IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501
Vol.4, No1, February 2014

Performance Analysis of Selective Reject ARQ with effect on Efficiency

Prof.Swati A. Kemkar
Department of B.Sc IT
Ramanand Araya DAV College .
Bhandup(e),Mumbai-400604
swateekemkar@gmail.com

Dr.(Prof.)Atul N.Kemkar
Department of EXTC,
SIESGST,
Nerul,Navimumbai
ankemkar@gmail.com

Abstract—We are using evaluating various ARQs on the basis of Efficiency. One parameter that have effect on Efficiency is Bit Error Rate. We are changing BER for Selectve Reject and for various window sizes trying to figure out optimum value of window size for range of BER.

Keywords-ARQ,FEC,NACK,CRC

Introduction

One of the important layers is the Data Link Layer. The two main functions of the data link layer are data link control and media access control. The first data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication. Important responsibilities of data link layer are Flow control and Error control. Collectively, these functions are known as Data Link control. There are two basic approaches to error control in digital communications: Forward error correction(FEC) and Automatic repeat request (ARQ)

In the FEC systems, parity-check bits are added to each transmitted message block to form a code word based on the error-correcting code that is being used. The receiver attempts to locate and correct the errors that it has detected in a received word. After the error-correction procedure, the decoded data block is delivered to the end user.
A decoding error occurs if the output of the decoder is a different codeword than the one that was originally transmitted. The FEC systems are designed for use in simplex channels, where information in one direction.
In an ARQ scheme, a high-rate error-detecting code is used together with some retransmission protocol. If the receiver detects errors in the received word, it generates a retransmission request, or a negative acknowledgement (NACK). If no errors are detected in the received word, the receiver sends a positive acknowledgement, called an ACK, to the transmitter. The most widely used error-detecting codes are the cyclic redundancy check (CRC) codes because of ease of implementation. Unlike the FEC systems, the ARQ schemes require the presence of a feedback channel. Error control allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In data link layer term Error control refers primarily to methods of error detection and retransmission. This implemented simply: Any time error is detected in an exchange, specified frames are retransmitted. This process is called Automatic Repeat Request (ARQ). Another continuous ARQ strategy, selective-repeat (SR) ARQ, is much more efficient than GBN, since only negatively acknowledged code words are retransmitted. After resending a negatively acknowledged codeword, the transmitter continues transmitting new code words in the transmitter buffer.
Whereas the GBN scheme automatically preserves the original order of the Code words, the receiver in the SR scheme must have some buffer space to store the correctly received code words that can not yet be released.
In the section 2 system model is explained. Channel model is explained in section 3.section 4 comprises of graphs and results. Concluding remarks are in section 5.

2. System Model:

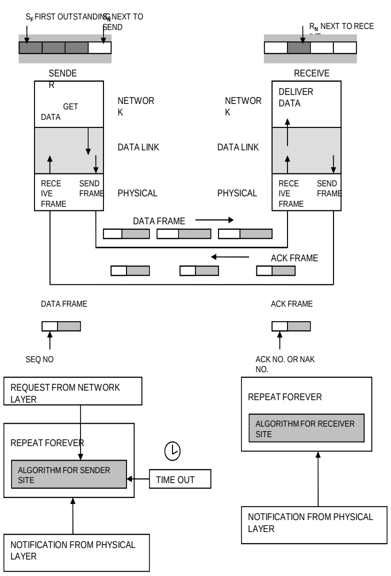


Figure:2 System model

In the figure 2 shown a transmitter block and receiver block for selective repeat ARQ scheme. The Selective Repeat Automatic ARQ does not resend N frames when just one frame is damaged; only the damaged frame is resent. The timer is set for all the outstanding windows and the frame whose timer is expired is resent. This protocol also uses NAK (Negative Acknowledgement).

The **Selective Repeat** protocol improves on the Go Back N protocol by having buffers on both the sending and receiving sides. This allows the sender to have more than one outstanding frame at a time and receiver to accept out of order frames and store them in its window.

Sender for *Selective Repeat* is only slightly modified from that for *Go Back N* .The Maintenance of buffers and logical timers is exactly the same. The only difference is that if a negative acknowledgement is received, the sender retransmits the corresponding frame identified by the nack. Other than this timeouts, loop iterations and retransmissions are all the same as *Go Back N* . This differs from *Go Back N* in that it retransmits only the frame for which a nack is received and not all subsequent frames. As the receiver keeps a window of frames only the timed out frame needs to be retransmitted and not the whole series.

The receiver busy waits until a frame arrives. If a timeout occurs or if a corrupt frame arrives, or if an out of sequence frame arrives, and a nack has not been sent yet then a nack is sent for the expected sequence number. If there is room in the receiver's buffer a packet is stored in the correct slot (sequence number) and the slot is flagged as used. Loop is run starting at buffer slot for expected sequence number. If this slot is full the packet is passed to the network layer, a flag is set to send an acknowledgement, buffer slot is reset to empty, the upper bound of the window is increased, and the lower bound is circularly incremented.

It then loops back to check the buffer slot for expected sequence number. The loop continues till the expected slot is empty. In this manner all buffered packets are passed to the network layer in order. If the flag that indicates whether an acknowledgement has to be sent or not is set then an acknowledgement is sent for the last correct in sequence frame received. Then we go back to the main busy waiting loop and start over again. The buffers on both ends consist of arrays of packets. The size of these arrays is 4, the range of sequence numbers used is 0-7. The selection criteria were the same as that used for *Go Back N* . The sender window's Lower Bound is represented by the Sf and the Upper Bound by the Sn. The receiver window's Lower Bound is represented by Rn. The Logical Timers are implemented in the same fashion as *Go Back N* . *Main loop* in the sender consists of while loop that gets and sends packets then checks for acknowledgements and other loop loads empty slots in the buffer with new packets. The ***acknowledgement loop*** clears all previous unacknowledged frames up to the acknowledgement received. If the frame is a nak instead of entering the loop it retransmits the requested frame. ***Timer loop*** updates the timers and checks for timeouts to retransmit.

Main Loop in the receiver consists of a busy wait loop to retrieve frames from the physical layer. ***Data transfer loop*** passes buffered packets in order to the network layer when the correct in sequence frame is received. Timeouts and bad frames result in naks being sent. Otherwise an acknowledgement is sent for the last correct in sequence frame received. This is necessary to keep the sender and receiver in synchronization when frames and acknowledgements are lost. The ***Selective Repeat*** protocol was difficult to implement. The sequence numbers need to be greater than the window size so that no overlap can occur in the window. This allows receiver and sender to be kept in synchronization even when frames and acknowledgements are lost at a very high rate. The buffering and acknowledgements allow this protocol to easily handle congestion, bad frames and lost frames. It was found that a much higher timeout value is needed than in ***Go Back N*** in order to reduce the number of frames sent. A lower timeout value results in too many frames timing out and being retransmitted unnecessarily, since the receiver maintains a buffer of frames and can send a nak for exactly the frame sequence number it needs. We have chosen 25 as a timeout value, and kept the from_physical_layer parameter to 100,000 because these two values worked the best for a wide range of error rates.

3.Channel model:

Most of the channel models considered in this work are discrete-time models, which are characterized by the values of bit error rate (BER) or packet error rate (PER). Depending on whether the time unit of the model is the transmission time of one bit or one packet, these models can be divided into bit-level and packet-level models.

The simplest discrete-time model is the memory less binary symmetric channel (BSC), which is also often referred to as the random-error channel. In a BSC, a bit is received in error with a certain probability (the BER), independently of all the other bits. As a result, the number of bit errors in a received n-bit packet is binomially distributed; if the BER is equal to ε, the PER is given by

$$Pe(n,\varepsilon) = 1 - (1-\varepsilon)^n.$$

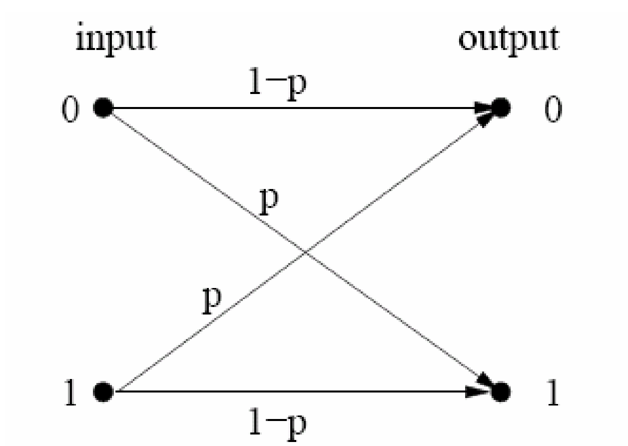


Figure 3.Binary Symmetrical Channel

In many practical channels, especially in the presence of fading, the bit errors are not statistically independent, but occur in bursts. They are called channels with memory, or burst-error channels. If FEC is used and the error-correcting code is designed to correct random errors, the channel errors can be made to look more random by using an interleaver after encoding the data block and a de-interleaver before decoding the received word. On the other hand, ARQ' schemes typically perform better with bursty errors than with random errors if the average error probability is the same.

A binary symmetric channel corrupts a binary signal by reversing each bit with a fixed probability. Such a channel can be useful for testing error-control coding. To model a binary symmetric channel, use the bsc function. The two input arguments are the binary signal and the probability, p.

4.Results and Graphs:

Throughput Efficiency

The most important performance measure for the ARQ schemes is the throughput efficiency, or simply the ***Throughput*** *(*η). The throughput is defined as "the ratio of the average number of information bits successfully accepted by the receiver per unit time to the total number of bits that could be transmitted per unit time". It can be noted that the throughput of an FEC scheme is a constant irrespective of the channel conditions, and it is equal to the rate of the error-correcting code.

A related performance measure, which we will call the packet throughput and denote by T, is defined as the average number of packets accepted successfully per one transmission. It is the inverse number of the average number of transmission attempts needed until a packet is received successfully. The difference between η and T is that in computing η, only the information bits are considered 'useful', and hence η represents the 'real' transmission efficiency. The quantities η and T relate to each other as follows:

$$\eta = (k/n)*T = (k/n)/E[X]$$

where k/n is the rate of the error-detecting code used by the ARQ scheme, and X is the random variable that represents the number of transmission attempts needed until a packet is received successfully. Naturally, the distribution of X depends on the channel (also the return channel) error statistics.

We have prepare signal model, Transmitter and Receiver model and Time function model in MATLAB. Performance measure for model is Throughput efficiency and performance parameter is BER and Window size.

Statistical Analysis

Window Size (sw)	Propagation time (sec)	Total transmission time (status_time) Sec	Probability of error (BER)	Efficiency (sw/total_frames)*100 (%)
1	1	2.2256	0	100
1	1	2.2256	0.0001	100
1	1	2.2256	0.001	100
1	1	2.2256	0.01	100

Window size 1

Window size 2

Window Size(sw)	Propagation time (sec)	Total transmission time (status_time) Sec	Probability of error (BER)	Efficiency (sw/total_frames)*100 (%)
2	1	3.425	0	100
2	1	3.4456	0.0001	100
2	1	3.4894	0.001	100
2	1	5.2836	0.01	83.33

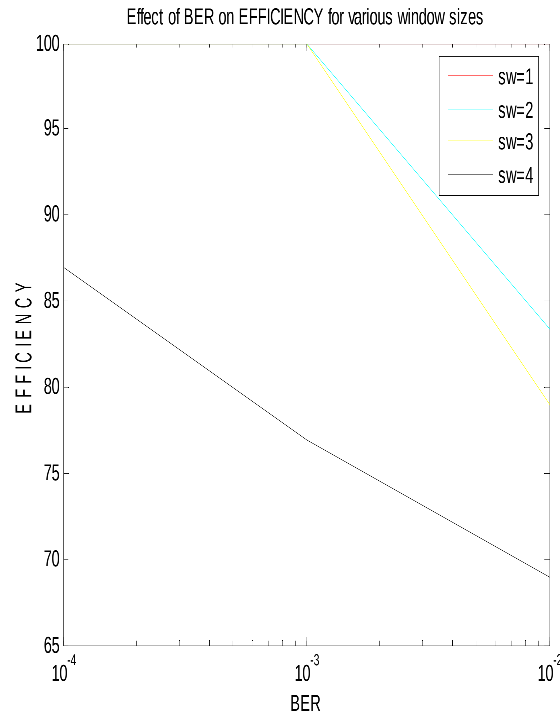
Window size 3

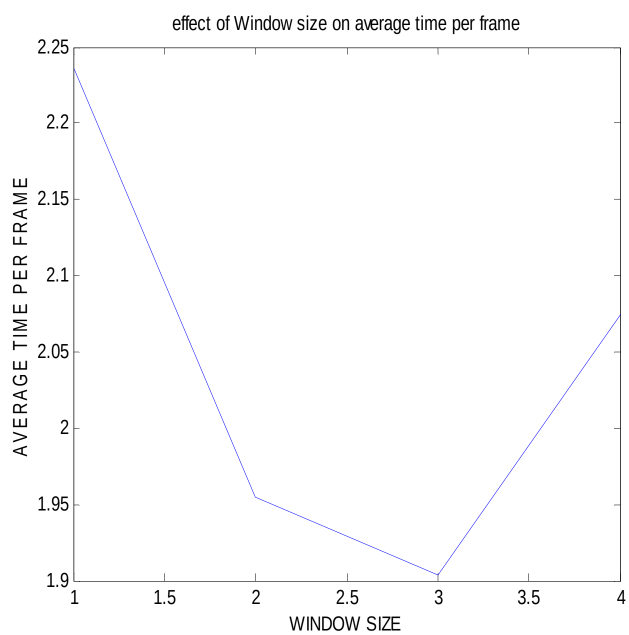
Win dow Size(sw)	Propag ation time (sec)	Total transmi ssion time (status_t ime) Sec	Proba bility of error (BER)	Efficiency (sw/total_fra mes)*100 (%)
3	1	4.4506	0	100
3	1	4.4976	0.0001	100
3	1	4.5088	0.001	100
3	1	9.3882	0.01	78.94

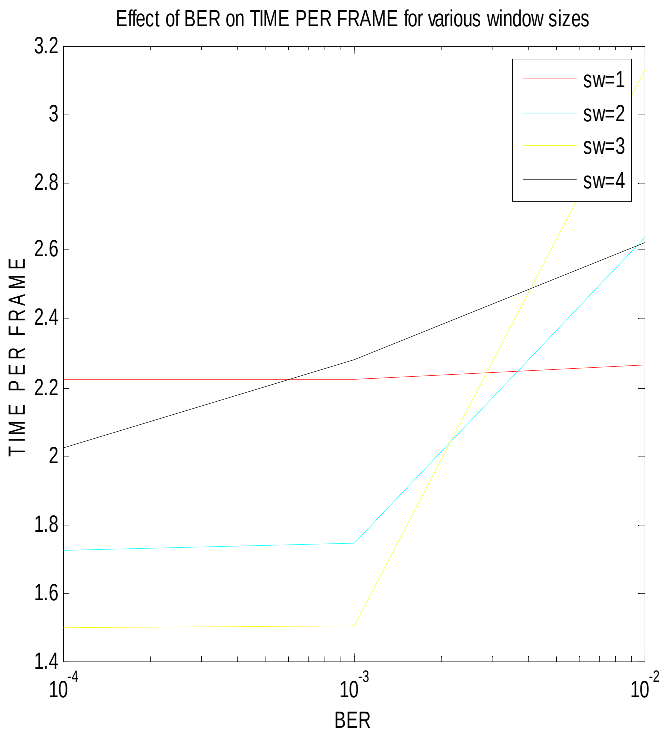
Window size 4

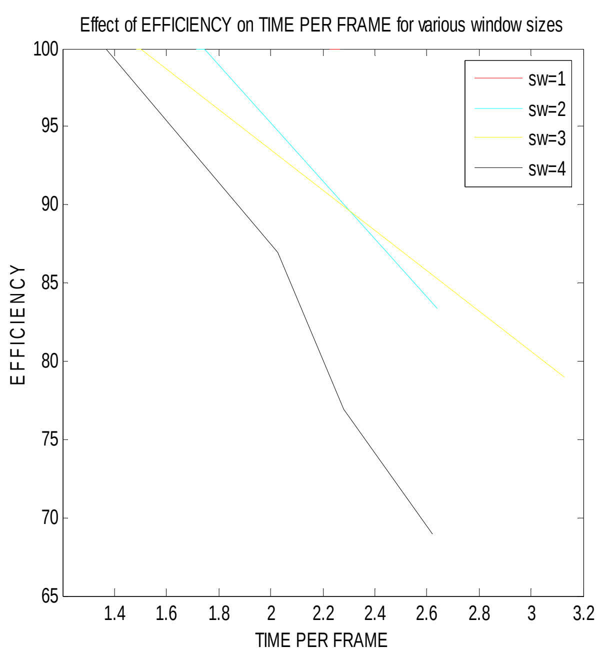
Win dow Size (sw)	Propa gation time (sec)	Total transm ission time (status _time) Sec	Proba bility of error (BER)	Efficiency (sw/total_fr ames)*100 (%)
4	1	5.478	0	100
4	1	8.0996	0.0001	86.96
4	1	9.1194	0.001	76.92
4	1	10.488	0.01	68.96

Graphical Analysis:









5. Conclusion:

We can conclude following aspects from the observations and graphs of Selective Reject.

Efficiency depends on Bit Error Rate and Window size.

Generally as BER increases efficiency decreases. Rate of fall of efficiency depends on Window size. As window size increases, retransmission increases, thus decreasing efficiency at faster rate.

Average time for frame, (considering BER over range 0.0001 to 0.01) goes on decreasing as window size increase, as more frames are transmitted and we get cumulative acknowledgement. Then it decreases since at higher BER number of retransmission increases.

Thus optimum window size is 3 for transmission of 7 frames.

6. References

[1] Masakazu Yoshimoto, Tetsuya Takine, Yutaka Takahashi, and Toshiharu Hasegawa, “Waiting Time and Queue Length Distributions for Go-Back-N and Selective-Repeat ARQ Protocols” IEEE Transactions on Communications, Vol. 42, No. 21314, February/ March/April 1994.

[2] Data communication and networking ,fourth edition by Behrouz A Forouzan.

[3] Data and computer communication by William stallings.

Authors Frofile:

Prof. Swati kemkar working as Asst.Prof in DAV college in the department of B.Sc IT at Mumbai.

Prof. Atul Kemkar is working as a Professor in EXTC Dept. at SIESGST Nerul.Navimumbai.