RESEARCH ARTICLE OPEN ACCESS

Analyzation of Automatic Repeat Request (ARQ) Protocols

[1]Jeshvina.S, [2]Sneha.P, [3]Saraanya.S
Final year BCA, Dept of Computer Science New Horizon College Kasturinagar, Bangalore

Abstract:

Automatic Repeat Request (ARQ) protocols provide a widely recognized avenue for guaranteeing reliable transmissions over unreliable transmission of data. They are implemented in data link layer. The main functions of data link layer are data link control and media access control. Data link control ensures that the data is received at the destination side without any error and loss; by two mechanisms i.e. flow control and error control. Whereas Media access control is about how to share the link. The protocols are used in error and flow control mechanisms to ensure the reliability of the transferred data; they are divided into two Protocols for noiseless channel and Protocols for noisy channel.

Keywords — **Flow control, Error control, ARQ, ACK (acknowledgement), NAK, Frames.**

1. INTRODUCTION:

Communication means sharing of data between two devices. It is exchange of data between two devices via some form of transmission medium such as a wired cable. The data communication system consists of five components i.e. Sender, Receiver, Message, Transmission medium and Protocol. Sender sends the message to other nodes of the network. Receiver receives the message from sender. Message is data that is to be communicated. Transmission medium is a path by which message travels from one node to another.

Protocol is a set of rules that helps to control data communication between communication devices. OSI (Open System Interconnection) model is commonly used to specify and describe computer communication protocol. It was first taken into consideration in late 1970s. The motive of the OSI model is to show how to facilitate communication between different systems. It is not a protocol; it is a model for designing and understanding network architecture. The second layer of OSI model is Data Link Layer it handles a system which have two nodes connected via a physical link.

The major work of this layer is to establish a connection between the nodes of network, one for sending and other for receiving. Data Link Layer has two sub layers: Data Link Control and Media Access Control. The data link layer is responsible for designing and communication between two contiguous nodes and it controls and ensures that the data is received at the destination side without any error and loss. In the process of transmission, frames may be lost, the mechanism of detecting and correcting this loss is known as error correction. Automatic repeat request (ARQ) is a technique for ensuring accurate data delivery to the user in spite of the error occurrence. ARQ provides reliable transmission of information. The two main transmission errors include, frame injury and frame loss. ARQ and Forward Error Correction (FEC) are being used as control protocols to fix these errors. They use time-out retransmission, negating and retransmission mechanism, positive acknowledgement and error detection to solve the errors. ARQ is applied both in wired and wireless networks and it provides this error detection and recovery based on feedback messages and retransmission. Reliability of the transmission can be improved by adding error control at the transport layer. ARQ can detect lost packet at the sender site using a timer which starts when the packet is sent [1].

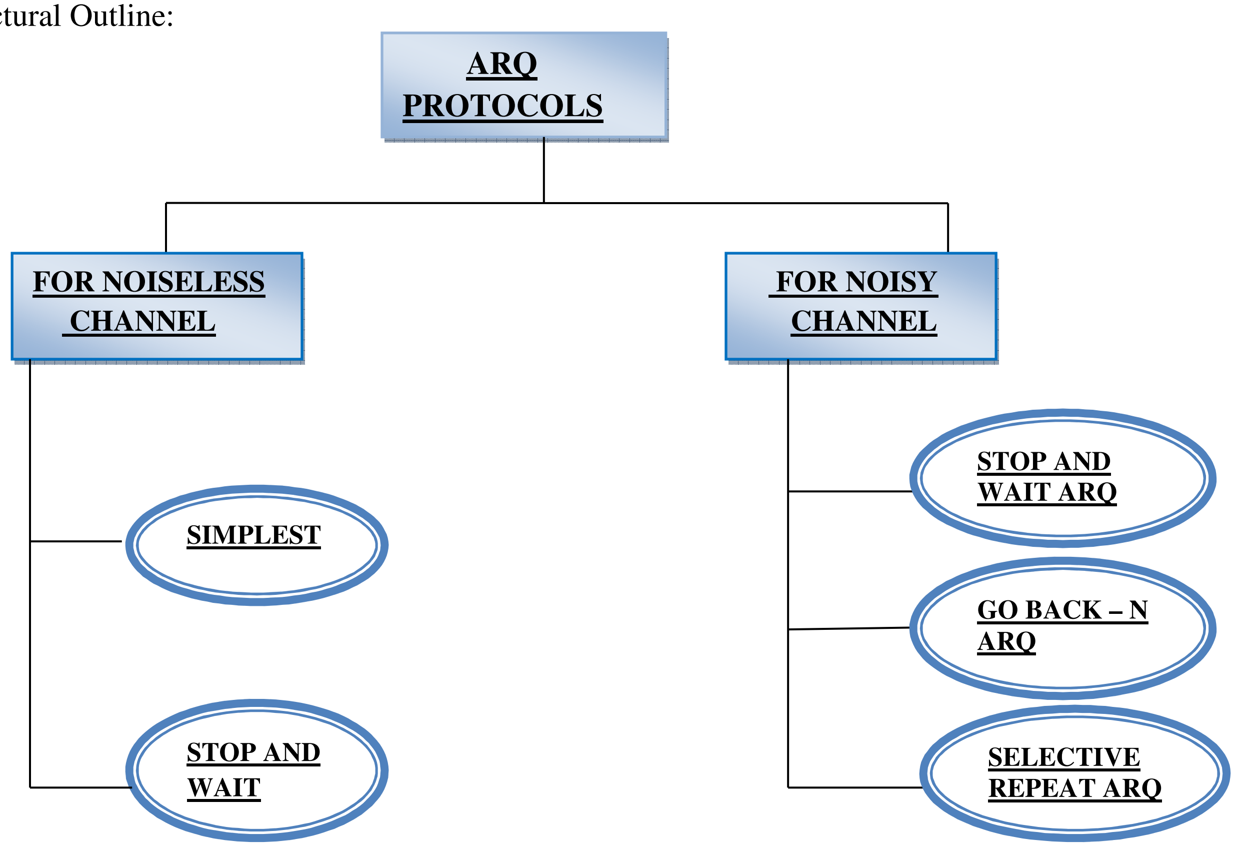
ARQ PROTOCOLS:

Automatic Repeat Request Protocol (ARQ), also called Automatic Request Query (ARQ) to implement flow and error control mechanism, there are different protocols to automatically initiate a call to retransmit any data packet or frame after receiving incorrect data. Protocols are normally implemented in software by using any common programming languages. These protocols are divided into two categories:

i. Protocols For Noisy Channel
ii. Protocols For Noiseless Channel

In real life, noiseless channels are not available. So the concentration is more on noisy channel protocols, which emphasis Flow control and Error Control mechanisms. These Protocols reside in the Data Link Layer and Transport Layer of the OSI model.

Structural Outline:



2. NOISELESS CHANNELS:

A Noiseless Channel is an ideal channel in which frames are not lost, duplicated or corrupted. There are two protocols for Noiseless Channels namely:

a. Simplest Protocol
b. Stop And Wait Protocol

a) Simplest Protocol:

As the name indicates, this protocol is very simple having no flow or error control. It is a unidirectional protocol in which data frames travel only from source to receiver. The receiver can immediately handle any frame it travels with negligible processing time. The protocol consists of two distinct procedures: a sender and receiver. The sender runs in the data link layer of the source machine and the receiver runs in the data link layer of the destination machine. No sequence number or acknowledgements are used here.

Design:

- The Sender site cannot send a frame until its network layer has a data packet to send.
- The Receiver site cannot deliver a data packet to its network layer until a frame arrives.
- **Working**: If the Protocol is implemented as a procedure, we need to introduce the idea of events in the Protocol. The procedure at the sender site is constantly running; there is no action until there is a request from the network layer. The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.

FIG 1.1 DESIGN OF THE SIMPLEST PROTOCOL WITH NO FLOW OR ERROR CONTROL

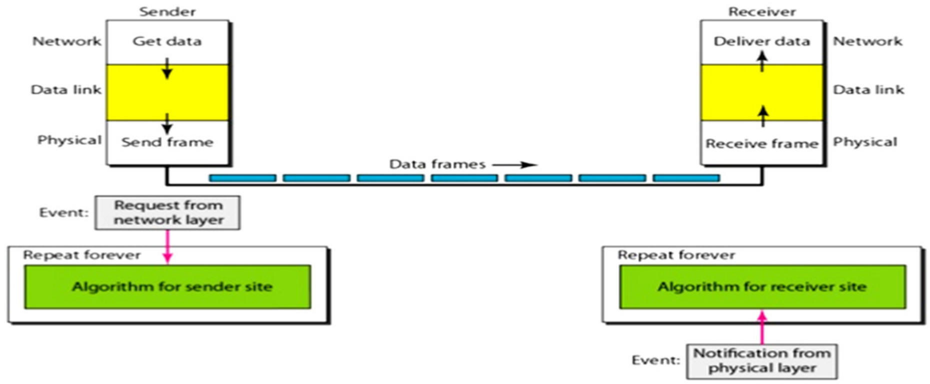
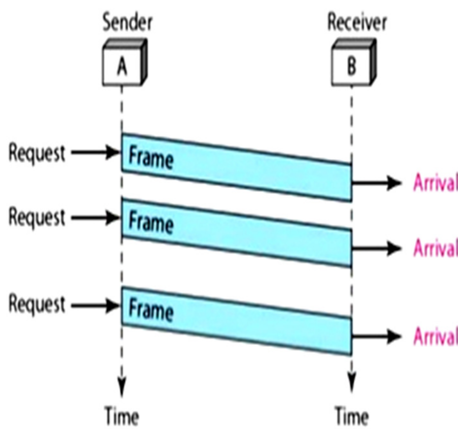


FIG 1.2 SEQUENCE OF EVENTS:



- The sender sends a sequence of frames without even thinking about the receiver.
- To send three frames, three events occur at the sender site and three events at the receiver site.
- Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame [2].

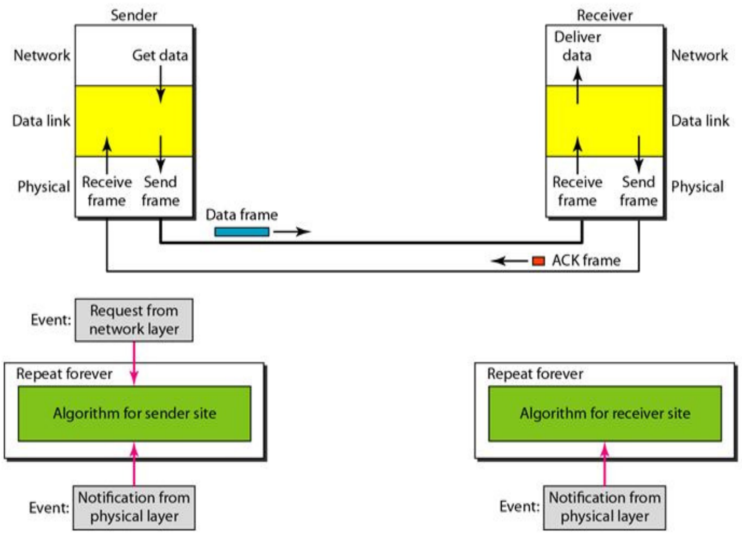
b) STOP AND WAIT ARQ:

The data frames arrive at the receiver faster than they can be processed, the frames must be stored until they are used. If the receiver does not have enough storage space, then packets are discarded. To prevent receiver from drowning with the packets, someone has to tell the sender to slow down. There must be feedback from the receiver to the sender. In stop-and-wait ARQ the transmitter and the receiver work on the delivery of one frame at a time. The sender sends one frame and stops until it receives confirmation from the receiver and then sends the next frame.

Design:

- The data arrives from network layer.
- Then in data link layer frame is made and sent to physical layer and timer is started at the sender side.

FIG 2.1: DESIGN OF THE STOP AND WAIT ARQ PROTOCOL:



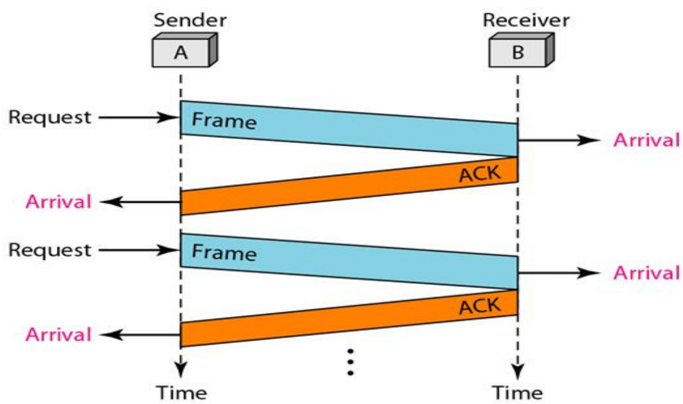
Algorithm at Sender site:

- Get data from Network layer
- Make frame in Data link layer and store copy of the frame
- Send frame to Physical Layer
- Start timer and wait for the Ack frame from the receiver
- Receiver Ack frame from physical layer
- If Ack is received before timer expires success or else retransmit frame.

Algorithm at Receiver site:

- Receive frame from Physical Layer
- Extract data in Data Link Layer
- Deliver data to Network layer
- Send Ack frame to sender

FIG 2.2: SEQUENCE OF EVENTS IN STOP AND WAIT ARQ PROTOCOL:



- Sender A transmits frame 0 and starts timer and then waits for an Ack frame.
- Frame 0 is received without error so receiver B transmits an Ack frame.
- The Ack frame B is also received before timeout
- Therefore the Sender A knows the frame 0 has been received without any error, loss, or duplication.
- Sender A transmits frame 1 and then resets the timer, and process continues.

3. NOISY CHANNELS:

Although the stop and wait protocol gives the clear idea of how to add flow control to its predecessor, noiseless channels are nonexistent, they can be ignored. Or one can add error control to the protocols. The protocols that can be used in this process are:

a. Stop-and-Wait ARQ
b. Go-Back-N ARQ
c. Selective Repeat ARQ.

a) Stop-and-Wait ARQ:

The Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

SEQUENCE NUMBERS:

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. For example, if we decide that the field is *m* bits long, the sequence numbers start from 0, to *2m* - 1, and then are repeated.

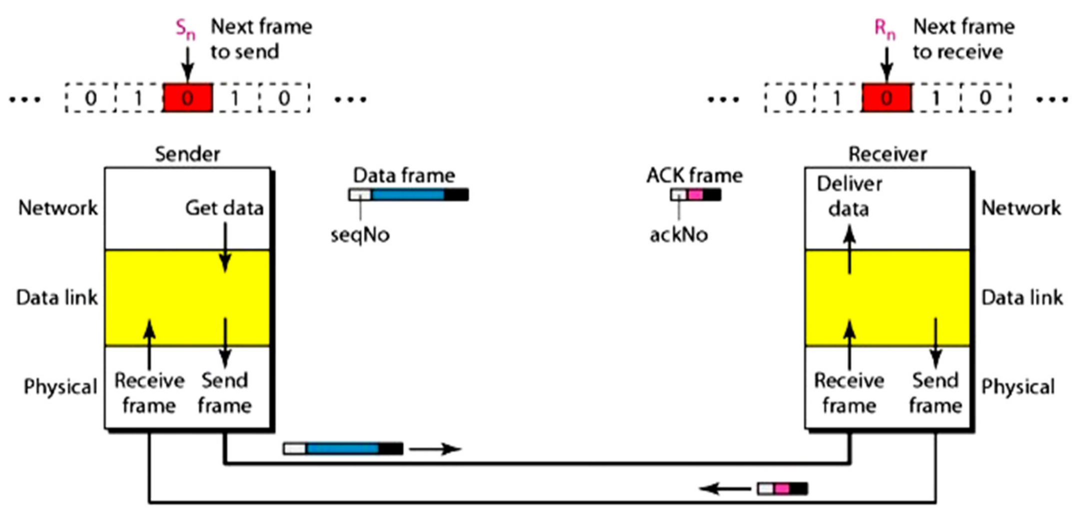
ACKNOWLEDMENT NUMBERS:

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected). The receiver has a control variable called as R_n that holds the number of the next frame expected. When the frame is received, the value of R_n is incremented (modulo-2) which means if it is 0, it becomes 1 and vice versa. And for this purpose the Sender and Receiver will have control variables called S_n and R_n. Initially $S_n = 0$ and $R_n = 0$. S_n holds the sequence number for the next frame to be sent (0 or 1). When a frame is sent, the value of Slast is incremented (modulo-2) which means if it is 0, it becomes 1 and vice versa, so the sequence numbers are numbered as 0, 1, 0, 1, 0, 1 and so on.

Design:

Figure 3.1shows the design of the Stop-and-Wait ARQ Protocol. The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seq No (sequence number); an ACK frame uses an ack No (acknowledgment number).

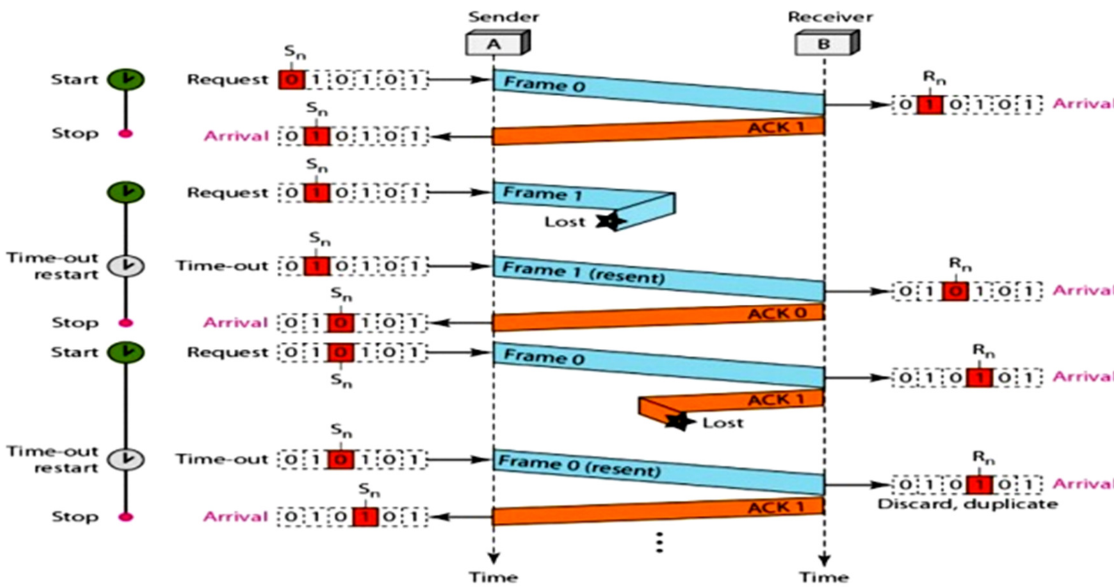
FIG 3.1: DESIGN OF THE STOP-AND-WAIT PROTOCOL:



SEQUENCE OF EVENTS:

FIG 3.2: SEQUENCE OF EVENTS FOR STOP-AND-WAIT ARQ PROTOCOL FOR NOISY CHANNEL:

- **Pipelining:** In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining. There is no pipelining in Stop-and-Wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent. However, pipelining does apply to our next two protocols because several frames can be sent before we receive news about the previous frames. Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product.



- Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out.
- The resent frame 1 is acknowledged and the timer stops.
- Frame 0 is sent and acknowledged, but the acknowledgment is lost.
- The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

b) GO-BACK-N ARQ:

In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive. It overcomes the inefficient transmissions that occurs in stop and wait ARQ. To improve the efficiency of transmission multiple frames must be in transition while waiting for an acknowledgement [3].

SEQUENCE NUMBERS:

Frames from a sending station are numbered sequentially. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to $2^m - 1$. For example, if m is 2 the sequence numbers are 0 to $2^m - 1$, i.e. 0 to $2^2 - 1$ to $4 - 1$ i.e. 0-3 i.e. 0,1,2,3,0,1,2,3,0, and so on.

SLIDING WINDOW:

Sliding window is a concept that defines the range of sequence numbers both for sender and receiver. The range which is of concern to the sender is called *Send Sliding Window,* the range that is the concern of the receiver is called *Receiving Sliding Window.*

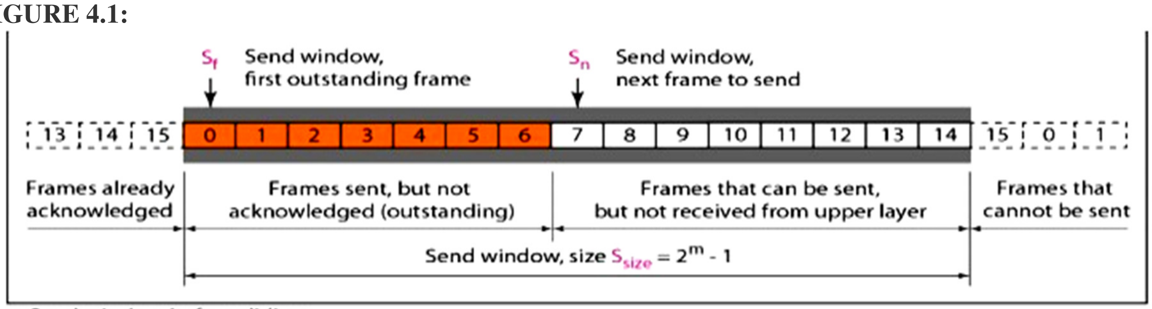
SEQUENCE NUMBERS:

Frames from a sending station are numbered sequentially. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2m–1.

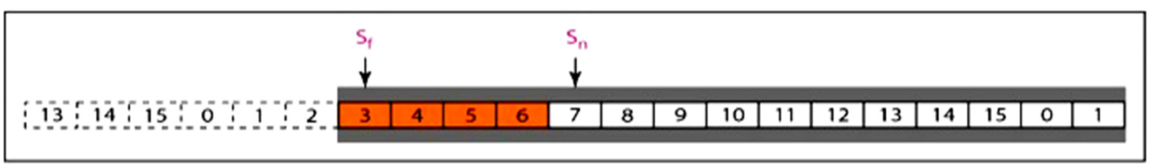
SEND WINDOW FOR GO-BACK-N ARQ:

The sender does not worry about these frames and keeps no copies of them. The second region, colored in Figure 4.1 a, defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The window itself is an abstraction; three variables define its size and location at any time. We call these variables *Sf (send* window, the first outstanding frame), *Sn* (send window, the next frame to be sent), and Ssize (send window, size). The variable *Sf* defines the sequence number of the first (oldest) outstanding frame. The variable *Sn* holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable Ssize defines the size of the window, which is fixed in our protocol.

FIGURE 4.1:



a. Send window before sliding

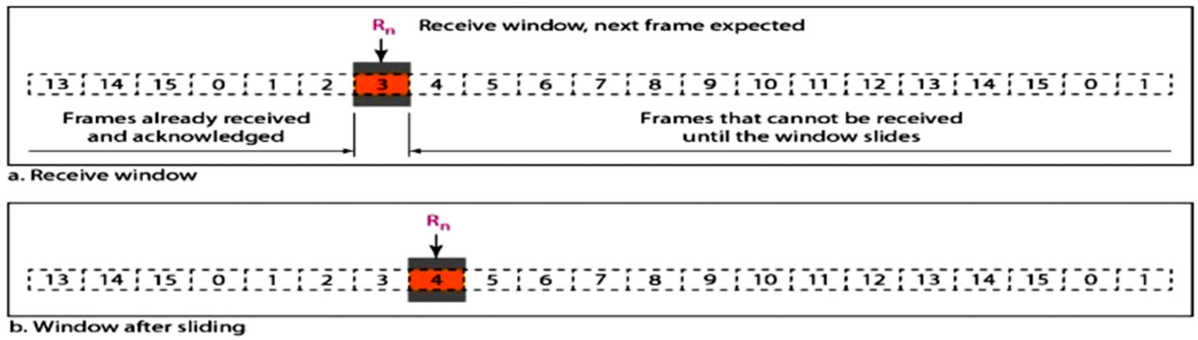


b. Send window after sliding

RECEIVE WINDOW FOR GO-BACK-N ARQ:

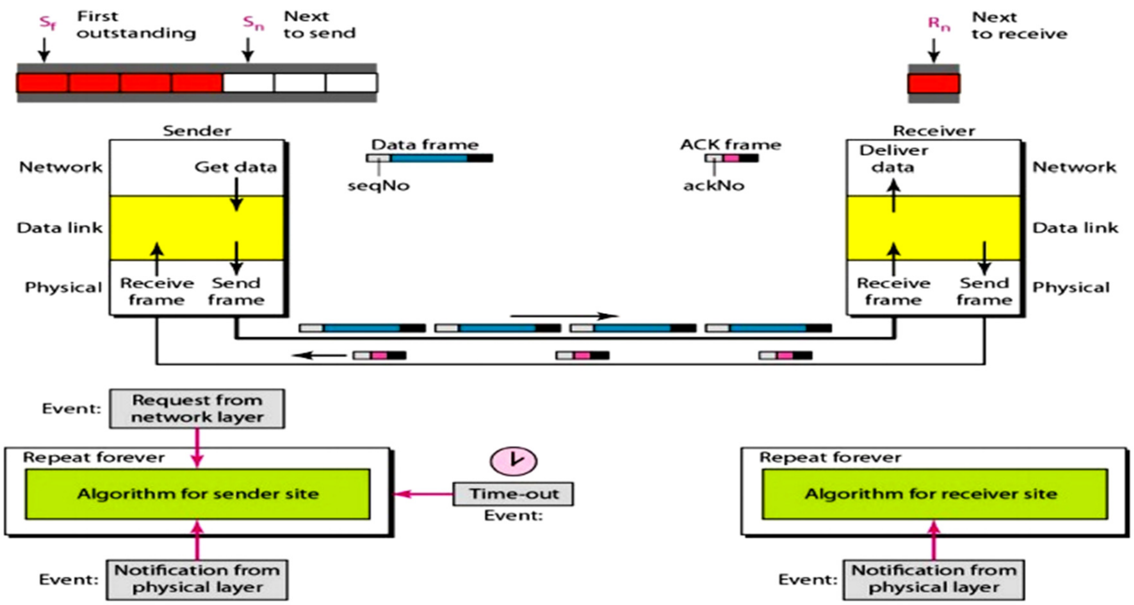
The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always 1.

FIGURE 4.2:



DESIGN:

Figure 4.3 shows the design for this protocol. As we can see, multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction. The idea is similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window.



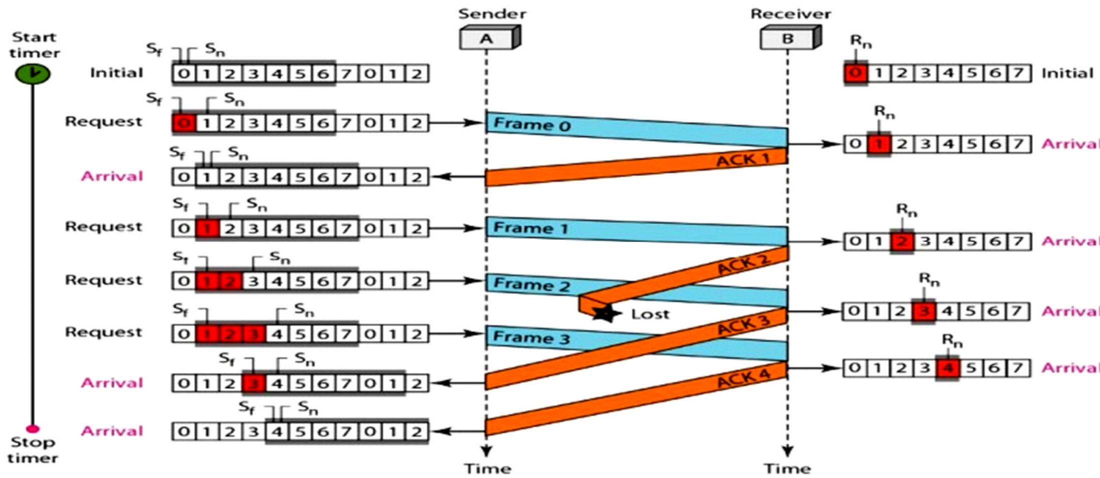
SEQUENCE OF EVENTS:

Figure 4.4 shows an example of Go-Back-*N*. This is an example of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost.

FIGURE 4.4: SEQUENCE OF EVENTS FOR Go-Back-N ARQ:

- After initialization, there are seven sender events.
- Request events are triggered by data from the network layer; arrival events are triggered by acknowledgments from the physical layer.
- There is no time-out event here because all outstanding frames are acknowledged before the timer expires.

Note that although ACK 2 is lost, ACK 3 serves as both ACK 2 and ACK3. There are four receiver events, all triggered by the arrival of frames from the physical layer.



SELECTIVE REPEAT ARQ:

In Go-Back-N ARQ if one frame is corrupted then all other N frames were resent. Also the receiver has to keep track of only one frame and does not buffer out of order frames as they are discarded. The resending of multiple frames by the sender uses up the bandwidth and slows down the transmission. Selective Repeat ARQ overcomes the drawback of Go-Back-N ARQ. In this mechanism, instead of resending N frames only the damaged frame is resent. This is more efficient approach for noisy links; but the processing at the receiver end becomes more sophisticated.

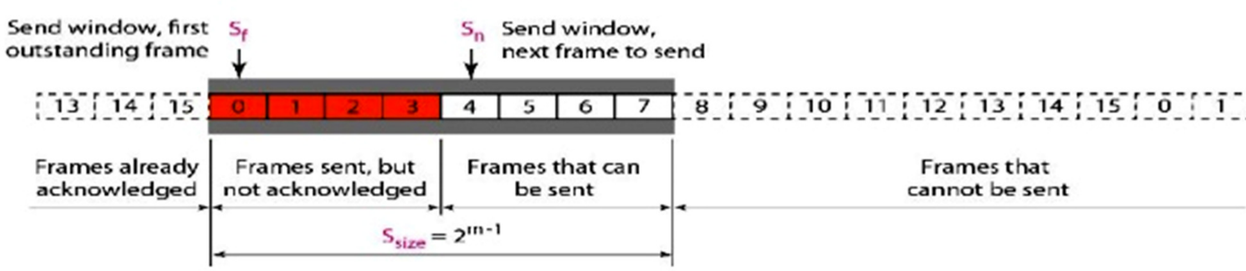
WINDOWS:

The Selective Repeat Protocol also uses two windows: a send window and a receive window.

SEND WINDOW FOR SELECTIVE REPEAT ARQ:

The size of the send window is much smaller; it is 2m- 1. Second, the receive window is the same size as the send window. The send window maximum size can be 2m- 1. For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this.

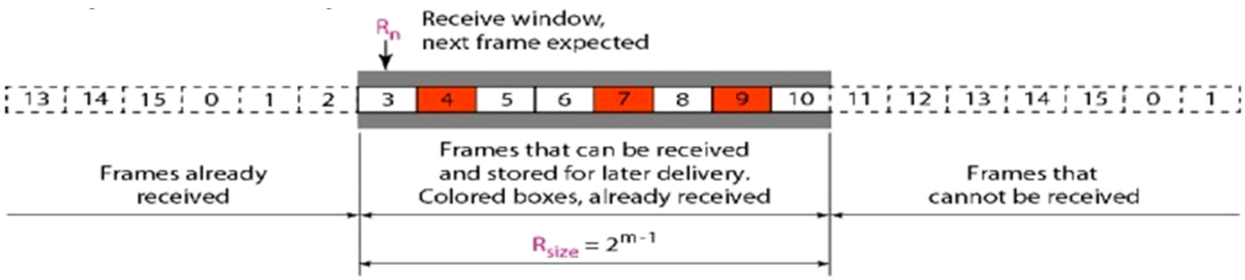
FIG: 5.1



RECEIVE WINDOW FOR SELECTIVE REPEAT ARQ:

The receive window in Selective Repeat is totally different from the one in Go Back- N. First, the size of the receive window is the same as the size of the send window (2m- 1). Figure 5.2 shows the receive window in this protocol. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer [4].

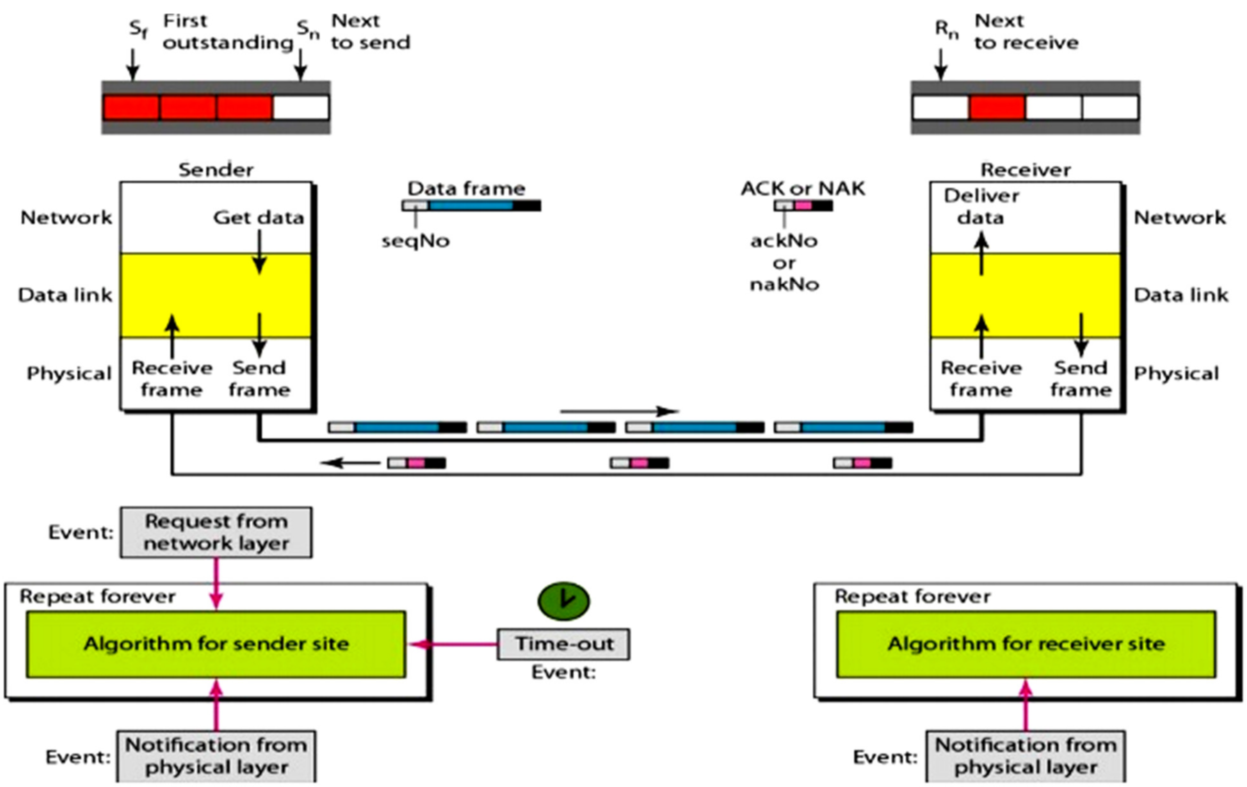
FIG: 5.2



DESIGN:

The design in this case is to some extent similar to the one we described for the 00Back-N, but more complicated, as shown in Figure 5.3.

FIG 5.3: DESIGN FOR SELECTIVE REPEAT ARQ PROTOCOL:



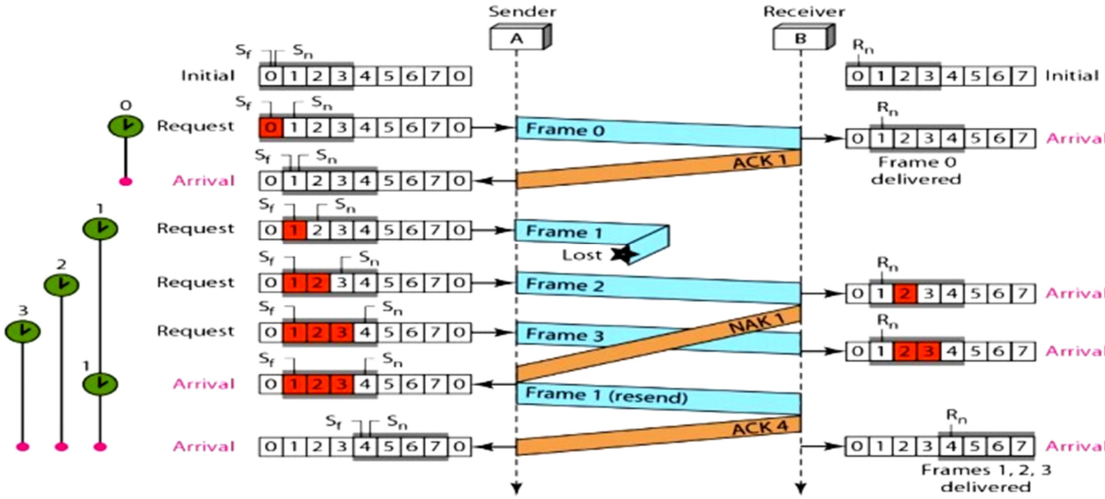
SEQUENCE OF EVENTS:

Frame 1 is lost. The behavior of Selective Repeat ARQ is as shown,

FIG 5.4:

One main difference is the number of timers. Here,

- Each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3).
- The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives. The timer for frame 1 starts at the second request restarts when a NAK arrives, and finally stops when the last ACK arrives. The other two timers start when the corresponding frames are sent and stop at the last arrival event.



4. PIGGYBACKING:

The three protocols discussed in noisy channel protocols are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction. In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A. Along with ACK is also transmitted in one frame. I.e. FRAME= DATA + ACK. This method increases the overall efficiency of transmission [5].

5. CONCLUSION:

Automatic Repeat Request (ARQ) is a technique used to ensure that the data stream is delivered accurately to the users despite the errors that occur during transmission. ARQ forms the basis for the peer – to – peer protocol that provides reliable transfer of information. In this paper the three basic types of protocols starting with the simplex and building up to the most complex is discussed. And also the protocols for the noiseless channels are also analyzed.

Selective Repeat is always considered the best of the three main types of ARQ protocols because there is time and transmission utilization. Unlike Stop and wait which though has both flow and error controls is inefficient because of the time it waste waiting for acknowledgement. The Go-Back-N is a better version of Stop and Wait which utilizes pipelining but also waste time when erroneous frame is encountered. So the comparison between them can be done by considering various factors,

FACTORS:	STOP-AND WAIT ARQ	GO-BACK-N ARQ	SELECTIVE REPEAT ARQ
1- Sender slide window size	1	$2^m - 1$	$2^m - 1$
2- Receiver slide window size	1	1	$2^m - 1$
3- ACK	YES	YES	YES

4- NAK	NO	NO	YES
5- Frame sequence	0, 1, 0, 1, 0, 1…	0……….2^m - 1	0……….2^m - 1
6- Band width utilization	low	medium	high

This is not just simply to say which protocol is better or faster than the others. It depends on the environment and all the other parameters. One needs to choose proper protocol for suitable situation. and hence ARQ protocol is used in noisy protocols for flow and error control, and the protocols are divided into two the noisy channel and noiseless channels; the noisy channel consisting of Stop-And-Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ protocols. The noiseless channel consists of the Simplest ARQ and Stop-and-Wait ARQ protocols. There is no loss of frame, error or duplication scenarios in Noiseless channels but whereas there are chances of error, loss of frame or duplication of frame in Noisy channel, and hence the Automatic Repeat Request Protocol (ARQ) is introduced to ensure data delivery, to control the flow and error control.

6. REFERENCES:

[1] H. Zimmermann, "OSI Reference Model - The IS0 Model of Architecture for Open Systems Interconnection", IEEE trans. On Communications, Vol COM-28, pp 425-432, April 1980.

[2] **International Journal of Engineering Trends and Technology (IJETT) – Volume 18 Number2- Dec 2014** ISSN: 2231-5381 http://www.ijcttjournal.org Page 64 Implementation of Data Link Control Protocols in Wired Network Sudhanshu Maurya#1, Vikas Kumar Nayak*2, Dr. A Nagaraju#3

[3] Geetha Vidya, “Data Communication and Networks” skyward publications, July 2016 edition.

[4] Google Trends.

[5] Anurag Kumar, D.Manjunath, Joy Kuri, “Communication Networking”, Morgan Kaufmann Publishers 2004 edition.