NEGATIVE ACKNOWLEDGEMENT-BASED ARQ PROTOCOL EFFICIENCY IN UNICAST DATA NETWORKS

Mahdi H. A. Ahmad, Ph.D
University of Taizz, Faculty of Engineering and IT
Eng.dr_mahdi@gawab.com

Abstract

Automatic Repeat reQuest (ARQ) protocols are essential in modern data communication systems to guarantee reliable transmission over imperfect physical links. Due to the large variation in implementation complexity and performance, depending on the link quality, performance analysis is very important for the right choice of protocol for a particular implementation. This paper models different error rate data (wireless, wireline) links and analyses the performance of positive acknowledgment based (ACK) Stop-and-Wait, Go-Back-N and Selective-Repeat protocols and negative Acknowledgement based (NAK) Selective-Repeat protocol.

INTRODUCTION

When data are transmitted over channels with high error probabilities, mechanisms are required to recover from errors, which may occur more frequently than the application can tolerate. In usual protocol stacks (like TCP/IP model), error control is performed at multiple levels, e.g., at the physical layer by error correction codes, at the data-link layer by ARQ techniques, as well as at the transport layer by TCP [1].

ARQ communication systems are based on the detection of errors in a coded block or frame and on the retransmission of the block or frame when errors have been detected. In this case a two way channel is needed in order to request retransmissions. For a given code, the error-detection capability possible in an ARQ system is higher than the error-correction capability of its FEC (Forward Error Correction) counterpart, because the error-control capability of the code is spent only on detection, while the correction requires not only the detection but also the localization of the errors. On the other hand, there is an additional cost in an ARQ system, which is the need for a retransmission link. There are also additional operations for the acknowledgement and repetition processes that reduce the transmission rate of the communication system [2]. In order to cut the right tradeoff between data reliability, latency, and efficient bandwidth usage, error control techniques must be carefully designed and their performance well understood [1].

There are two main ways of achieving reliability in data transfer: positive acknowledgement (ACK) and negative acknowledgement (NAK). The positive acknowledge-based transfer means that if a sender sends a message to the receiver, the receiver must send a message acknowledging its receipt. If the sender does not get an acknowledgement, then a failure in the system has occurred and the sender needs to take action e.g. send the message again or report an error. In this case, the reliability is maintained by the sender in that the sender needs to know all the members of the group and has the job of matching the acknowledgements with the list of group members. The receivers can be fairly lightweight; all they have to do is to send an acknowledgement for every message they receive. ACK based protocols are not very popular because they do not scale well.

The other approach is the Negative Acknowledgement (NAK). For this, the sender includes a sequence number in each outgoing message that receivers can check. If the receiver detects message loss, by detecting a broken sequence, it sends back a NAK. Hence a NAK based system is not suitable for a system where the frequency of sending messages is low. The reliability in this case is very much dependent on the receivers. The NAK system also needs to detect whether the group is active or not, i.e. all or any members are accessible. To achieve this, each member can send a "heartbeat" message periodically to the group. NAK systems can be scaled up relatively easily. For a group of N members with one sender, to achieve a successful delivery of a message we need a minimum of only one message (no NAK).

ARQ Protocols

We can categorize ARQ schemes into two categories, i.e.acknowledgement based and negative acknowledgment based protocols.

2.1. Acknowledgement-Based Protocols:

In this class we have three schemes, namely stop-and-wait, go-back-n and selective repeat.

2.1.1. Stop-and-Wait (SAW) ARQ

The usually used method to monitor sending and receiving data at the Data Link Layer DLL is to transmit frame(s) and wait for an acknowledgement(s) to be sure that the data have been received correctly or negative acknowledgement to retransmit a destroyed or lost frame, Fig 1.

This method has a guaranteed quality that all sent frames received correctly. But in turn it costs a lot because the receiver has to send an ACK to every correctly received frame. For example if we receive 1000 frames correctly then we have to answer with 1000 ACKs. Moreover in this algorithm we have to wait a timeout for every ACK or resend the unacknowledged frame. And this causes a long delay in data transfer. Hence, the efficiency of this scheme is usually under 50%.

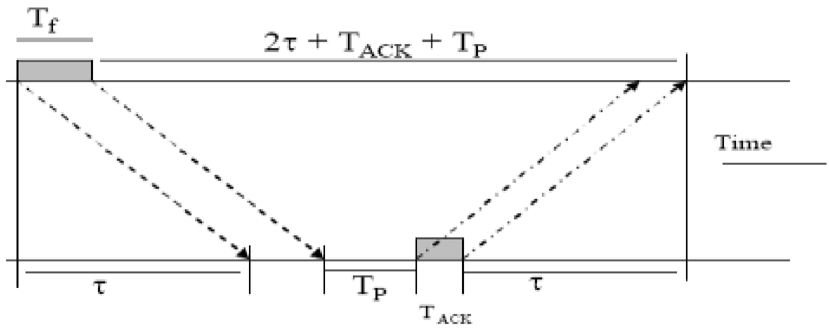

Fig.1 Stop-and-Wait ARQ method

2.1.2. Go-Back-N (GBN) ARQ

The Go-Back-N algorithm uses the sliding window (frames buffer) concept. The sender has a variable size window where the receiver has a window that can hold just one frame. So in this method the ACKs are used as well, but we it does not wait for each ACK for a timeout, instead a number of frames equal to the window size is sent (e.g. **8** frames) and the ACKs are sent during that transmission, Fig 2. If a frame is lost or destroyed a NAK is sent to the sender. And because the size of the receiver window is just **1** it discards all frame following the lost or damaged fame. Upon receiving the NAK, the sender retransmits the erroneous frame with all following discarded frames. This also have a high cost since the sender retransmit **N** frames when we need to resent only one frame. The same problem with the big number of ACKs arises here as in SAW scheme.

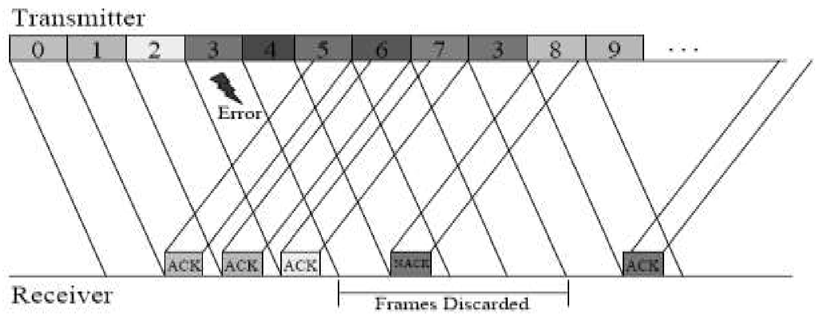

Fig.2 Go-Back-N ARQ method

2.1.3. Selective Repeat (SR) ARQ

This method is similar to the Go-Back-N but only the negatively acknowledged frame is retransmitted by the sender.

The GBN ARQ scheme becomes quite ineffective for communications systems with high data rates and large round-trip delays. This ineffectiveness is caused by the retransmission of many error-free frames following a frame detected in error. This can be overcome by using the selective-repeat ARQ protocol.

In SR error-control system, frames are also transmitted continuously. However, the transmitter only resends those frames that are negatively acknowledged (NAK'ed). After resending a NAK'ed frame, the transmitter continues transmitting new frames in the transmitter buffer. With this scheme, a buffer must be provided at the receiver to store the error-free frames following a received frame detected in error, because, ordinarily, frames must be delivered to the end user in correct order, for example, in point-to-point communications. When the first NAK'ed frame is successfully received, the receiver then releases any error-free frames in consecutive order from the receiver buffer until the next erroneously received frame is encountered. Sufficient receiver buffer storage must be provided in the SR ARQ system; otherwise, buffer overflow may occur and frames may be lost [3].

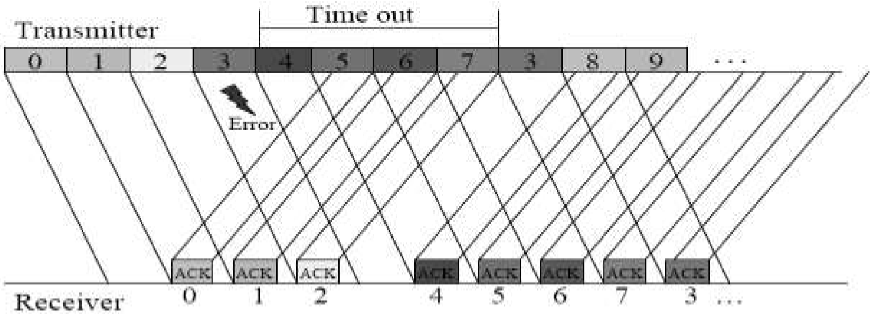

Fig.3 Selective Repeat ARQ method

2.2. NACK-based Protocols

NACK based protocols are more likely to be used in multicast transmissions, since they don't generate ACKs. In case of many receivers stream of ACKs could impair the sender performance, however there still exists problem of so called "NACK implosion", that is the situation when suddenly huge amount of repeat requests overloads the sender.

2.2.1. Go-Back-N (NAKGBN)

NAK-based GBN protocol is a very simple protocol that facilitates error free transmission of frames over a faulty network. The idea is to assume that the transmission of the data is successful until repeat request is received. It does not use ACKs to confirm the data arrival at the receiver, and only a NAK is generated when a faulty or out of order frame is received. It is very dependent on the parameters of the link, especially on the round trip time, which determines its ability to recover from the fault conditions. The protocol behaves exactly as ACK-based GBN, so it resends the faulty frame and all following frames. Big round trip time results in a big number of unnecessary retransmissions, which makes this protocol very inefficient in such conditions (e.g. satellite networks). Sender should have a buffer at least of the size of the data that can be sent by the network interface during one round trip time.

2.2.2. Selective Repeat (NAKSR)

A NAK-based SR protocol uses the available bandwidth much more efficiently, as compared to the Go-Back-N protocol above. Retransmission requests are made for every missing frame individually so only the requested frame is resent. This results in a smaller number of unnecessary retransmissions, making the protocol very efficient. The negative impact of this approach is that, in erroneous environment the number of frames sent back to the sender dramatically increases, so the possibility of a "NAK implosion" becomes higher. Another flaw of this model is the fact that it requires the sender to have all data available to be retransmitted until the transmission is completed. This could be made better, but for simplicity the decision was made to leave the protocol at this stage. At the same time the receiver has to allocate the memory for the whole size of the file that is transmitted, because the sequence of frame arrivals is not guaranteed. However, even though these disadvantages exist in this protocol, it achieves very good results, as will be shown.

Problem statement and model assumptions:

As discussed above the most important problem is the large number of ACKs that must be sent during the transmission process, e.g. for the SAW method the ACKs feedback represents about 40% of the total frames sent. This is more evident in wireless networks (long distance and satellite networks). So, we state the problem here as: how to decrease the proportion of feedback (ACKs) to the total data transmitted in data networks.

The proposed solution here is to use NAKs to control the flow and errors in data frame transmission. The most proper protocol to use is NAK-based SR scheme. NAK-based GBN is not useful in our case since it requires a lot of bandwidth and causes a longer delay because it resends all frames sent after the erroneously received frame. The idea in this solution is to send frames continuously until a NAK is received or until transmission ends. A NAK is initiated by the receiver if a frame is either lost or damaged. The receiver knows about lost frames when the next frame is received and knows about lost NAK if the requested frame is not received within the timeout. When a NAK is received by the sender it just retransmits the negatively acknowledged (NAKed) frame and continues transmission. If no NAK is received the receiver considers that all sent frames are correctly received. Of course the sliding window must be limited so the storage buffering the transmitted frames at the sender and the receiver is as low as possible. It is possible also to send all frames without using sliding window, but then we will need an unlimited memory for the sender to buffer sent frames for unexpected further retransmission and within the receiver to reorder the frames in the case some errors happened. In this case we can partially free memory when a NAK is received since this means that all former frames are received correctly so sender can delete their copies.

Analysis and Simulation

Herein, we will evaluate the proposed solution (i.e. NAK-based SR) and compare it to other ARQ techniques like SAW, GBN and SR through simulation of the throughput efficiency under various link conditions(in the sense of bit error rate, link speed, and frame size).

Analysis

The throughput efficiency of an ARQ scheme 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 [3]. In our analysis we would like to expose the effect of acknowledgement frames on the throughput so we will include ACKs as a part of the number of the average number of bits that could be sent during the transmission session. We will consider the possibility of errors not only in the data frame but also in the acknowledgement frames.

If we assume that the channel introduces a bit error with probability P and the number of bits in a frame is n_f, then frame error rate would be [4] :

$$P_f = (1\text{-}P)\ [\textit{no bit errors in frame}] = 1-(1-p)^{n_f} \approx 1-(1-n_f P) \approx n_f P$$

STOP-AND-WAIT ARQ protocol

Some models are devised for SAW in [2] and [3] but without taking into account the ACK frames. Here we consider this option and derive the next analytic model.

The average number of bits (including the ACK bits - n_a) that could be transmitted in the SAW is :

$$N_{SAW} = (n+\Delta tR)(1-P_f) + 2(n+\Delta tR)(1-P_f)P_f + 3(n+\Delta tR)(1-P_f)P_f^2 + \ldots + n_a$$

$$= (n+\Delta tR)(1-P_f)\left[1+2P_f+3P_f^2+\cdots\right] + n_a = (n+\Delta tR)(1-P_f)\sum_{i=1}^{\infty} iP_f^{i-1} + n_a$$

$$= \frac{(n+\Delta tR)(1-P_f)}{(1-P_f)^2} + n_a = \frac{(n+\Delta tR)}{(1-P_f)} + n_a$$

Where, n_a is the number of bits in an acknowledgement frame, and the total number of bits is $n = n_f + n_a$, and the round trip time is Δt and the data transmission rate is R.

The throughput efficiency for SAW scheme would be: $\eta_{SAW} = \dfrac{n_f}{N_{SAW}}$

GO-BACK-N ARQ protocol

The next analytic model is based on models derived in [2] and [3] but with ACKs frames included in calculations.

$$N_{GBN} = n(1\text{-}P_f) + n(N+1)(1\text{-}P_f)P_f + n(2N+1)(1\text{-}P_f)P_f^2 + \cdots\cdots + n_a(1\text{-}P_a) +$$
$$+ n_a(1\text{-}P_a)P_a + n_a(1\text{-}P_a)P_a^2 + \cdots = n(1\text{-}P_f)\left[NP_f(1+2P_f+3P_f^2+\cdots)+(1+P_f+P_f^2+\cdots)\right]$$
$$+ n_a(1\text{-}P_a)\left[1+P_a+P_a^2+\cdots\right] = \frac{n(1\text{-}P_f)NP_f}{(1\text{-}P_f)^2} + \frac{n(1\text{-}P_f)}{(1\text{-}P_f)} + \frac{n_a(1\text{-}P_a)}{(1\text{-}P_a)} =$$
$$= n\left[\frac{NP_f}{(1\text{-}P_f)} + 1\right] + n_a$$

Where N is the sliding window size (i.e. the number of frames that could be transmitted without waiting for an acknowledgement).

The throughput efficiency for GBN scheme would be : $\eta_{GBN} = \dfrac{n_f}{N_{GBN}}$

Selective Repeat ARQ protocol

SR model is also mentioned in [2] and [3] but here we consider the ACKS frame as well.

$$N_{SR} = n(1\text{-}P_f) + 2\,n(1\text{-}P_f)P_f + 3n(1\text{-}P_f)P_f^2 + \ldots\ldots + n_a(1\text{-}P_a) +$$
$$+ n_a(1\text{-}P_a)P_a + n_a(1\text{-}P_a)P_a^2 + \cdots = n(1\text{-}P_f)\left[1+2P_f+3P_f^2+\cdots\right] + n_a(1\text{-}P_a)\left[1+P_a+P_a^2+\cdots\right]$$
$$= \frac{n(1\text{-}P_f)}{(1\text{-}P_f)^2} + \frac{n_a(1\text{-}P_a)}{(1\text{-}P_a)} = \frac{n}{(1\text{-}P_f)} + n_a$$

The throughput efficiency for GBN scheme would be: $\eta_{SR} = \dfrac{n_f}{N_{SR}}$

Negative Acknowledgement SR (NAKSR) ARQ scheme

As mentioned previously in this method the frame is not acknowledged and only the failure frame is negatively acknowledged. The next model of NAKSR is derived basing on the ACK-based SR by neglecting the ACK frames and considering NAK frames. NAK frames size is considered the same as ACK frames (n_a).

$$N_{NKSR} = n(1\text{-}P_f) + (2n - n_a)(1\text{-}P_f)P_f + (3n - n_a)(1\text{-}P_f)P_f^2 + \ldots\ldots + n_a(1\text{-}P_a) +$$
$$+ n_a(1\text{-}P_a)P_a + n_a(1\text{-}P_a)P_a^2 + \cdots = n(1\text{-}P_f)\left[1+2P_f+3P_f^2+\cdots\right] - n_a(1\text{-}P_f)P_f\left[1+P_f+P_f^2+\cdots\right]$$
$$+ n_a(1\text{-}P_a)\left[1+P_a+P_a^2+\cdots\right] = \frac{n(1\text{-}P_f)}{(1\text{-}P_f)^2} - \frac{n_a P_f(1\text{-}P_f)}{(1\text{-}P_f)} + \frac{n_a(1\text{-}P_a)}{(1\text{-}P_a)} = \frac{n}{(1\text{-}P_f)} + n_a(1-P_f)$$

And the throughput efficiency for NAKSR would be: $\eta_{NKSR} = \dfrac{n_f}{N_{NKSR}}$

Simulation

Using the formerly derived equations for different ARQs, we will conduct some calculations for various values of frame size and link speed. The calculations and plots are done using MATLAB. For all experiments we assume a point-to-point communication with noisy channel.

First Experiment (Testing the effect of the data frame size)

We take the Go-Back-N window size as 8, the transmission rate R = 128kbps, the round trip time Δt = 40 ms [long distance link], n_f = 1024Bytes = 8192bits; n_a = 32Bytes = 256bits.

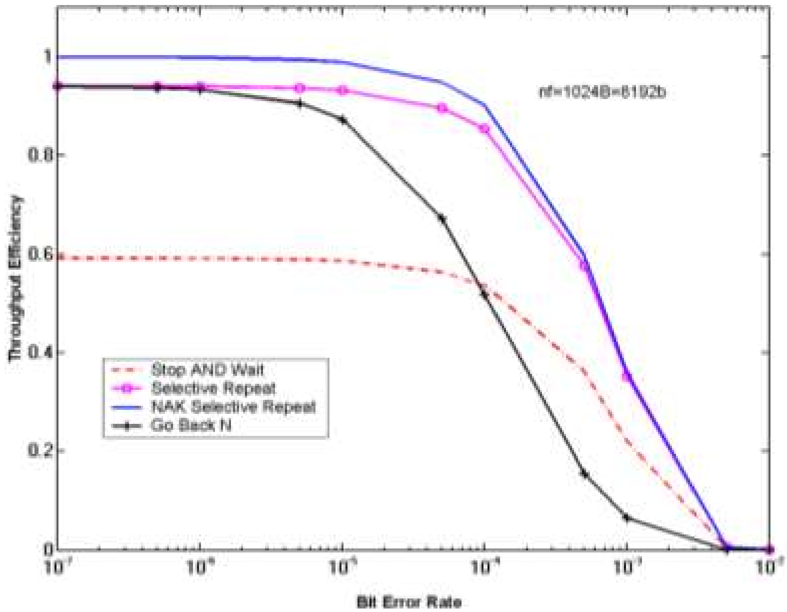

Fig. 4 Throughput efficiency for the 4 studied ARQ protocols n_f=1024B

This experiment shows (Fig. 4) that when error rate is low the performance is acceptable for the GBN, SR and NAKSR ARQ schemes, but SAW efficiency is about 60% so it is unacceptable. Also we can note that NAKSR outperforms all other methods and for as high error rate as $5*10^{-4}$. for higher error rates (i.e. $<10^{-3}$) the performance decreases under 50%. Of course GBN protocol shows very bad performance for error rate of 10^{-5}.

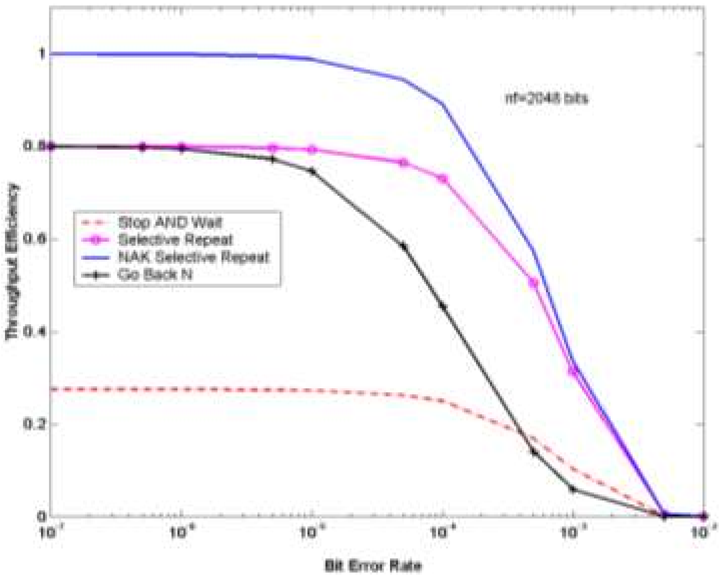

Fig. 5 Throughput efficiency for the 4 studied ARQ protocols n_f=256B

Having decreased the size of the frame to 256B the efficiency of SAW SR and GBN is decreased by levels of 30% - 20%, Fig 5. However the NAKSR kept its performance unchanged.

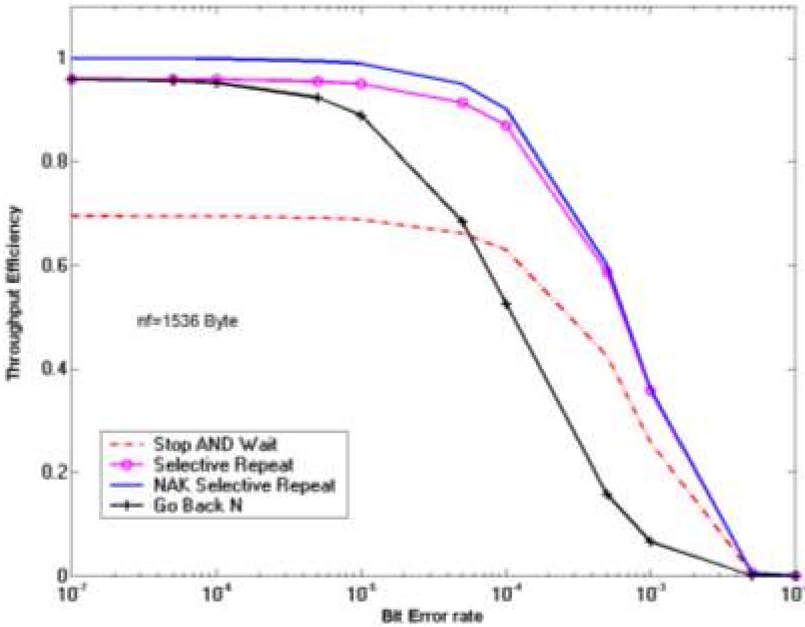

Fig. 6 Throughput efficiency for the 4 studied ARQ protocols n_f=1536B

Increasing the data frame size shows important improvement in the performance of all 4 protocols. In Fig 6, as the frame size become 1536B the efficiency of SAW increased to 70% while other protocols achieved ~95% throughput efficiency.

The Second Experiment (the effect of the data rate)

Now we will consider the effect of changing the data transmission rate. As we can see from the equations above the data rate has only effect on the efficiency of the SAW ARQ scheme. Hence, we will run just one experiment with transmission rate R = 10Mbps, and frame size n_f = 8192 bits and the acknowledgement frame size n_a = 256bits.

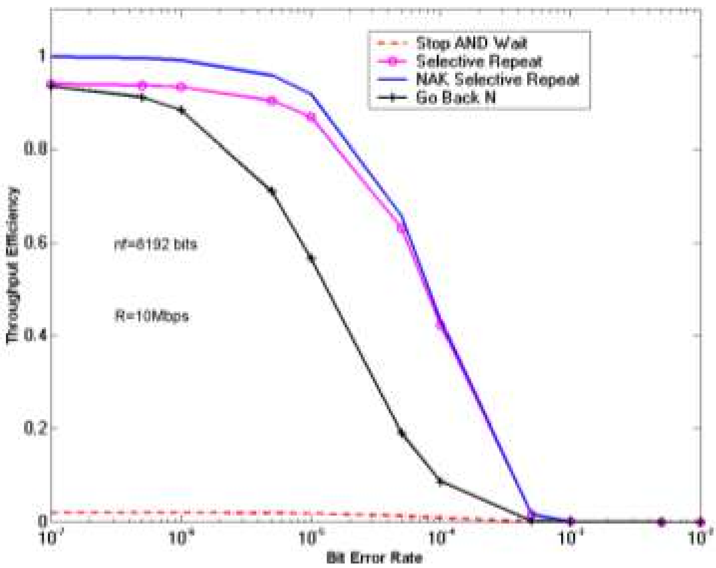

Fig. 7 Throughput efficiency for the 4 studied ARQ protocols, R=10Mbps

As it is illustrated on Fig 7, the throughput of the SAW ARQ scheme degrades dramatically when the data rate is increased, this can be explained by the fact that we are modelling a long distance link about 12000km, and hence the round-trip time is long (Δt = 40ms) and because the rate is high (R = 10Mbps) during this wasted time (idle period waiting for ACK) the transmitter could send Δt *R = 10M*40ms = 400kb. This amount of data could be sent during the ACK waiting time but this does not happen in SAW so it is considered as a wasted Bandwidth.

Third Experiment (changing the acknowledgement frame size)

Here, we decrease the size of the ACK frame from 256 bits to 128 bits, and then we observe the change in the efficiency of the studied protocols. We keep the values of the other parameters as in the second experiment. Results of this experiment are depicted on Fig 8.

From the results, we can conclude that the size of ACK frame has a little effect on the performance of ARQ protocols. This happens because of the proportion of the data frame size to the ACK frame size, which is here (8192/128) 64 times, so the small size of ACK frame can be neglected when compared to the big data frame size. But if there sizes were close the performance will be degraded as were shown in Fig 5, where the proportion ration was (2048/256) 8 times.

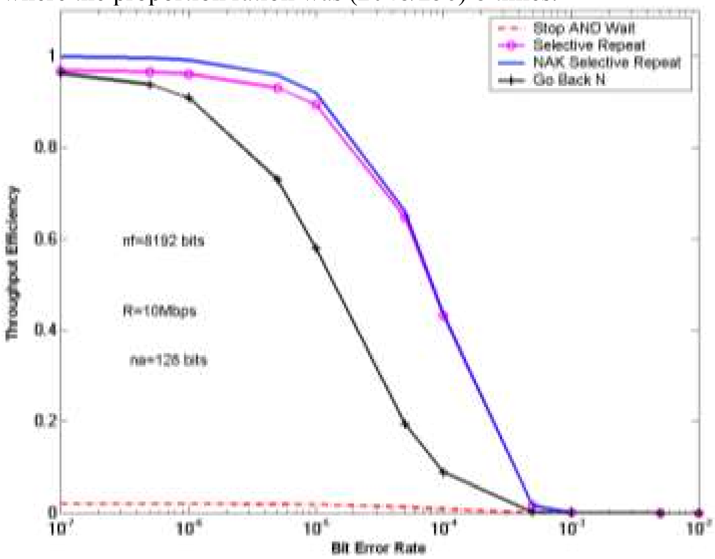

Fig. 8 Throughput efficiency for the 4 studied ARQ protocols, n_a=128bits

Conclusions

In this paper, the concept of the NAK-based ARQ approach were introduced and evaluated as an alternative ARQ method in point-to-point connection networks. This method was aiming to decrease the traffic sent as feedback (ACKs) from the receiver to sender.

The analysis performed and results achieved have shown that NAK SR scheme outperforms all other known ARQ protocols by the meaning of the throughput (the number of bits that must be sent in order to deliver the information data).

Many studies (e.g. [5] and [6]) have been conducted on the use of NAK method but just in multicasting connections. They proofed the suitability of this scheme for this type of connections. Herein, we have proofed the possibility of using this approach in point-to-point connections especially when data frame size is big, the links are of a long distance and the error rate is low.

Of course this approach is not acceptable in the noisy environments where the error rate reaches a degree of more than $5*10^{-4}$. In this case the throughput efficiency decreases rapidly and gets into unacceptable throughput.

References

[1] M. Rossi, L. Badia, M. Zorzi; "*On the delay statistics of an aggregate of SR-ARQ packets over Markov channels with finite round-trip delay",* In Proceedings of IEEE WCN Conference 2003, pp. 1773-1778.

[2] J. C. Moreira and P. G. Farrell; "*ESSENTIALS OF ERROR-CONTROL CODING"*, 2006 , John Wiley & Sons Ltd, pp. 68-72,

[3] Lin, S. D. Costello, and M. Miller "*Automatic Repeat Request Error Control Schemes* ", December 1984-Vo1.22, No. 12 IEEE Communications Magazine

[4] A. Leon Garcia and I. Widjaja, "*Communication Networks"*. McGraw Hill, 2000.

[5] M. Yamamoto, Y. Sawa, S. Fukatsu, and H. Ikeda, "NAK-based flow control scheme for reliable multicast communications," IEICE Trans. Comm., vol.E82-B, no.5, pp.712–720, May 1999

[6] K. Yamamoto, et. al**.**, **"***performance Evaluation of Ack-based and NAK-based Flow Control Mechanisms for Reliable Multicast Communications"*, IEICE Trans. Comm., Vol.E84–B, No.8 August 2001.