UNIT 2 RETRANSMISSION STRATEGIES

Structure **Page Nos.**

2.0 INTRODUCTION

In the previous unit we discovered that, Data Link Layer is responsible for ensuring that data is received at the receiver's end in line and error free. For the same task it has two important functions to perform that is Flow control and Error control. Flow and Error control are performed by the data link protocol. Before starting discussion on methods for flow and error control, firstly, we will define Flow control and Error control.

Flow control deals with when the sender should send the next frame and for how long the sender should wait for an acknowledgement. Data link protocol takes care of the amount of data that a sender can send and that a receiver can process as, the receiver has its own limitation in terms of speed for processing the frames. It also sees the compatibility of speed of both the sender and the receiver.

Error control deals with error detection and correction method that we have already discussed in the previous unit. If, an error is found in the frame either due to loss of frame or due to damage of frame, retransmission of the same is required by the sender. Retransmission is required when a sender does not receive a positive acknowledgement in time, due to a loss of frame or loss of acknowledgement or if, the sender receives negative acknowledgment from the receiver due to frame not been error free. This process of retransmission is called ARQ (Automatic Repeat Request). The set of rules that will determine the operations for the sender and the receiver are named the ARQ protocol. This ARQ protocol makes the network reliable, and that is, one of the important requirements of a network system if, data transmits from one node to another over the network and ensures that data received at receiver's site is complete and accurate. Here, we will refer to ACK, for positive acknowledgement (that is receiver has correct data) and NAK (REJect) to refer to negative acknowledgement (that is frame is received with some error). In this unit, you will study three commonly used methods for flow and error control that is Stop & Wait ARQ, GoBack-n ARQ and Selective Repeat ARQ.

2.1 OBJECTIVES

After going through this unit, you should be able to :

- define flow and error control;
- define is ARQ;

- define functionality of data link protocol;
- define Stop & Wait ARQ method for error and flow control;
- define GoBack-n ARQ method for error and flow control;
- perform selective Repeat ARQ method for error and flow control, and
- define pipelining.

2.2 STOP & WAIT ARQ

This is the simplest method for flow and error control. This protocol is based on the concept that, the sender will send a frame and wait for its acknowledgment. Until it receives an acknowledgment, the sender cannot send the next frame to the receiver. During transmission of frame over the network an error can appear.

Error can be due to a frame getting damaged/lost during transmission. Then, the receiver discards that frame by using error detection method. The sender will wait for acknowledgement of frame sent for a predetermined time (allotted time). If timeout occurs in the system then, the same frame is required to be retransmitted. Hence, the sender should maintain a duplicate copy of the last frame sent, as, in future it can be required for retransmission. This will facilitate the sender in retransmitting the lost/damaged frame.

At times the receiver receives the frame correctly, in time and sends the acknowledgment also, but the acknowledgment gets lost/damaged during transmission. For the sender it indicates time out and the demand for retransmission of the same frame appears in the network. If, the sender sends the last frame again, at the receiver's site, the frame would be duplicated. To overcome this problem it, follows a number mechanism and discards the duplicate frame.

Another situation when a error can appear in the network system is when the receiver receives the frame out of order, then it simply discards that frame and sends no acknowledgement. If, the acknowledgement is not received by the sender in time then, it assumes that the frame is lost during transmission, and retransmits it.

For distinguishing both data frame and acknowledgement frame, a number mechanism is used. For example A data frame 0 is acknowledged by acknowledgement frame 1, to show that the receiver has received data frame 0 and is expecting data frame1 from the sender.

Both the sender and the receiver both maintain control variable with volume 0 or 1 to get the status of recently sent or received. The sender maintains variable S that can hold 0 or 1 depending on recently sent frame 0 or 1. Similarly the receiver maintains variable R that holds 0 or 1 depending on the next frame expected 0 or 1.

Here, we are considering one directional information flow (Frame) and other direction control information (ACK) flow. If transmission of frames leads to an error then, the recovery process of the same demands, a retransmission strategy to be used that leads to four different possible outcomes, that are:

- Normal Operation
- When ACK is lost
- When frame is lost
- When ACK time out occurs

Normal Operation

If the sender is sending frame 0, then it will wait for ack 1 which will be transmitted by the receiver with the expectation of the next frame numbered frame1. As it receives ACK1 in time (allotted time) it will send frame 1. This process will be continuous till complete data transmission takes place. This will be successful transmission if ack for all frames sent is received in time. It is shown with the help of *Figure.1*

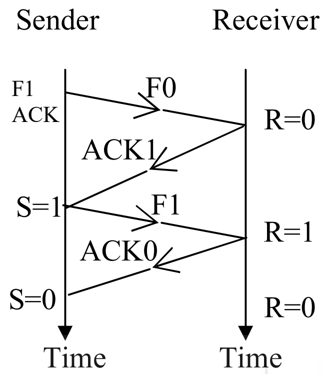

Figure 1: Stop and wait protocol

When ACK is lost

Here the sender will receive corrupted ACK1 for frame sent frame 0. It will simply discard corrupted ACK1 and as the time expires for this ACK it will retransmit frame 0. The receiver has already received frame 0 and is expecting frame1, hence, it will discard duplicate copy of frame 0. In this way the numbering mechanism solves the problem of duplicate copy of frames. Finally the receiver has only one correct copy of one frame. This is explained with the help of *Figure.2*.

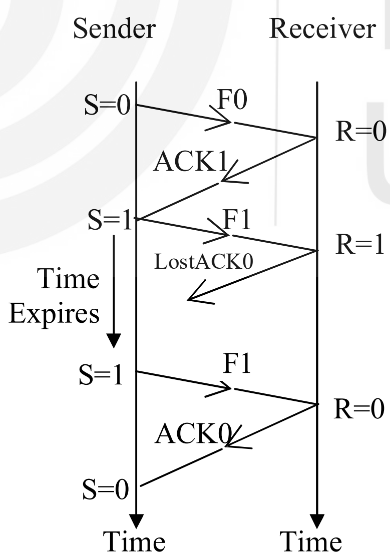

Figure 2: Loss of ACK

When Frame is lost

If the receiver receives corrupted/damaged frame1, it will simply discard it and assumes that the frame was lost on the way. And correspondingly, the sender will not get ACK0 as frame has not been received by the receiver. The sender will be in waiting stage for ACK0 till its time out occurs in the system. As soon as time out occurs in the system, the sender will retransmit the same frame i.e frame1(F1) and the receiver will send ACK0 in reply as shown in *Figure.3*.

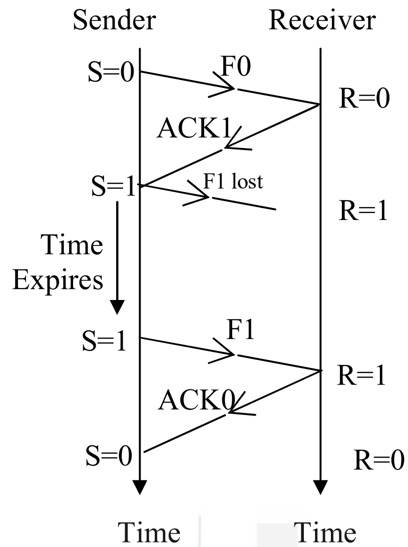

Figure 3: Loss of a frame

When ACK time out occurs

In this operation, the receiver is not able to send ACK1 for received frame0 in time, due to some problem at the receiver's end or network communication. The sender retransmits frame0 as ACK1 is not received in time. Then, the sender retransmits frame0 as ACK1 time expires. At the receiver end, the receiver discards this frame0 as the duplicate copy is expecting frame1 but sends the ACK1 once again corresponding to the copy received for frame0. At the sender's site, the duplicate copy of ACK1 is discarded as the sender has received ACK1 earlier as explained with the help of *Figure 4*.

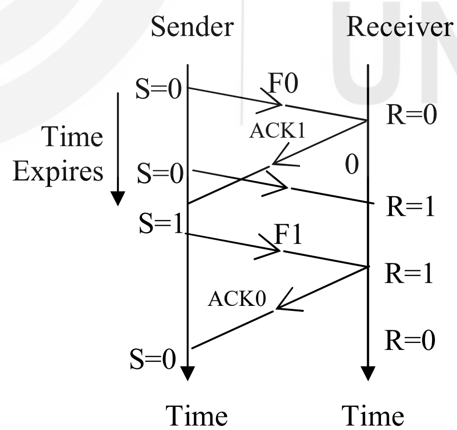

Figure 4: ACK time out

These operations indicate the importance of the numbering mechanism while, transmitting frames over the network. The method above discussed has low efficiency due to improper use of the communication channel. So, now, we will discuss the case if flow of data is bidirectional. In Bidirectional transmission both the sender and the receiver can send frames as well as acknowledgement. Hence, both will maintain S and R variables. To have an efficient use of bandwidth the ACK can be appended with the data frame during transmission. The process of combining ACK with data is known as **Piggybacking.** The concept of piggybacking is explained in a later section. This reduces transmission overhead and increases the overall efficiency of data transmission.

The process is shown in *Figure*.5.

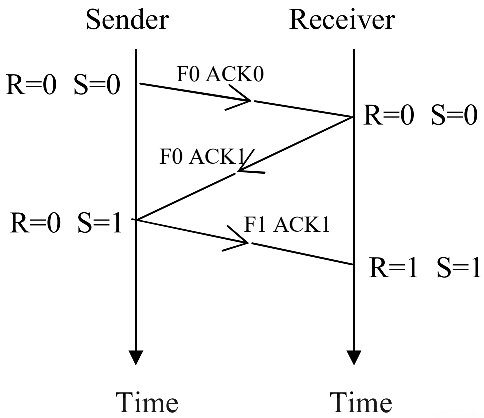

Figure 5: Piggybacking 1

The problem with stop and wait is that only one frame can be transmitted at a time and this leads to inefficiency of transmission. To deal with this, we have another error and flow control method that we will discuss in the next section.

2.3 GO-BACK-N ARQ

In this section, we will try to overcome the inefficient transmission that occurs in Stop & Wait ARQ. In this method, many frames can be transmitted during the process without waiting for acknowledgement. In this, we can send n frames without making the sender wait for acknowledgements. At the same time, the sender will maintain a copy of each sent frame till acknowledgement reaches it safely. As seen earlier in the Stop & Wait ARQ where, we used number mechanism, here also, we will use number method for transmission and receipt of frames. Each frame will have a sequence number that will be added with the frame. If, the frame can have k bit sequence number then the numbers will range between 0 to 2^k-1. For example if k is 2 bit then numbers will be 0,1,2,3,0,1,2,3,0,….

The sender can send 4 frames continuously without waiting for acknowledgment. But, the receiver will look forward to only one frame that must be in order. If, the frame received is not in order, it will simply keep on discarding the frame till, it receives the desired sequence number frame. The receiver is not bound to send an individual acknowledgment for all frames received; it can send a cumulative acknowledgment also. The receiver will send a positive acknowledgement if, the received frame is the desired sequence number frame. Otherwise, it will keep on waiting if, the frame received is corrupted or out of order. As soon as the timer expires for the frame sent by the sender, the sender will GoBack and retransmit all frames including the frame for which the timer expired, till, the last sent frame. Hence, it is named as Go-Back-N ARQ. The process of retransmission for an error frame is shown in *Figure.6.*Go-Back-N is used in HDLC protocol.

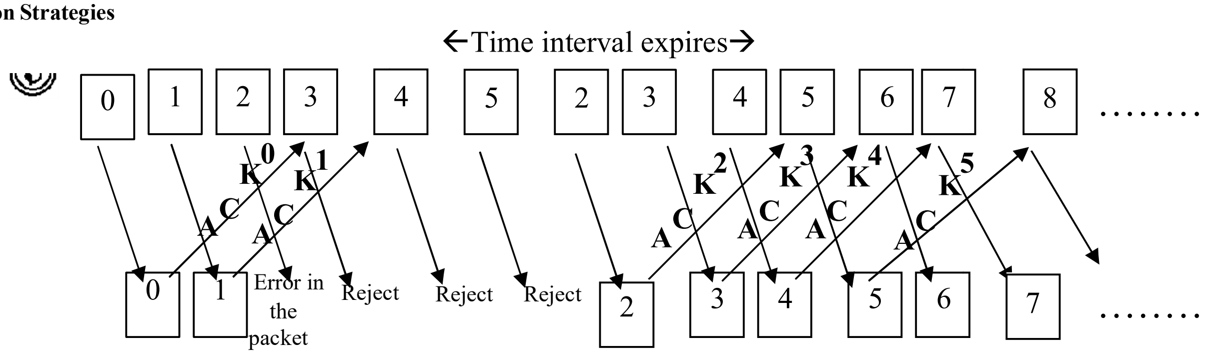

Figure 6: Go-Back-N

If, the error rate is high, then, this leads to a lot of wastage of bandwidth as the sender will retransmit all the frames from, the frame in which error appears till the last sent. To increase the efficiency of transmission, when the error rate is high, another protocol called Selective Repeat ARQ is used which is discussed in the next section.

2.4 SELECTIVE REPEAT ARQ

This method increases the efficiency of the use of bandwidth. In this method, the receiver has a window with the buffer that can hold multiple frames sent by the sender. The sender will retransmit only that frame which has some error and not all the frames as in Go-Back-N ARQ. Hence, it is named as selective repeat ARQ. Here, the size of the sender and the receiver window will be same. The receiver will not look forward only to one frame as in Go-Back-N ARQ but it will look forward to a continuous range of frames. The receiver also sends NAK for the frame which had the error and required to be retransmitted by the sender before the time out event fires. As in the earlier section we discussed, each frame will have a sequence number that will be added with the frame. If, the frame can have k bit sequence number then the sequence number of frames will range between 0 to 2^k-1. For example, if k is 2 bit then numbers will be 0,1,2,3,0,1,2,3,0,….Size of sender and receiver window would be $2^k/2$ i.e 4/2=2 or it can be written as 2^{k-1}. If the window size is 2 and acknowledgement for frame0 and frame1 both gets lost during transmission then, after timer expires the sender will retransmit frame0 though the receiver is expecting frame2 after frame1 was received without any error. Hence, frame0 will be discarded by the receiver as a duplicate frame. If the receiver window size is more than two, the receiver will accept duplicate frame0 as a new frame and hence, the size of window should be $2^k/2$. The process is shown in *Figure.7.*

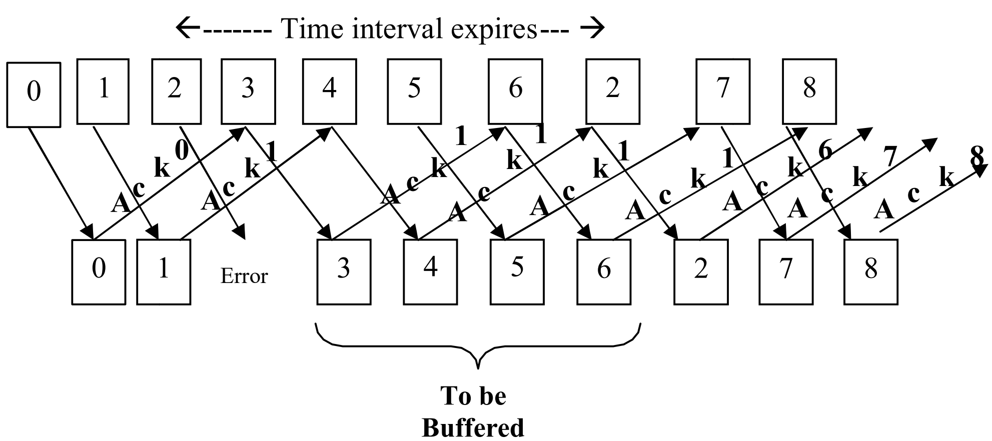

Figure 7: Selective Repeat

TCP uses Selective repeat ARQ strategy to deal with flow and error control

If, we consider bidirectional transmission i.e., data and acknowledgement flow from both sender and receiver then, the concept of Piggybacking can be used in a similar fashion as already discussed in Stop & Wait ARQ method, in order to better utilise bandwidth.

2.5 PIPELINING

Pipelining in the network is one task that starts before the previous one is completed. We might also say that the number of tasks is buffered in line, to be processed and this is called pipelining. For example, while printing, through the printer before one task printing is over we can give commands for printing second task. Stop & Wait ARQ does not use pipelining. As in Stop & Wait ARQ the sender cannot send the next frame till it receives acknowledgement for the frame sent. Here, Pipelining is used in Go-Back-N ARQ and Selective repeat ARQ as both methods can send multiple frames without holding the sender for receiving the acknowledgement for frame sent earlier. This process of pipelining improves the efficiency of bandwidth utilisation.
Now, we will explain with the help of *Figure 8* how pipelining is used in Go-Back-N ARQ.

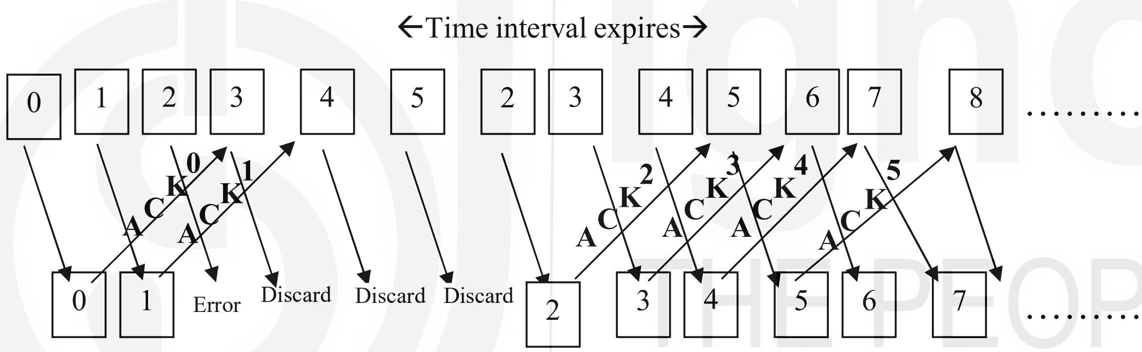

Figure 8: Pipelining in Go-Back-N

Here frame0 is received by the receiver and without waiting for acknowledgment of frame 0 sent at sender site, the sender is allowed to send frame1. This process is known as pipelining.

Similarly, selective repeat pipelining is used as shown in *Figure 9* below.

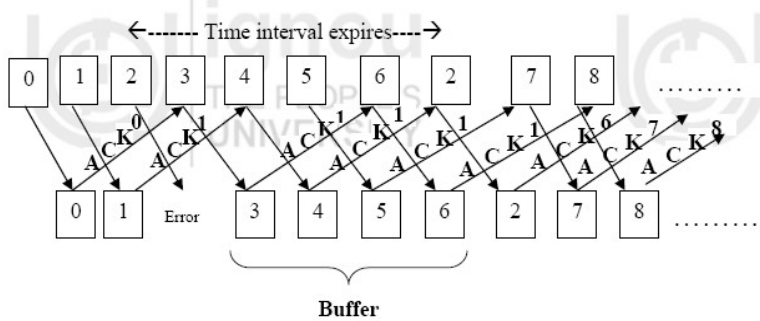

Figure 9: Pipelining in selective repeat

In this example we can show how pipelining increases the overall utilisation of bandwidth. Frame0 and frame1 are sent in continuous order without making the sender wait for acknowledgment for frame0 first.

☞ Check Your Progress 1

1) Define flow and error control.

…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………

2) Explain Go-Back-N ARQ and Selective repeat ARQ are better methods for retransmission.

…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………

3) Give an example where pipelining can be applied.

…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………

4) What is the significance of control variables S and R?

…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………

2.6 PIGGYBACKING

Piggybacking is the process which appends acknowledgement of frame with the data frame. Piggybacking process can be used if Sender and Receiver both have some data to transmit. This will increase the overall efficiency of transmission. While data is to be transmitted by both sender and receiver, both will maintain control variable S and R. We will explain the process of piggybacking when the sender and the receiver both are transmitting data with the help of *Figure 10.*

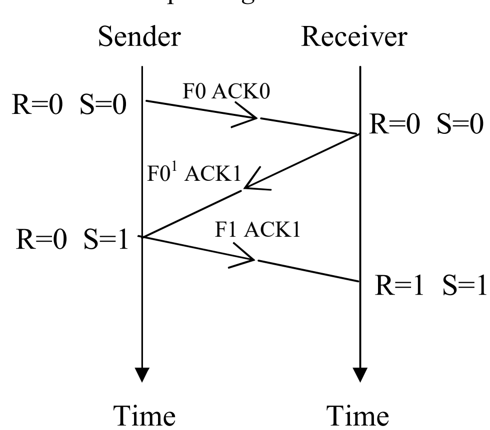

Figure 10: Piggybacking

Here, both the sender and the receiver maintain control variables S and R. The sender sends frame 0 (F0) with ACK0 appended along with it. Similarly, the receiver sends Frame 0(F0) with ACK1 appended to it. This way transmitting both frame and acknowledgement will concurrently increase optimal efficiency of bandwidth utilisation because piggybacking will get a free side.

2.7 SUMMARY

This unit focuses on one prime function of the Data link layer that is flow and error control for achieving the goal of reliable data transmission over the network. For flow and error control retransmission strategies are considered. Flow control specifically talks about the speed of sending the frame by the sender and processing the received frame by the receiver. The speed for the sender and the receiver must be compatible. So, that all frames can be received in order and processed in time. Error control technique combines two processes error detection and error correction in the data frame. In Stop & wait ARQ protocol sender waits for acknowledgment for the last frame sent. After the acknowledgment is received by the sender then only the next frame can be sent. In Go-Back-N ARQ frames can be sent continuously without waiting for the sender to send the acknowledgement. If an error is found in any frame then frames received after that will be discarded by the receiver. Retransmission of frames will start from the error frame itself. In selective repeat Process frames can be sent continuously. But here, the receiver has a buffer window that can hold the frames received after the error frame. Hence, retransmission will be only for error frame. This increases the efficiency of data transmission on a network. For better utilisation of bandwidth, piggybacking can be used. Piggybacking process appends acknowledgement along with the data frame. This will be efficient for bidirectional communication.

2.8 SOLUTIONS/ANSWERS

Check Your Progress 1

1) Flow Control: A method to control rate of flow of frames .
Error Control: A method to detect and handle the errors during data transmission.

2) Go-Back-N ARQ and Selective repeat ARQ are better method for retransmission as they keep the copy of each sent frame till the frame is acknowledged by receiver. Go-Back-N ARQ retransmits all frames in which there is an error and the following frames. Selective repeat ARQ retransmit only the error frame.

3) Pipelining is used in Go-Back-N ARQ and Selective repeat ARQ because many frames can be sent without waiting for previous frame acknowledgement.

4) Numbering system that is used by sender and receiver by using variable S and R that can hold 0 or 1 removes the occurrence of possible ambiguities like either duplicate frame or duplicate acknowledgement.

2.9 FURTHER READINGS

1) *Computer Networks,* Andrew S. Tanenbaum, 4th Edition, Prentice Hall of India, New Delhi.

2) *Data and Computer Communications,* William Stalling, 5th Edition,Prentice Hall of India, New Delhi.

3) *Data Communications and Networking,* Behrouz A. Forouzan, 3rd Edition, Tata McGraw Hill, New Delhi.

4) *Communication Networks, Fundamental Concepts and Key Architectures,* Leon Garcia and Widjaja, 3rd Edition, Tata McGraw Hill, New Delhi.