ARQ Protocol on Mobile Communication and Networks

Prajit Paul¹, Subhabrata Barman²

Abstract—Modified Packet Combining (MPC) scheme is an error correction scheme, which can correct double or, higher bit errors. In some earlier work, a scheme known as Packet Reversed Packet Combining (PRPC) scheme that will correct error if occurs at same bit location of erroneous copies, was studied. Both PRPC and MPC scheme are believed to offer higher throughput and better error correction capability at high bit error rate (BER) and large packet size only. Exact analysis was performed to substantiate this claim of higher throughput and better error correction capability in high as well as in low bit error rate (BER) for large and medium packet size. To combat high bit error rate some researches proposed to use Error Correction Code (ECC) in place of Error Detection Code (EDC) in ARQ scheme. But no conclusive result is found to justify the idea in terms of throughput. The current investigation is an attempt to explore the basic gap. In this paper, we investigate the exact analysis of both PRPC and MPC in a Modified way over Conventional PRPC, and report the findings thereof. We propose a combined protocol of *Modified PRPC and MPC*, and find that it will offer higher throughput and better error correction capability at high and low BER with medium and large packet size as we have used MPC (Modified Packet Combining) scheme with PRPC (Packet Reverse Packet Combining) scheme in a Modified way by combining BEC (Backward Error Correction) with FEC (Forward Error Correction) Codes for Mobile Communication and Networks.

Keywords—PRPC, MPC, Modified-MPC+PRPC, ECC

I. INTRODUCTION

The promise of the Global Inter net for high-speed data services implies a network with both terrestrial wireless, mobile and satellite communication channels. However, the presence of a satellite segment introduces two particular problems that constitute the primary challenge for high-speed data services: burst errors and long propagation delay. To resist errors, ARQ and FEC have been widely adopted - ARQ combats channel errors through retransmission and FEC through redundancy. Forward Error Correction (FEC) coding is the process of adding redundancy to a bit stream according to some rule known to both the transmitter and receiver.

Related Work:

In the event that the data stream is corrupted by channel impairment, this redundancy can be used to compensate for some of these errors. Packet combining (PC) scheme for correction of bit error using erroneous copies at the receiver was introduced by Chakraborty [1]. In the scheme two erroneous copies are XORed for locating the position of bit(s)

in error of the packet, so that the receiver can correct the error rather than requesting transmitter to retransmit the packet. The correction process proposed is the brute force bit by bit inversion of the located bit positions and followed by FCS (Frame Check Sequence) check. The scheme Chakraborty fails: (i) When bit error locations in erroneous copies are same and (ii) When multiple bit errors occur as then the application of brute force bit inversion for correcting will be huge and complex. For n bits in error ($n > 1$), on average $2^{(n-1)}$ trails of attack are required.

A scheme known as Packet Reversed Packet Combining (PRPC) has been studied [2] to tackle the first problem of PC. Modified Packet Combining (MPC) [2, 3] was reported to tackle the multiple bit errors in the received erroneous copies of packet. MPC can't also tackle error if occurs at the same location until odd number of erroneous copies are available that too only when transmitted bit 0 is converted to 1 in all copies in same location but not when transmitted bit 1 is converted to 0 in all copies in same location.

Both PRPC and MPC scheme individually or in combination are believed to offer higher throughput than that of basic ARQ. Any error correction scheme for networks will mainly address the correction for single bit error or maximum for double bits error. When bit error rate is 10^{-3} or less, the probability of more than double bit error in the packet is insignificant. In view, of this PRPC and MPC are prominent scheme subject to offering tolerable throughput. In previous studies neither adequate investigation nor exact analysis was done to substantiate this claim of higher throughput. In this work we investigate exact analysis of both PRPC and MPC. We also propose a combined scheme of PRPC and MPC, and find that proposed combined scheme offers higher throughput and better error correction capability.

A. PRPC

Any error correction scheme for networks will mainly address the correction for single bit error. When bit error rate is 10^{-2} or less, the probability of double bit error or higher in the packet is insignificant compared to that of single bit error. This makes the PC as a powerful scheme for networks. To investigate, how to correct error when bit locations of the error in two erroneous copies used in PC, the idea is that, the receiver when receives an erroneous packet and requests for retransmission of another copy without discarding the first erroneous copy, the transmitter transmits a bit reversed packet of original packet [2-3].

Example- (1) Say the original packet is, 00110101. Say on first transmission the receiver receives the packet as 001**1**1101 (call it first copy) (error location is marked bold, 5th bit from left). Receiver requests for retransmission. Transmitter retransmits a copy with bit reversed as:

Prajit Paul is with Asansol Engineering College, W.B, India.
Subhabrata Barman is with Haldia Institute of Technology, W.B, India.

10101100 (bit wise reversed copy of original packet). Say receiver gets the bit reversed copy erroneously with error at same error location (5th bit from left). The receiver will receive as: 10100100 (second copy).

(2) Say the original packet is, 01. Say on first transmission the receiver receives the packet as 00 (call it first copy). Receiver requests for retransmission. Transmitter retransmits a bit reversed copy as: 10. Say receiver gets the bit reversed copy erroneously with same error location. Thus receiver will receive as: 11 (2nd copy).

(3) Say the original packet is, 11111111. Say on first transmission the receiver receives the packet as 11110111 (call first copy). Receiver requests for retransmission. Transmitter retransmits a bit reversed copy as: 11111111. Say receiver gets the bit reversed copy erroneously with same error location.

Thus receiver will receive as: 11110111 (2nd copy).

Receiver will now perform correction operation as below:

The receiver reverses the second copy bit wise. In the eg:(1), we get second copy on reversing as 00100101. Now receiver does the correction as in PC with reversed second copy and first copy. In the example, XOR of first and reversed second copy will result in 00011000. Thus now application of brute force bit inversion on 4th & 5th bit will correct the error and it will require on average 2 trials only.

In the eg:(2), XOR operation will result 11. Brute force bit inversion scheme be employed to correct.

In eg:(3) correction is not possible. This is because error is exactly at the middle bit of packet. (The packet is of 9 bits. Middle bit is 5th bit from both ends.) Bit reversion does not change its position. Thus so long packet is not of odd number of bits, the proposed technique will work certainly.

The proposed scheme will be able to correct single bit errors by using two consecutive erroneous packets when error occurs at same location, because the packet reversing changes the bit position as: i th bit from the right of original packet of k bits to $(k-i+1)$ th bit in reversed packet for $i=1$ to k .

The proposed scheme may be called Packet Reversed Packet Combining (PRPC) scheme. PRPC will correct errors occurring at the same locations of all erroneous packets of all packets made of even numbers of bits. PRPC will also do correction of all bit error locations except exactly the middle bit of all packets made of odd number of bits. These corrections are not possible with PC and possible with MPC but with more processing and steps as required in [3]. PRPC is practical proposition as because the packet size is mostly in units of bytes that make them even number of bits.

B. MPC

In the MPC technique, on getting a retransmission call from the receiver the transmitter sends i ($i>1$) copies of the requested packet [4].

Receiver on getting i copies make a pair-wise XOR ed to locate error positions. For example if $i=2$, we have three copies of the packet (Copy-1= the stored copy in receiver's buffer, Copy-2= one of the retransmitted copies, Copy-3= another retransmitted copy) and three pairs for XOR operation are:

Copy-1 and Copy-2, Copy-2 and Copy-3 Copy-3 and Copy-1

Assume that a transmitted packet 10100011 received erroneously at receiver on first transmission is: Copy-1=10101011,

And subsequently retransmitted two copies received at receiver are Copy-2=10101111 and Copy-3=10100001.

Under XORed operation of combination as below we have:

Copy-1 XORed Copy-2 (say, C_{12})=00000100 (one bit in error), Copy-2 XORed Copy-3 (C_{23})=00001110 (three bits in error), Copy-3 XORed Copy-1 (C_{31})=00001010 (two bits in error).

The bit inversion process is done by ascending order of number of bits in error by XOR operation.

C. CRC Codes

A better method to detect errors in a large block of data involves the use of CRC codes. The hardware required to implement a CRC code is slightly more complicated than other error detection codes but it has an error detection effectiveness greater than 99.9%. The CRC is calculated by dividing the message block by a generator polynomial. The quotient is discarded and the remainder is appended to the end of the message block for transmission. At the receiving end, the transmitted block with the CRC is divided by the same generator polynomial. Once again the quotient is discarded and the remainder is checked to determine if any errors occurred. A remainder of zero indicates that no errors were detected. Since the CRC is only an error detecting code, the position of an error in the received message cannot be determined. CRC codes are used in communication protocols that use automatic repeat request (ARQ). Here we have used CRC-16 Code as an EDC (Error Detection Code).

D. CRC-16

The CRC-16 polynomial is used for block containing characters that are eight bits in length. The generator polynomial for the CRC-16 is [5]:

$$G(x) = X^{16} + X^{15} + X^2 + 1$$

The circuit which implements this polynomial is shown below. The block of data is shifted into the data input line. After all the data has been shifted in, the remainder will be stored in the registers 0 through 15 [5].

One of the best uses for the redundancy in a multi-level code is to implement error checking and correcting. Like other error checking methods, the check bits are transmitted along with the message bits.

The ability to perform error correction at the receiving end without retransmission is called forward error correction (FEC). FEC is one of the best methods when the communication is simplex or broadcasted to many users.

E. Code Selection

We now consider criteria that must be taken into account when selecting a FEC scheme for any given application [6].

1. Probability of uncorrected errors- Since it is impossible for any coding scheme to detect all errors and correct them, it is important to choose coding schemes for which the probability of both undetectable and uncorrectable (but detectable) errors is minimized (or satisfies the application is under consideration).

2. Overhead- The FEC codes should add as little as possible overhead and maximize the code rate. However, increased code capability generally leads to lower code rate.
3. Complexity- The implementation complexity of the coding/decoding scheme which typically increases with increase in code length and its capability to detect and correct errors.

F. BCH Codes

BCH (Bose – Chaudhuri - Hocquenghem) Codes form a large class of multiple random error- correcting codes. They were first discovered by A. Hocquenghem in 1959 and independently by R. C. Bose and D. K. Ray-Chaudhuri in 1960. BCH codes are cyclic codes. The original applications of BCH codes were restricted to binary codes of length $2m-1$ for some integer m . These were extended later by Gorenstein and Zieler (1961) to the nonbinary codes with symbols from Galois field $GF(q)$ [7].

For any positive integers, $m \geq 3$ and $t < 2m-1$, there is a binary BCH code with the following parameters, (referred to as an (n, k, t) BCH code) [27]:

- (i) Block length: $n = 2m - 1$,
- (ii) Number of parity check bits: $n - k \leq mt$,
- (iii) Minimum distance: $d_{\min} \geq 2t + 1$.

Each binary BCH code (n, k, t) can correct up to t -bit errors, and thus it is also referred to as a t -error-correcting code.

G. The Binary Golay Code

The binary form of the Golay Code is one of the most important types of linear binary block codes. It is of particular significance since it is one of only a few examples of a nontrivial perfect code. A t -error-correcting code can correct a maximum of t errors. A perfect t -error correcting code has the property that every word lies within a distance of t to exactly one code word. Equivalently, the code has $d_{\min} = 2t + 1$, and covering radius t , where the covering radius r is the smallest number such that every word lies within a distance of r to a codeword [8].

The $(23, 12)$ binary code with $d_{\min} = 7$ has the largest minimum distance for any known code with the same values of n and k . Golay was in search of a perfect code when he noticed that $2^{11} = 2^{23-12}$ which indicated the possible existence of a $(23, 12)$ perfect binary code that could correct up to three errors. In 1949, Golay discovered such a perfect code, and it is the only one known capable of correcting any combination of three or fewer random errors in a block of 23 elements.

II. THROUGHPUT ANALYSIS

A. Previous Case

If the schemes are employed with Stop-and-Wait ARQ, the throughput for PRPC and PC will be respectively:

$$\eta_{\text{PRPC}} = [1 - P + k\alpha(1 - \alpha^{k-1})] / (1 + 2a) \quad (1)$$

$$\eta_{\text{PC}} = [1 - P + k\alpha^3(1 - \alpha^{k-1})] / (1 + 2a) \quad (2)$$

where, α = bit error rate, P = packet error probability,

k = packet size,

a = (propagation time) / (transmit time)

If the schemes are employed with GBN, the respective throughput will be:

$$\eta_{\text{PRPC}} = (1 - p_1) / (1 + (N-1)p_1) \quad (3)$$

$$\eta_{\text{PC}} = (1 - p_2) / (1 + (N-1)p_2) \quad (4)$$

where, N = window size

$$p_1 = 1 - (1 - \alpha)^k - k\alpha(1 - \alpha)^{k-1}$$

$$p_2 = 1 - (1 - \alpha)^k - k\alpha^3(1 - \alpha)^{k-1}$$

As per eqs: (1-2 and 3-4) the gain of PRPC over PC is by a factor of $k\alpha(1 - \alpha^2)(1 - \alpha)^{k-1}$. This suggests the superiority of PRPC over PC over a range of k & α .

For a fixed α , the maximum gain is obtained when: $(k\alpha(1 - \alpha)^{k-1}) / dk = 0$

$$\text{or, } k = -1 / [\alpha(1 - \alpha^2) \ln(1 - \alpha)] \quad (5)$$

Eqn (5) shows that both at high and low bit error rate, gain is high with large packet size. With high packet size, the probability of packet in error will be high at all bit error rate conditions. But as PRPC will correct errors that are not possible to be corrected with PC, PRPC will offer higher throughput than that of PC. For fixed packet size, again the PRPC will be more efficient with high bit error rate as packet error probability increases with bit error rate. To get an idea of operational range for k using Taylor's series in eqn (5), if α is small, gives:

$$k \sim 1 / [\alpha^2(1 - \alpha^2)] \sim 1 / \alpha^2 \quad (6)$$

when, $\alpha = 10^{-2}$, the size of the packet that will provide maximum gain is about 10,000 bits. This is a practical size in network communication.

When for fixed packet size, we proceed to find the bit error at which the maximum gain is achievable, we find from:

$$d(k\alpha(1 - \alpha)^{k-1}) / dk = 0,$$

when k = fixed and α is variable that:

$$4\alpha / [1 - \alpha] = k + 1 \quad (7)$$

For any size of k , we find from eq (7), the required $\alpha \sim 0.999$. This should be the case as maximum gain is obtained when either k is very high or, bit error rate is very high the conditions at which packet error probability is high. The superiority of the PRPC over PC may be quantified with application of multi-nomial probability. The application of multinomial probability gives the probability of occurrence of single bit error consecutively twice at the same location as:

$$P_m = 2! / 0! * \{ {}^nC_1 \alpha (1 - \alpha)^{n-1} \}^2 \quad (8)$$

The gain of PRPC will be measured by P_{mas} :

$$\text{Gain of PRPC} = 1 / [1 - P_m] \quad (9)$$

PRPC as proposed has a limitation. When the error occurs in i th position from right in one erroneous copy and $(k-i+1)$ th position from right in the another erroneous copy, error correction will not be possible.

PC has other limitations. One of the important unaddressed limitations is as follows. The throughput is basically the inverse of the number of times, N a packet on average needs transmission before being correctly received by the receiver. For basics S/W ARQ and Chakraborty's scheme, N is given as:

$$\text{Basic S/W ARQ: } N = 1 / (1 - \alpha)^k$$

Chakraborty's ARQ with PC [3]:

$$N_{ac} = 1.(1-p) + 2.p(1-p) = (1-p)(1+2p) \sim 1+p$$

Where, p is the probability that the packet is received with single bit error, i.e

$$p = k \alpha (1-\alpha)^{k-1}$$

We need that, $N_{ac} \leq N$

This means that:

$$\{(1-\alpha)^k + k \alpha (1-\alpha)\} \leq 1 \quad (10)$$

This clearly shows that for a given α , Chakraborty's scheme is superior to the S/W ARQ for a limited k and vice-versa. Thus Chakraborty's scheme and the modified packet combining scheme are superior to S/W ARQ only to some limited extend of α for a given k. Such limitation as in eqn (10) does not exist for PRPC. In comparison of the limitations and advantages, it is clearly evident that PRPC is superior to PC.

(i)PRPC- Throughput of all ARQ techniques depend on the average number (n) of times a packet needs transmission (including retransmission) for successful reception by the receiver. As n decreases, throughput increases, and when n=1 throughput is 100%. We propose to study the gain of PRPC and MPC over the normal stop and wait ARQ. If gain is achieved, it will be equally true for other ARQ schemes namely GBN and SRQ. The gain will be measured by the parameter, n as defined.

In normal stop and wait ARQ: $n_{sw} = 1 / (1-P)$ (11) Where, P=Packet error probability = $1-(1-\alpha)^k$ In PRPC, all single bit errors will be corrected. The probability that a packet is with single bit error is:

$$P1 = {}^kC_1 \alpha (1-\alpha)^{k-1} \quad (12)$$

Thus the probability of packet in error except single bit error is: $P-P1$ (13)

In PRPC, for correction of single bit error, twice a packet is transmitted: first original and next reversed. Thus in PRPC when implemented in stop and wait ARQ protocol, the average number of times, n_{prpc} a packet needs transmission (including retransmission) for successful delivery is:

$$n_{prpc} = (1-P1)/(1-(P-P1)) + 2.P1 \quad (14)$$

First part of right hand side of eq:14 is for correction in normal stop and wait ARQ for bit errors other than single bit error, and second part is for PRPC in correcting single bit error.

Cases:

When $P=P1=0$ (no error case, result must be 1 packet per success) we got that as $n_{prpc}=1$;

When $P=P1=1$ (all single bit error, result must be 2 packets per success as in PRPC), we got that as

$$n_{prpc} = 2;$$

When $P=1$ & $P1 \neq 1$ (always packet is in error but error is not always single bit error, result must be towards infinite packets per success), we got that as

$$n_{prpc} \rightarrow \infty.$$

(ii) MPC with PRPC-

We propose MPC operation, which is a combined scheme of MPC and PRPC, called MPC with PRPC. This scheme is proposed with a view to correct both single bit error (with PRPC) and double bit error with (MPC) at receiver by

erroneous copies.

In proposed scheme, when a negative acknowledgement is received, transmitter will transmit two copies one in PRPC mode and another copy in original form of the original packet. Then in MPC with PRPC, up to double bit errors will be corrected at receiver. The probability of packet in error except with single bit error and double bit error is:

$$P - P1 - P2$$

Where, $P = 1-(1-\alpha)^k$ k= Packet size(bits)

P1 is the probability of packet with single bit error,

$$P1 = {}^kC_1 \alpha (1-\alpha)^{k-1}$$

α = Bit Error Rate (BER)

P2 is the probability of packet with double bit error, and $P2 = {}^kC_2 \alpha^2 (1-\alpha)^{k-2}$

For correction up to double bit error, thrice a packet is transmitted: first original and next two copies.

Thus in MPC with PRPC when implemented in stop and wait protocol, the average number of times, n_{mpc} a packet needs transmission(including retransmission) for successful delivery is:

$$n_{mpc} = (1-P1-P) / ((1-(P-P1-P2)) + 3.(P1 + P2)) \quad (15)$$

First part of right hand side of eq:15 is for correction in normal stop and wait ARQ for bit errors other than single bit and double bit error, and second part is for MPC with PRPC correcting upto double bit error. Thus the gain of MPC with PRPC if any over PRPC is:

$$\text{Gain}_{mpcprpc} = n_{prpc} - n_{mpc}$$

The percentage gain is then:

$$\text{Gain}_{mpcprpc} \% = [n_{prpc} - n_{mpc}] * 100 / n_{prpc}$$

(iii) MPC with PRPC over PRPC-

PRPC corrects all single bit error in packet. MPC with PRPC corrects up to double bit error in packet. Then the probability gain in correcting packet by MPC with PRPC over PRPC:

$$\text{Gain \%} = P2/P1 * 100 \quad (16)$$

B. Proposed Case

To combat high bit error rate with Error Detection Code (EDC) in Conventional PRPC scheme, in place of Error Correction Codes (ECC). But no conclusive result is found to justify the idea in terms of throughput. The investigation is an attempt to explore the basic gap.

(i) Modified MPC with PRPC over Conventional PRPC with ECC-

We propose MPC operation, which is a combined scheme of MPC and PRPC, called MPC with PRPC in a Modified way correct both single bit error (with PRPC) and double bit error with (MPC) at receiver by erroneous copies. In proposed scheme, when a negative acknowledgement is received, transmitter will transmit two copies one in PRPC mode and another copy in original form of the original packet. Then in Modified MPC with PRPC, the probability of single bit as well as double bit errors will be corrected at receiver is as follows: If P1 and P2 are the single bit error and double bit error, then the receiver will acknowledge without single bit and double bit error is:

$$P1*P1 + P1*(1-P1) + P2*(1-P1) + P2*P1 +$$

$$P2*P2 + P2*(1-P2) = P1 + 2*P2 \quad (17)$$

The probability of packet in error except with single bit error and double bit error is:

$$P'' = P - P1 - P2$$

Where, P1 and P2 are the probability of packet with single bit and double bit error respectively,

$$P1 = {}^kC_1 \alpha^1 (1 - \alpha)^{k-1}$$

$$P2 = {}^kC_2 \alpha^2 (1 - \alpha)^{k-2} \text{ and}$$

$$\alpha = \text{Bit Error Rate (BER)}$$

Thus in *Modified MPC* with PRPC when implemented in stop and wait protocol, the average number of times, n_{mmpc} a packet needs transmission (including retransmission) for successful delivery is:

$$n_{(\text{MMP}+\text{PRPC})} = [(P1 + 2.P2) + (P''/(1-P''))] \quad (18)$$

First part of right hand side of eq:18 is for *Modified MPC* with PRPC correcting up to double bit error and second part is for correcting in Normal Stop and Wait ARQ with bit errors other than single bit and double bit error. PRPC corrects all single bit error in packet. *Modified MPC* with PRPC corrects up to double bit error as well as with single bit error.

Then the probability gain in correcting packet by *Modified MPC* with PRPC over conventional PRPC is:

$$\text{Gain}_{\text{mmpcprpc}} \% = (P1 + 2*P2)/P1 * 100 \quad (19)$$

(ii) *Throughput efficiency of Conventional PRPC Scheme in Normal S/W ARQ-*

The Throughput of PRPC in Normal S/W ARQ with single bit error is:

$$(2.P1 + (P'/(1-P'))) \quad (20)$$

where, $P' = P - P1$

The first part is for PRPC in correcting single bit error, second part is for Normal S/W ARQ other than single bit error.

Now Coding Efficiency is:

$$k (\text{Packet Size in Bits}) / (k + c);$$

where, c is the Check Bits

Here we are using CRC-16 as an Error Detection Code (EDC). So the Throughput efficiency will be:

$$(\text{Throughput}) * (\text{Coding Efficiency})$$

i.e:

$$\text{Throughput}_{\text{eff}}_{\text{S/W}} = (2.P1 + (P'/(1-P')))*(k/(k+16)) \quad (21)$$

(iii) Throughput efficiency of *Modified MPC+PRPC* scheme in Normal S/W ARQ-

If P, P1 and P2 are the packet error, single-bit error and double-bit error probabilities, then the probability without single and double-bit error will be

$$P'' = P - (P1 + P2),$$

Where, $P = 1 - (1 - \alpha)^k$, k = Packet size(bits)

$$P1 = {}^kC_1 \alpha (1 - \alpha)^{k-1}$$

$$\alpha = \text{Bit Error Rate (BER)}$$

$$P2 = {}^kC_2 \alpha^2 (1 - \alpha)^{k-2}$$

So, for this scheme in S/W ARQ system, the throughput will be, from eq: (18)

$$n_{(\text{MMP}+\text{PRPC})} = [(P1 + 2.P2) + (P''/(1-P''))] \quad (22)$$

From eq(22) we can get 3 cases as in below:

(i) When $P'' = 0$, $P1 = 1$, $P2 = 1$, then $n = 3$

(ii) When $P'' = 1$, $P1 = 0$, $P2 = 0$, then $n = \infty$

(iii) When $P'' = 0$, $P1 = 1$, $P2 = 0$, then $n = 1$

Now Coding efficiency = $k / (k + c)$, where, c is the check-bits.

In eq (21), we have used CRC-16.

Using BCH (127, 120, 1) Code in eq: 22 we are getting the Coding efficiency = $k / (k / 120 * (127 - 120) + k + 1)$

$$= (120*k / (127*k + 120))$$

So the Throughput efficiency of *Modified MPC+ PRPC* scheme will be:

$$\text{Throughput}_{\text{eff}}_{(\text{MMP}+\text{PRPC})}$$

$$= [(P1 + 2.P2) + (P''/(1-P''))] * (120*k / (127*k + 120)) \quad (23)$$

Using BCH (127, 64, 10) Code in eq: 22 we are getting the Coding efficiency = $k / (k / 64 * (127 - 64) + k + 10)$

$$= (64*k / (127*k + 640))$$

So the Throughput efficiency of *Modified MPC+ PRPC* scheme will be:

$$\text{Throughput}_{\text{eff}}_{(\text{MMP}+\text{PRPC})}$$

$$= [(P1 + 2.P2) + (P''/(1-P''))] * (64*k / (127*k + 640)) \quad (24)$$

By using Golay (23, 12, 3) Code in eq:22 we are getting the Coding efficiency = $k / (k / 12 * (23 - 12) + k + 3)$

$$= (12*k / (23*k + 36))$$

So the Throughput efficiency of *Modified MPC+ PRPC* scheme will be:

$$\text{Throughput}_{\text{eff}}_{(\text{MMP}+\text{PRPC})} = [(P1 + 2.P2) + (P''/(1-P''))] * (12*k / (23*k + 36)) \quad (25)$$

III. DISCRETE EVENT SIMULATION SIMULATION PARAMETERS

Simulation Parameters	Values
Simulation time	10 s
Uplink frame	0.5 ms
Downlink frame	0.5 ms
Duplex mode	TDD
Application profile	CBR (100 bytes, 0.2 ms)
Agent profile	UDP
ARQ _ WINDOW _ SIZE	500
PHY rate	1 Mbps

IV. SIMULATED RESULTS AND JUSTIFICATION

Fig: 8 Throughput with respect to Pause time, for Modified MPC+PRPC scheme without ECC and PRPC in Normal S/W with CRC-16 scheme.

Fig: 9 Throughput with respect to Pause time, for Modified MPC+PRPC scheme without ECC, Modified MPC+PRPC scheme with (127, 64, 10) BCH Code/ (23, 12, 3) Golay code and Normal S/W with CRC-16

Fig:10 Throughput with respect to Pause time, for Modified MPC+PRPC scheme without ECC, Modified MPC+PRPC scheme with (127, 120, 1) BCH Code and Normal S/W with CRC-16.

Fig 11: Throughput with respect to large packet size (k), for Modified MPC + PRPC scheme without ECC and PRPC Scheme in Normal S/W with CRC-16.

Fig: 12 Throughput with respect to large packet size (k), for Modified MPC + PRPC scheme without ECC, Modified MPC+PRPC scheme with (23, 12, 3) Golay Code and PRPC Scheme in Normal S/W with CRC-16.

Fig: 13 Throughput with respect to large packet size (k), for Modified MPC + PRPC scheme without ECC, Modified MPC+PRPC scheme with (127, 120, 1) BCH Code and PRPC Scheme in Normal S/W with CRC-16.

Fig:14 Throughput with respect to medium packet size (k), for Modified MPC + PRPC scheme without ECC and PRPC Scheme in Normal S/W with CRC- 16

Fig: 15 Throughput with respect to medium packet size (k), for Modified MPC+PRPC scheme without ECC, Modified MPC+PRPC scheme with (23, 12, 3) Golay Code and PRPC Scheme in Normal S/W with CRC- 16.

Fig:16 Throughput with respect to medium packet size (k), for Modified MPC + PRPC scheme without ECC, Modified MPC+PRPC scheme with (127, 120, 1) BCH Code and PRPC Scheme in Normal S/W with CRC- 16

V. JUSTIFICATION OF THE SIMULATED GRAPHS

In fig: 8, we see that the *Modified MPC+PRPC Scheme* in Normal S/W without ECC shows better simulating result

than PRPC Scheme in Normal S/W with CRC-16 with respect to pause time.

In fig:9, we see that the *Modified MPC+PRPC Scheme* in Normal S/W without using ECC shows better simulating result than the *Modified MPC+PRPC Scheme* in Normal S/W with using BCH (127, 64, 10) Code / (23, 12, 3) Golay Code and PRPC Scheme in Normal S/W with CRC-16 Code with respect to pause time. Because, here, in BCH Code an error will be corrected up to 10-bits and for Golay Code up to 3-bits. In fig: 10, we see that the *Modified MPC+PRPC Scheme* in Normal S/W without using ECC shows better simulating result than the *Modified MPC+PRPC Scheme* in Normal S/W with using BCH (127, 120, 1) Code and PRPC Scheme in Normal S/W with CRC-16 Code with respect to pause time. But here BCH Code will be corrected only 1-bit so it shows much better result than other two ECCs in fig:8 and fig:9.

In fig: 11, we see that the throughput is very high with respect to large packet size (20, 000 bits) for *Modified MPC+PRPC Scheme* in Normal S/W without using ECC than PRPC scheme in Normal S/W with CRC-16.

In fig: 12, fig: 13, we see that the throughput is very high with respect to large packet size for *Modified MPC+PRPC Scheme* in Normal S/W without using ECC than *Modified MPC+PRPC Scheme* in Normal S/W with using BCH (127, 64, 10) Code / (23, 12, 3) Golay Code, BCH (127, 120, 1) Code, PRPC Scheme in Normal S/W with CRC-16. Among this three figures, using BCH (127, 120, 1) shows better throughput than the rest of other Codes, since, here BCH Code will be corrected error only 1-bit, whereas, BCH (127, 64, 10) Code will be corrected error up to 10-bits, (23, 12, 3) Golay Code'll be corrected up to 3-bits.

In fig: 15, fig: 16, we see that the throughput is very high with respect to medium packet size (1000 bits) for *Modified MPC+PRPC Scheme* in Normal S/W without using ECC than *Modified MPC+PRPC Scheme* in Normal S/W with using BCH (127, 64, 10) Code / (23, 12, 3) Golay Code, BCH (127, 120, 1) Code, PRPC Scheme in Normal S/W with CRC-16. Among this two figures, using BCH (127, 120, 1) shows better throughput than the rest of other Codes due to error correcting ability of 1-bit.

In fig. 14, we see that the throughput is very high with respect to medium packet size (1000 bits) for *Modified MPC+PRPC Scheme* in Normal S/W without using ECC than PRPC scheme in Normal S/W with CRC-16.

VI. CONCLUSION AND FUTURE WORK

Throughput comparison of *Modified MPC+PRPC Scheme* in Normal S/W with and without using ECC and conventional PRPC scheme in Normal S/W with CRC-16, with respect to medium as well as large packet size for high and low BER (Bit Error Rate), with respect to pause time in ARQ protocol for Mobile Communication and Networks have been investigated in this thesis.

Modified MPC+PRPC Scheme always shown better result but the throughput does not increase when we are using ECC's in our proposed scheme than without using ECC, though it provides better reliability than the previous schemes.

We propose to study an hybrid approach of achieving reliability by combining BEC with FEC (Forward Error Correction) with Turbo Code. Our initial theoretical work and practical implementation with one bit error correction code give encouraging results

REFERENCES

- [1] Shyam S Chakraborty, " An ARQ scheme with packet combining", IEEE Communication Letters, Vol 2, no:7, pp200-202, july' 1995.
- [2] C.T.Bhunia, "Modified Packet Combining Scheme"Proc. ICITA, IEEE Comp.Soc, Vol2.
- [3] C.T.Bhunia, SMIEEE, "Exact Analysis of Packet Reversed Packet Combining Scheme & MPC scheme; & a Combined scheme " IEEE Comp.Soc.
- [4] Prof. C.T. Bhunia, InformationTechnology Network & Internet, NEW AGE INTERNATIONAL LIMITED, PUBLISHERS
- [5] Bruce Peterson for Virtium Technology, Data Coding and Error Checking Techniques.
- [6] Eytan Modiano, Communication Systems Engineering Lectures 15: ARQ Protocols.
- [7] Hank Wallace, Error Detection and Correction Using BCH Codes.
- [8] Hang Liu, Hairuo Ma, Magda El Zarki and Sanjay Gupta, Error control schemes for networks: An overview