Available at: http://www.ictp.it/~pub_off IC/2006/046

United Nations Educational Scientific and Cultural Organization
and
International Atomic Energy Agency

THE ABDUS SALAM INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

PACKET REVERSED PACKET COMBINING SCHEME

C.T. Bhunia[1]
Haldia Institute of Technology, PO, Haldia, 721657, India
and
The Abdus Salam International Centre for Theoretical Physics, Trieste, Italy.

Abstract

The packet combining scheme is a well defined simple error correction scheme with erroneous copies at the receiver. It offers higher throughput combined with ARQ protocols in networks than that of basic ARQ protocols. But packet combining scheme fails to correct errors when the errors occur in the same bit locations of two erroneous copies. In the present work, we propose a scheme that will correct error if the errors occur at the same bit location of the erroneous copies. The proposed scheme when combined with ARQ protocol will offer higher throughput.

MIRAMARE – TRIESTE
July 2006

[1] Senior Associate of ICTP. ctbhunia@vsnl.com

I. INTRODUCTION

Packet Combining (PC) scheme for correction of bit error using two successive erroneous copies of packet 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 the transmitter to retransmit the packet. The correction process proposed is the brute force bit by bit inversion in the located bit error positions and FCS check. The scheme of Chakraborty fails: (i) when the bit error locations in both the erroneous copies are the same and (ii) when multiple bit errors occur, as then the application of brute force bit inversion for correction will be huge and complex. For n bits in error (n>1), on average 2^{n-1} trails of attack are required.

A scheme called Modified Packet Combining (MPC)[2,3] was reported to tackle the multiple bit errors in the received erroneous copies of a packet. MPC cannot also tackle errors when they occur at the same location until an odd number of erroneous copies are available and that too only when transmitted bit 0 is converted to 1 in all copies in the same location but not when transmitted bit 1 is converted to 0 in all copies in the same location. This is exactly the same in PC. In the present study we propose a scheme to tackle the error correction when the bit error locations are same in both the erroneous copies under PC or MPC schemes.

II. NEW BASIC IDEA

Any error correction scheme for networks will mainly address the correction for single bit error. When the 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 a single bit error. The type of links available except very long haul wireless satellite links for networks, does guarantee bit error rate of 10^{-2} or less. This makes the PC a powerful scheme for networks. The only investigation left for the PC is how to correct an error when bit locations of the error in two erroneous copies used in PC are the same. The current idea leads to this investigation. The basic idea is that when the receiver 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.

Example: 1. Say the original packet is, 00110101. Say on the first transmission the receiver receives the packet as 0011**1**101 (call it first copy) (error location is marked in bold, 5th bit from left). Receiver requests for retransmission. Transmitter retransmits a copy with bit reversed as: 10101100 (bit wise reversed copy of original packet, LSB of original packet is now MSB of bit reversed packet and vice versa). Say receiver gets the bit reversed copy erroneously with an error at the same error location (the 5th bit from left). Thus the receiver will receive it as: 1010**0**100 (call it second copy).

2. Say the original packet is, 01. Say on the first transmission the receiver receives the packet as 0**0** (call it first copy) (error location is marked in bold). Receiver requests for retransmission. Transmitter retransmits a bit reversed copy as: 10 (bit wise reversed copy). Say receiver gets the bit reversed copy erroneously with same error location. Thus the receiver will receive it as: 1**1** (call it second copy).

3. Say the original packet is, 111111111. Say on first transmission the receiver receives the packet as 1111**0**1111 (call it first copy) (error location is marked in bold). Receiver requests for retransmission. Transmitter retransmits a bit reversed copy as:

1<u>11111111</u> (bit wise reversed copy). Say receiver gets the bit reversed copy erroneously with the same error location. Thus the receiver will receive it as: <u>1111**0**1111</u> (call it second copy).

[NOTE: To mark the bit reversed copy, we have underlined the copy. This is for illustration purpose.]

The receiver will now perform the correction operation as below:

I. The receiver reverses the second copy bit wise. In the example 1, we get a second copy on reversing as 0011**0**101. Now the receiver does the correction as in PC with a reversed second copy and first copy. In the example, XOR of the first and reversed second copy will result in 00011000. Thus now the application of brute force bit inversion on 4^{th} and 5^{th} bit will correct the error and it will require on average 2 trails only. In example 2, XOR operation will result as 11. Brute force bit inversion scheme will be employed to correct.

II. In example 3, correction is not possible. This is because the error is exactly at the middle bit of the packet. (in examples 1 and 2,each of the packets is of 8 bits. In example 3, the packet is of 9 bits. The middle bit is the 5^{th} bit from both ends.) Bit reversion does not change its position. Thus so long as the packet is not of an odd number of bits, the proposed technique will certainly work.

The proposed scheme will be able to correct single bit errors by using two consecutive erroneous packets when error occurs at the 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 the 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 correct 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 but possible with MPC with more processing and steps as required in[2]. PRPC is a practical proposition because the packet size is mostly in units of bytes that make them an even number of bits.

III. THROUGHPUT ANALYSIS

If the schemes are employed with stop and wait ARQ[4,5], the throughput for PRPC and PC will be respectively:

$$\eta_{PRPC} = \frac{1 - P + k\alpha(1-\alpha)^{k-1}}{1 + 2a} \qquad (1)$$

$$\eta_{PC} = \frac{1 - P + k\alpha^3(1-\alpha)^{k-1}}{1+2a} \tag{2}$$

where α = bit error rate, P=packet error probability, k=packet size, $a = \frac{propagation\ \ time}{transmit\ \ time}$

If the schemes are employed with GBN[4,5], the respective throughput will be:

$$\eta_{PRPC} = \frac{1-p_1}{1+(N-1)p_1} \tag{3}$$

$$\eta_{PC} = \frac{1-p_2}{1+(N-1)p_2} \tag{4}$$

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)-(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 and α. For a fixed α, the maximum gain is obtained when:

$$\frac{d\left(k\alpha(1-\alpha)^{k-1}\right)}{dk} = 0$$

Or $k = \frac{-1}{\alpha(1-\alpha^2)\ln(1-\alpha)}$ (5)

Eq.(5) shows that at both 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 the operational range for k for a maximum gain of PRPC over PC we approximate eq.(5) by the Taylor Series expansion of the ln term when α is small. This gives:

$$k \approx \frac{1}{\alpha^2(1-\alpha^2)} \approx \frac{1}{\alpha^2} \tag{6}$$

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

For fixed packet size, we proceed to find the bit error rate at which the maximum gain is achievable, we find from

$\frac{d\left(k\alpha(1-\alpha)^{k-1}\right)}{dk} = 0$ when k =fixed and α is variable that:

$$\frac{4\alpha}{1-\alpha} = k+1 \tag{7}$$

For any size of k, we find from eq.(7), the required $\alpha \approx 0.999$. This should be the case as the maximum gain is obtained when either k is very high or the bit error rate is very high. These are the conditions at which packet error probability is high.

The superiority of PRPC over PC may be quantified with the application of multinomial probability. The application of multinomial probability gives the probability of occurrence of single bit error consecutively twice at the same location as:

$$P_m = \frac{2!}{0!} \times \left\{ \binom{n}{1} \alpha(1-\alpha)^{n-1} \right\}^2 \tag{8}$$

The gain of PRPC will be measured by P_m as:

$$\text{Gain of PRPC} = \frac{1}{1-P_m} \tag{9}$$

PRPC as proposed has a limitation. When the error occurs in the i^{th} position from the right in one erroneous copy and the $(k-i+1)^{th}$ position from the right in 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:

Basic S/W ARQ

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

"Chakraborty" ARQ with PC[3]:

$N_{ac} = 1.(1-p) + 2.p(1-P) = (1-p)(1+2p) \approx 1+p$ where p is the probability that the packet is received with the single bit error, i.e.

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

We need $N_{ac} \le N$

This means that

$$\{(1-\alpha)^k + K\alpha(1-\alpha)\} \le 1 \qquad (10)$$

This clearly shows that for a given α, Chakraborty's scheme is superior to the S/W ARQ for 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 extent of α for given k or limited k for given α. Such limitation as in eq.(10) does not exist for PRPC. In comparison to the limitations and advantages, it is clearly evident that PRPC is superior to PC.

IV. A PROTOCOL up to double bit error correction at receiver

The application of FEC (Forward Error Correction) at the receiver with error correction code has been studied in [6]. For the links and over a bit error rate that impairs packet maximum up to double bit errors (under moderate links and bit error rate below 10^{-3}, this is acceptable), we propose ARQ (that may be any one of the basic schemes of Stop-and-wait or Go Back N) as shown in fig. 1, where only NAks operations are shown .

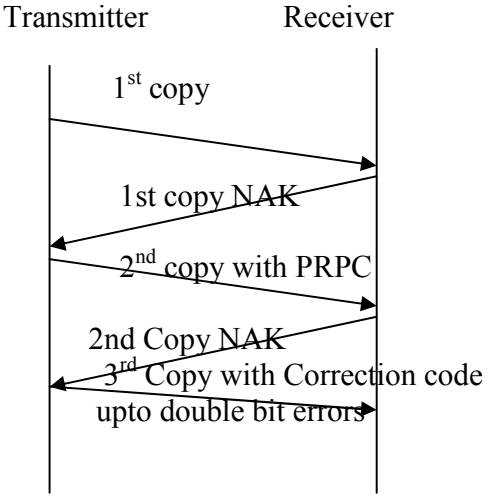

Fig. 1: A protocol for ARQ operation that can correct up to double bit errors at the receiver.

In fig. 1 we have shown the model for stop-and-wait ARQ, and this is extendable to the GBN model.

In the suggested protocol, the single bit error that may not be corrected at the PRPC stage will be definitely corrected at the next stage with correction code. The protocol is such that correction code be employed only when the PRPC operation gets a NAK. Thus for a correction code application three transmissions (one transmission, one PRPC test and one for test with error correction) are required. Similarly for the PRPC stage two transmissions are required. Under our assumptions, a packet may reach the receiver as (1) correct one, (2) erroneous with single bit error, and (3) erroneous with double bit error. Then the average number of packets required to be transmitted, including transmission and retransmission for successful transmission of a packet is:

$$N_{averagof\ \mathrm{mod}\,ele} = 1.(1-P) + 2.k\alpha(1-\alpha)^{k-1} + (2+\frac{k+r}{k})k(k-1)\alpha^2(1-\alpha)^{k-2}. \quad (11)$$

where r is the number of check bits per k bits for error correction code upto 2 bits. If stop and wait ARQ is made for only up to double bit error correction by retransmissions, then the average number of packets required to be transmitted, including transmission and retransmission for successful transmission of a packet is:

$N_{averagearqq} = \frac{1}{1-L}$ where

$$L = \left[P - k\alpha(1-\alpha)^{k-1} - k(k-1)\alpha(1-\alpha)^{k-2}\right] \quad (12)$$

When the value estimated in eq.(11) will be less than that of eq.(12), the proposed model will offer higher throughput. The choice of r or the error correction code therefore plays a vital role. The (15,7) code that corrects up to the double bit error will be a good choice as it offers higher coding efficiency compared to other double bit error correction codes and the error correction decoder for this code is just made of majority gate[7].

V. FUTURE RESEARCH

The comparison of the proposed scheme with PC, MPC and other related schemes may establish conclusively the superiority of PRPC. PRPC is a scheme simple as PC but with higher error correction capability. We propose to compare the techniques with some models in future works.

Acknowledgments. This work was done within the framework of the Associateship Scheme of the Abdus Salam International Centre for Theoretical Physics, Trieste, Italy.

REFERENCES

[1] Shayam S Chakraborty,, "An ARQ scheme with packet combining", IEEE Communication Letterss,Vol 2, No 7, pp200-202, July'1995

[2] Chandan T Bhunia, "Modified packet Co1bining Scheme"Proc. ICITA, IEEE Comp. Soc., Vol 2, pp 641-646, 2005

[3]Chandan T Bhunia, "Information Technology, Networks and Internet", New Age International Publishers, New Delhi, 2005

[4] Shyam S Chakraborty, "An Adaptive ARQ scheme with Packet Combining for time varying channels", IEEE Communication Letters, Vol 3, No 2, pp 52-54, Feb'1999

[5] Chandan T Bhunia, "ARQ-Review and Modifications", J IETE Tech Rev. Vol 18, pp381-401, Sept'2001

[6] Thomas Kostas and Scott Jordan, "Packet Erasure FEC on ARQ protocols", University of California, 544 D.

[7] William G Chambers, Basics of Communication and Coding, Oxford Science Publications, 1985, Ch 6.