated as below in (1),(2)

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

[3]

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

[3]
Using these results SNR is calculated from (3)

$$SNR = 10 \times \log\left(\frac{|\text{mean}|^2}{2 \times \text{var}}\right) \tag{3}$$

[3] Then, when the estimated value is less than 10dB , the z-value (intermediate value) is calculated

$$z = SNR \tag{4}$$

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

$$\text{SNR'} = \sqrt{(z - 2.5) \times 39.2} - 7 \tag{5}$$

[3]
If the SNR value is less than 10dB we will use SNR' otherwise we will proceed with SNR [3]. This algorithm has higher estimation accuracy and less deviation in a larger range between -10 to 30 dB [3] . This algorithm's complexity is low as it uses only the mean and variance of data [3].

B. Throughput of different schemes :

If the transmitter follows pure ARQ with GBN scheme, then the throughput is given by equation (6). Due to error detection bits, this throughput is multiplied by the weight i.e. ratio of message symbols to the total number of symbols.

$$\eta_{\text{L}} = \frac{1 - P_e}{1 + S \cdot P_e} \cdot \frac{N - CRC}{N} \tag{6}$$

[16] , [6] Here P_e is the probability of occurrence of an error in packet in pure ARQ method. P_e depends on channel bit error rate, burst errors, and block length [17]. This is expressed in (7)

$$P_e = 1 - (1 - P_b)^K \tag{7}$$

[6] where n is the length of the block, 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 time delay of acknowledgment to the time of block transmission

$$S = T_a/T_b \tag{8}$$

[17]

T_a is the time delay of acknowledgment, defined as the time delay from terminating of block transmission to receiving and processing block acknowledgment and T_b is the time of block transmission , N is the length of packet in symbols also termed as packet size and b is no.of bits per each symbol [17].

$$b = \log_2(N + 1) \tag{9}$$

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

$$\eta_H = \frac{K}{N} \times \frac{1 - \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1 - P_s)^{N-i}}{1 + S \cdot \sum_{i=t+1}^{N} \binom{N}{i} P_s^i (1 - P_s)^{N-i}} \tag{10}$$

[17], [16]

where, P_s is the probability that the symbol has error and is calculated as below

$$P_s = 1 - (1 - P_b)^b \tag{11}$$

[16] , [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

$$T_{\text{hr}} = \left(\frac{R_c}{T_r}\right) \cdot \left(\frac{k}{k + n_p}\right) \tag{12}$$

[18] 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. T_r is expressed as

$$T_r = \sum_{i=0}^{\infty} P(D_d)^i = \frac{1}{1 - P(D_d)} \tag{13}$$

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

C. Switching scheme :

In the proposed scheme, initially, the pure Go-Back-N (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 negative acknowledgments (NACKs) and acknowledgments(ACKs) 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 in such a way that the scheme with relatively higher throughput for subsequent SNR values is chosen up 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 Fig.9. To overcome this limitation in such scenarios, a key enhancement of switching to turbo codes is implemented. As discussed earlier, turbo codes include both convolutional and block codes in their range of encoding methods, offering high complexity but also enhanced accuracy. [5] To strike an optimal balance between complexity and accuracy, turbo codes are utilized only in scenarios characterized by very low SNR conditions.

When neither a positive nor a negative acknowledgment is received, it suggests a scenario of extremely low Signal-to-Noise Ratio (SNR), leading to no throughput, as illustrated in the referenced fig.9. This concept is further supported by the throughput formula, which is inversely proportional to term S.

S is the ratio of acknowledgment delay to block transmission time [17], 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 in the denominator of the throughput equation, a large value for S results in the throughput nearing zero. Consequently, the absence of any form of acknowledgment can be interpreted as an indication of very low SNR, specifically within the range of 0 to 3.47 dB.

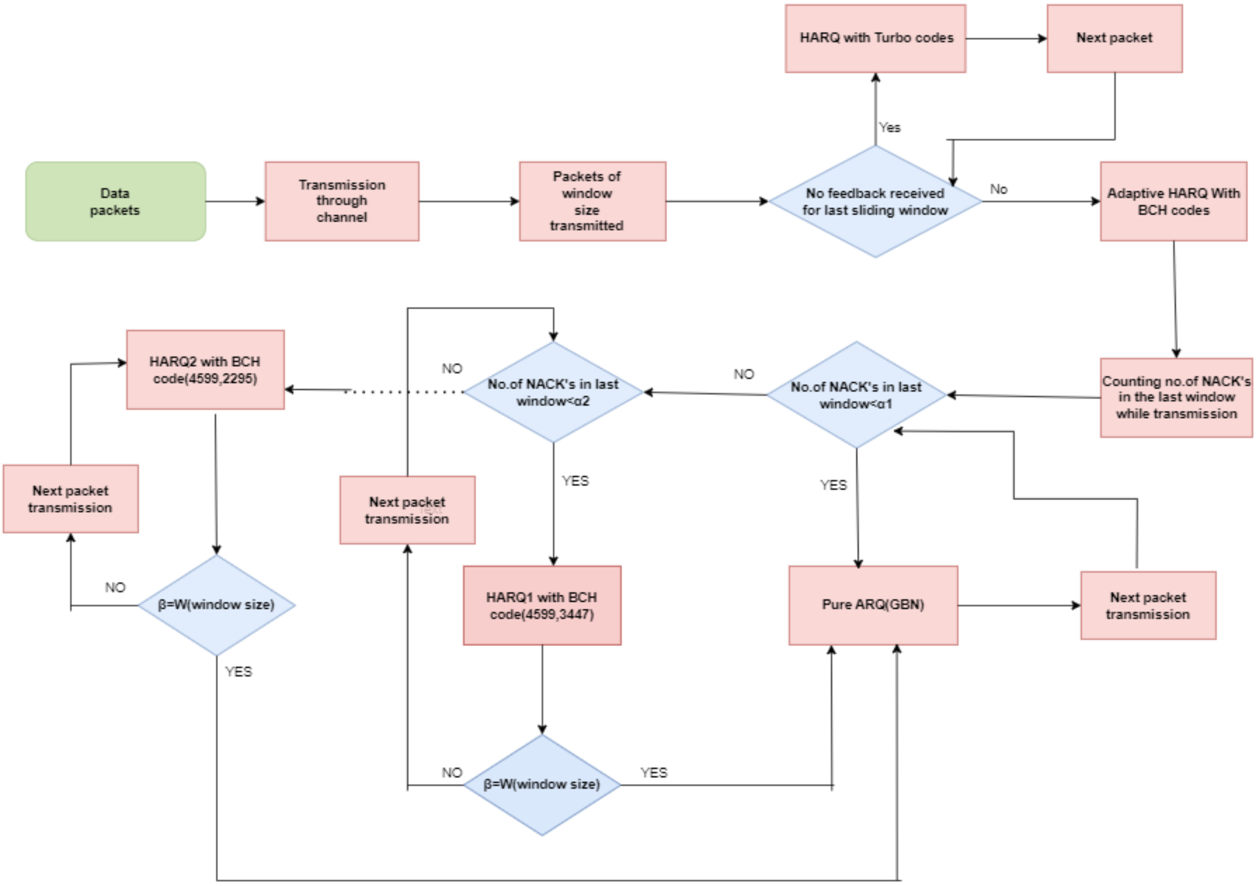

Figure 7: Flowchart illustrating the dynamic switching between different HARQ schemes

The main parameter used to switch in between the schemes is the number of negative acknowledgments. If the count of NACKs crosses the threshold α_1, the scheme switches from ARQ to HARQ1. To calculate α_1 ,First equate the throughput of ARQ with throughput of HARQ1.

$$\frac{1-P_e}{1+S\cdot P_e}\cdot\frac{N\cdot b-CRC}{N\cdot b}=\frac{K}{N}\times\frac{1-\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}}{1+S\cdot\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}} \tag{14}$$

[6]
Replace P_e with equation (7) and P_s with equation (11) in the equation (14).

The unknown variable P_b can be calculated by solving the above equality . By substituting this P_b value in (7), P_e can be derived and the threshold α_1 is given by

$$\alpha_1 = P_e \cdot W \tag{15}$$

[16] where W denotes window size.

Similarly, To calculate the threshold α_2 to switch from HARQ1 and HARQ2 , equate the throughput of HARQ1 with throughput of HARQ2.

$$\frac{K1}{N}\times\frac{1-\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}}{1+S\cdot\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}}=\frac{K2}{N}\times\frac{1-\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}}{1+S\cdot\sum_{i=t+1}^{N}\binom{N}{i}P_s^i(1-P_s)^{N-i}} \tag{16}$$

[6]

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

$$\alpha_2 = P_s \cdot W \tag{17}$$

[16]

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. This transition back to pure ARQ when channel conditions are favorable, significantly reduces unnecessary costs. This threshold for the number of acknowledgments is denoted by β. Where β=W [16], indicates that W acknowledgments are sent consecutively. The detailed flow of the switching scheme is depicted in the flowchart Fig.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 fig.10 , This suggests that turbo codes could effectively enhance throughput in this specific range.

The algorithm below explains the whole switching scheme.

Algorithm 1 Proposed SNR responsive Adaptive HARQ Scheme(Transmitter end logic)

initialize: mode $\leftarrow$ ARQ, α_1, α_2, β, S
while communication is active **do**
  count ACKs, NACKs
  if SNR is very low **then**
    mode $\leftarrow$ Turbo-HARQ
  else if SNR is moderate or high **then**
    if NACKs $\geq \alpha_1$ **then**
      mode $\leftarrow$ HARQ1
    else if NACKs $\geq \alpha_2$ **then**
      mode $\leftarrow$ HARQ2
    end if
  else if SNR is extremely high **then**
    if ACK is received and $\beta = W$ **then**
      mode $\leftarrow$ ARQ
    end if
  end if
  transmit packets according to mode
end while=0

D. Decision Making for Soft Combining at Receiver End

At the receiver end choice of soft combining is made by estimating SNR value of the received signal by using the EVM [3]algorithm as discussed above in section-IIIA. Using this algorithm is advantageous as it is capable of estimating accurate SNR values even at very low SNR conditions.Since calculating SNR for every signal can turn out to be complex at the receiver end, calculating at regular intervals reduces the overhead.

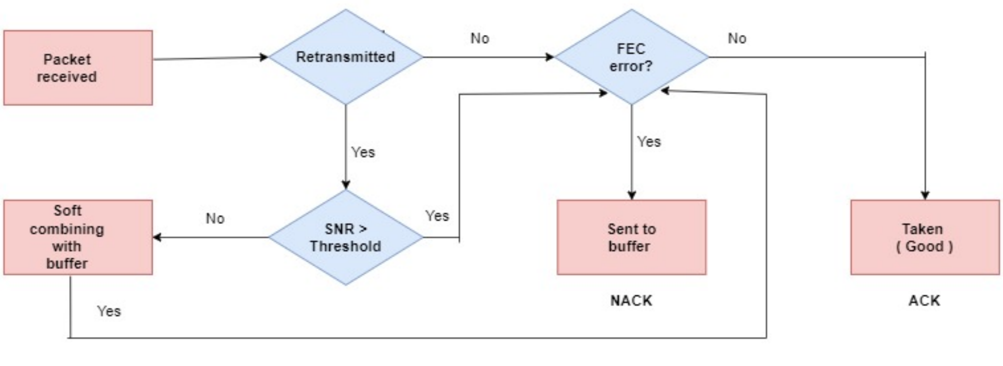

Figure 8: Flowchart illustrating the decision making for soft combining at the receiver end

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 Fig.11. 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.

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

On the other hand, if the packet is identified as a retransmitted one, a decision must be made on whether to apply soft combining. This decision is based on the estimated SNR, calculated from EVM [3] 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. Conversely, if the SNR does not meet the threshold, indicating a weaker signal, soft combining is employed. This strategy enhances throughput and decreases the number of required retransmissions by utilizing the cumulative information from multiple receptions of the same packet. This is shown step by step in Fig. 9 and Algorithm 2.

Algorithm 2 Soft Combining Based on Channel Conditions(Reciever end logic)

```
0: Receive packet
0: if packet is original transmission then
0:    Send to error detection module
0:    if no errors detected then
0:       Accept packet
0:    else
0:       Forward packet to buffer
0:       Send NACK for retransmission
0:    end if
0: else
0:    Calculate SNR using EVM algorithm
0:    if SNR > threshold then
0:       Send packet to error detection module without soft combining
0:    else
0:       Apply soft combining
0:       Merge packet with errored packets in buffer
0:    end if
0: end if=0
```

IV. Results and Anlaysis

Simulations of the proposed scheme were conducted within the Land Mobile and Satellite (LMS) channel, focusing on rural and suburban environments [19]. Under these conditions, the loss of energy in the Line of Sight (LOS) [19] wave is comparatively insignificant. Consequently, signal propagation in this context follows a lognormal distribution [19] , as detailed below

$$f(x) = \frac{e^{-\left(\frac{(\ln x)^2}{2\sigma^2}\right)}}{x\sigma\sqrt{2\pi}} \tag{18}$$

[19]

for x > 0, where σ is the distribution of additive white Gaussian (AWGN) noise [19].
The following table 1 describes the simulation environment The outcomes of the simulations are illustrated through a graph plotted between signal-to-noise ratio (SNR) versus throughput.

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 * No.of information symbols
Noice	white guassian (AWGN) noise
Simulation output	matlab with python codes
Simulated SNR range	0-18dB.

Table 1: Simulation environment and parameters

Table 2 shows information and redundancy bit allocations of various schemes, showcasing their code rates in error management during data transmission.

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

Table 2: Number of Information and redundancy bits for various schemes

Fig.9 illustrates a graph of an adaptive Hybrid Automatic Repeat request (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, for $\alpha 1 = 10$, $\alpha 2 = \beta$ =W=64 [6] the graph is almost aligned with the ideal curve. Where throughput of ideal curve at given SNR is expressed as

$$\eta_{\text{ideal}} = \max(\eta_{\text{ARQ}}, \eta_{\text{HARQ1}}, \eta_{\text{HARQ2}}) \tag{19}$$

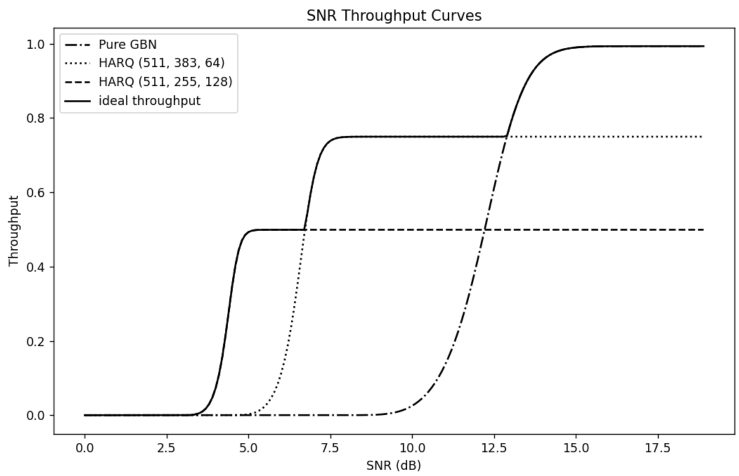

Figure 9: Transitions between various HARQ and ARQ schemes [6]

Fig.9, demonstrates the performance benefits of implementing an adaptive Hybrid Automatic Repeat reQuest (HARQ) strategy within the signal-to-noise ratio (SNR) range of 3.47 dB to 9 dB, compared to employing a traditional Go-Back-N (GBN) approach. The adoption of the adaptive HARQ method in this specific SNR interval results in a substantial increase in throughput, achieving a gain of approximately 2.16×10^{16} times.

Moreover, the figure highlights the enhanced throughput achieved by utilizing two different states of HARQ, each configured with Bose-Chaudhuri-Hocquenghem (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 proposed approach recommends transitioning from HARQ back to ARQ at higher SNR values, a strategy that is 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.

But, the main draw back observed from the graph is very negligible throughput nearing zero in the range 0 to 3.47dB.This is avoided in the proposed scheme by adapting HARQ with turbo codes in this range. As Fig.10, shows considerable throughput even at low SNR values, Thus an improvement of throughput from 0 to 0.467 bpcu by adapting turbo codes.

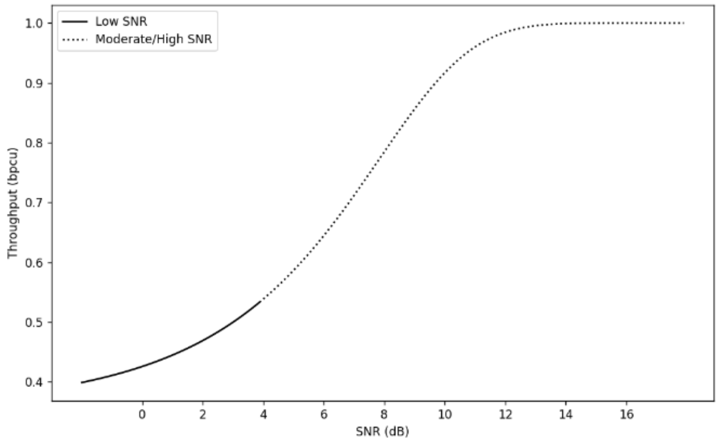

Figure 10: Estimated Throughput vs SNR for HARQ with Turbo codes (code rate = 1/3)

From the Fig.10, one might question the feasibility of utilizing turbo codes across all SNR ranges. However, the complexity vs. SNR graph from Fig.12 resolves this ambiguity. It is evident from the graph that turbo codes exhibit significantly high complexity, which, when employed at moderate to high SNR values, considerably increases overhead. Although turbo codes yield commendable throughput, the rapidly increasing complexity at moderate to high SNR levels essentially negates this advantage.

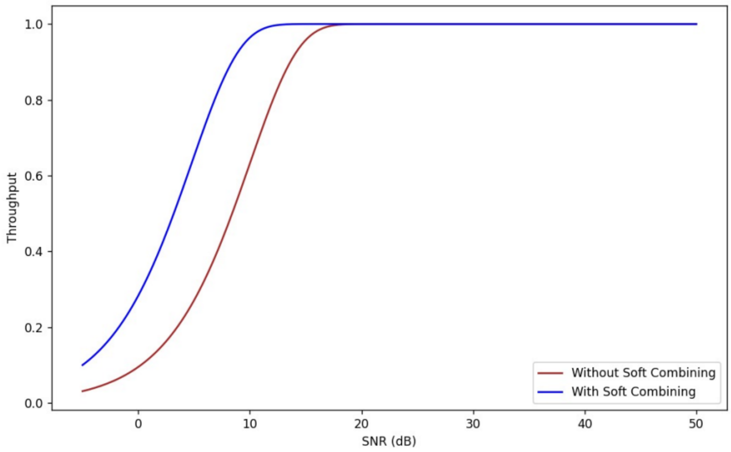

Figure 11: Performance comparision with and without Softcombing

As explained in SECTION-II, 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.11 shows the comparison of throughput with and without soft combining at all SNR values ranging from 0 to 50 dB. Thus the threshold to selectively soft combine is 17.46dB, therefore SNR above this threshold indicates that the channel quality is good enough to omit soft combining.

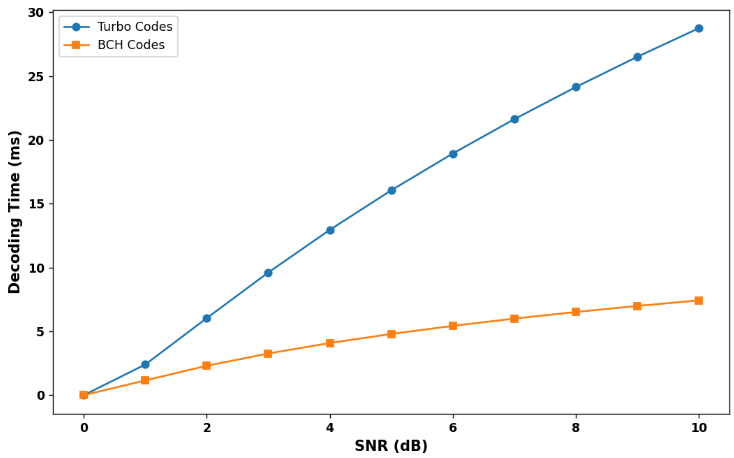

Figure 12: Decoding Time vs SNR for Turbo Codes and BCH Codes

V. Conclusion and Future works

In this paper, we have developed a SNR responsive scheme that switches between ARQ and HARQ using BCH and turbo coding. Additionally, we've integrated a selective soft combining method for improved performance. The strategic utilization of turbo codes under very low SNR conditions, coupled with 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 and also improves performance and maximizes throughput. The proposed scheme's effectiveness is validated through simulation results using turbo coding and selective soft combining. These results demonstrate a significant improvement in throughput while also reducing complexity at every level of data transmission.

The simulations in this paper were conducted within the framework of the Land Mobile Satellite (LMS) [19] channel, specifically under rural conditions. Future work could involve extending these simulations to other communication channels and determining thresholds through comprehensive simulations. Additionally, simulations for thresholds in our paper are done in trail and error methods.this can be made more accurate by leveraging ML and deep learning schemes.

References

[1] S. Lin, D. J.Costell, M. J. Miller, Automatic-repeat-request error-control schemes, https://ieeexplore.ieee.org/document/1091865, [Accessed: 08-01-2024] (1984).

[2] Y. Iraqi, H. Mukhtar, M. Naeem, E. Hossain, Hybrid Automatic Repeat Request (HARQ) in Wireless Communications Systems and Standards: A Contemporary Survey, https://ieeexplore.ieee.org/document/9471818, [Accessed: 07-01-2024] (2021).

[3] X. Qun, Z. Jian, Improved snr estimation algorithm, https://ieeexplore.ieee.org/document/8446894, [] (2017).

[4] L. Joiner, J. Komo, Decoding binary BCH codes, https://ieeexplore.ieee.org/abstract/document/513059, [Accessed: 10-01-2024] (2002).

[5] C. Berrou, R. Pyndiah, P. Adde, C. Douillard, R. L. Bidan, An overview of turbo codes and their applications, https://ieeexplore.ieee.org/document/1617639, [] (2005).

[6] K. Kotuliaková, J. Polec, F. Csóka, An adaptive arq - harq method with bch codes, https://ieeexplore.ieee.org/document/8687301, [Accessed: 08-01-2024] (2017).

[7] R. Kadel, K. Paudel, D. B. Guruge, S. J. Halder, Opportunities and challenges for error control schemes for wireless sensor networks: A review, https://www.mdpi.com/2079-9292/9/3/504, [Accessed: 05-01-2024] (2020).

[8] Q. Li, C.-X. Chen, A hybrid arq protocol for the communication system with multiple channels, https://ieeexplore.ieee.org/document/8539714, [Accessed: 05-01-2024] (2018).

[9] F. A. Nada, Performance analysis of go-back-n arq protocol used in data transmission over noisy channels, https://www.semanticscholar.org/paper/Performance-Analysis-of-Go-Back-N-ARQ-Protocol-Used-Nada/442ac15ee63f92acfcf523f41e3cfee8c2066b4d, [] (2020).

[10] Y.-D. Yao, An Effective Go-Back-N ARQ Scheme for Variable-Error-Rate Channels, https://ieeexplore.ieee.org/document/385946, [Accessed: 05-01-2024] (1995).

[11] X.-G. Xia, Understanding turbo codes: A signal processing study, https://www.sciencedirect.com/science/article/pii/S2949715923000616, [] (2024).

[12] M. El-Khamy, J. Lee, I. Kang, Soft turbo harq combining, https://ieeexplore.ieee.org/document/6655474, [Accessed: 05-01-2024] (2013).

[13] X.-G. Xia, Understanding turbo codes: A signal processing study, https://www.sciencedirect.com/science/article/pii/S2949715923000616, [] (2024).

[14] A. Sastry, Improving automatic repeat-request (arq) performance on satellite channels under high error rate conditions, https://ieeexplore.ieee.org/document/1092826, [] (1975).

[15] P. Fidler, K. Kotuliakova, J. Polec, New adaptive ARQ/HARQ scheme using RS coding and its throughput analysis, https://ieeexplore.ieee.org/document/5167866?denied=, [Accessed: 12-01-2024] (2009).

[16] M. Martinovič, J. Polec, K. Kotuliaková, An Adaptive ARQ – HARQ Method with Two RS Codes, https://publications.waset.org/5645/an-adaptive-arq-harq-method-with-two-rs-codes, [Accessed: 10-01-2024] (2013).

[17] K. Kotuliaková, D. Šimlaštíková J. Polec, Analysis of arq schemes, https://link.springer.com/article/10.1007/s11235-011-9659-1, [] (2011).

[18] M. Siala, M. W. E. Bahri, F. Raouefi, H. Boujemaa, Performance of harq i schemes using turbo codes, https://ieeexplore.ieee.org/document/4633543, [] (2006).

[19] J. Poctavek, K. Kotuliaková, J. Polec, M. Osadský, S. Ondrušová, Throughput parameter optimization of adaptive arq/harq scheme, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=234d771e74eafe3b5c5b4dcaa5c6eb257af851a7, [] (2011).