# Experiment 5: ARQ Mechanisms in DLL

**Aim:** To implement receiver algorithms for the different ARQ mechanisms at the Data Link Layer

**Objective:** After carrying out this experiment, students will be able to:

* implement receiver algorithms for the different ARQ mechanisms at the Data Link Layer
* Analyze the differences between the ARQ mechanisms

**Problem statement:** You are required to write a program that can receive frames at the data link layer. Assume that the user is entering the frames as the transmitter. You are required to implement stop and wait, go back N and selective repeat ARQ mechanisms. Consider that you have to transmit and receive a total of 20 frames using WT=WR=1, WT=5 and WR=1 and WT=WR=5 for stop and wait, go back N and selective repeat respectively

**Analysis:** While analyzing your program, you are required to address the following points:

* Difference between stop and wait, go back N and selective repeat.
* Comparison of the disadvantages of the different ARQ mechanisms.

**MARKS DISTRIBUTION**

|  |  |  |
| --- | --- | --- |
| **Component** | **Maximum Marks** | **Marks Obtained** |
| Preparation of Document | 7 |  |
| Results | 7 |  |
| Viva | 6 |  |
| **Total** | **20** |  |

Submitted by: Subhendu Maji

Register No: 18ETCS002121

1. Algorithm/Flowchart

The algorithm for Go-Back-N is as follows:

STEP 1: Start

STEP 2: initialize variables

STEP 3: numOfFrames, windowSize from user

STEP 4: while count is less than numOfFrames, do

4.1: numOfTransInCurrentWindow 0

4.2: for j from count to numOfFrames and count + windowSize, do

4.2.1: display transmission message

4.2.1: totalTransmissions++

4.3: for j from count to numOfFrames and count + windowSize, do

4.3.1: flag random number % 2

4.3.1: if flag is 0, do

4.3.1.1: display success message

4.3.1.2: numOfTransInCurrentWindow++

4.3.2: else, display failure message, break

STEP 5: display results

STEP 6: Stop

Algorithm for selective repeat is as follows:

STEP 1: Start

STEP 2: initialize variables

STEP 3: numOfFrames, windowSize from user

STEP 4: while count is less than numOfFrames, do

4.1: numOfTransInCurrentWindow 0

4.2: for j from count to numOfFrames and count + windowSize, do

4.2.1: display transmission message

4.2.1: totalTransmissions++

4.3: for j from count to numOfFrames and count + windowSize, do

4.3.1: flag random number % 2

4.3.1: if flag is 0, do

4.3.1.1: display success message

4.3.1.2: numOfTransInCurrentWindow++

4.3.2: else, display failure message, retransmit message

STEP 5: display results

STEP 6: Stop

Algorithm for stop and wait is as follows:

STEP 1: Start

STEP 2: initialize vars

STEP 3: numOfFrames from user

STEP 4: while count is less than numOfFrames, do

4.1: display sending message

4.2: flag = random number % 2

4.3: if flag is 0, display success, count++

4.4: else, display failure, retransmitting message

STEP 5: Stop

1. Program



Figure 1 Go Back N Source Code



Figure 2 Selective Protocol Source Code



Figure 3 Stop and Wait Source Code

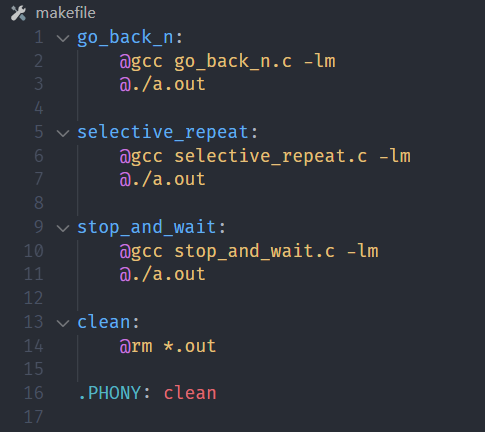


Figure 4 Makefile

1. Results

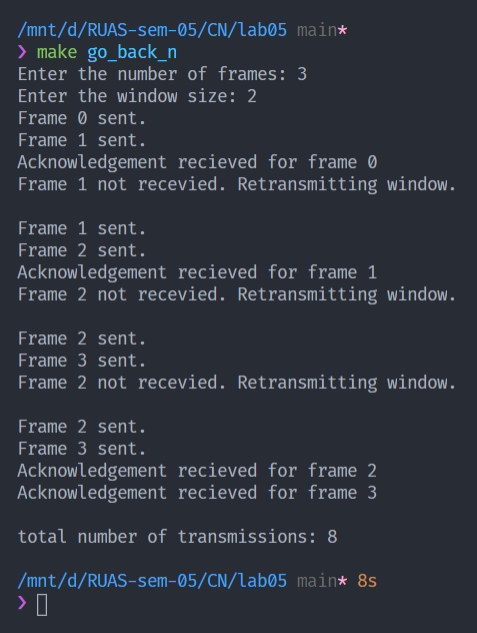


Figure 5 Execution of Go Back N

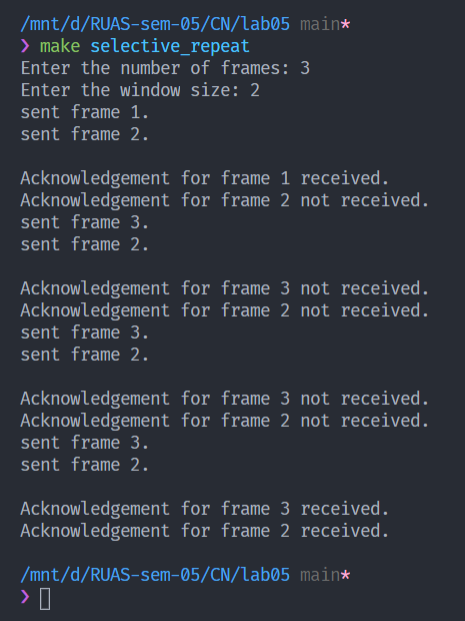


Figure 6 Execution of Selective Protocol

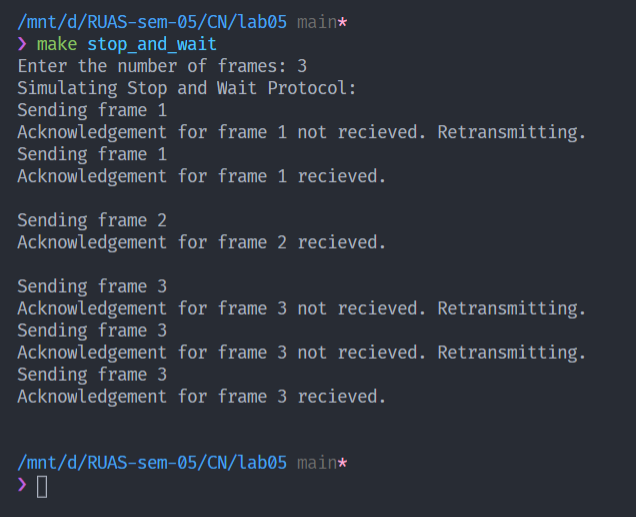


Figure 7 Execution of Stop and Wait

1. Analysis and Discussions

**Stop and Wait ARQ** mechanism sends frames one by one and waits for a certain time for the receiver to receive those frames. Requires a lot of bandwidth and is very inefficient. If a frame doesn’t reach the receiver in time, or is lost in the process or is corrupted, Stop and Wait mechanism, retransmits that frame. Both frames and acknowledgment (ACK) are numbered alternately 0 and 1. The sender keeps a copy of the sent frame transmitted until it receives an acknowledgment for reference.

**Go Back-N ARQ** mechanism uses sliding window protocol which makes it so that we send multiple frames in one go till the size of the sliding window. The sliding window is of the size on transmitter side is 2n-1. In Go Back-N, we send multiple frames in the sliding window at once and the receiver sends an acknowledgement of the next expected frame. If a frame is damaged or out of order, the receiver is silent and will discard all subsequent frames. In this protocol, send frames are numbered sequentially. The window size of the receiver is 1. Go Back-N suffers in a noisy channel. It also requires more bandwidth when we discard frames because of faults in between.

**Selective Repeat ARQ** mechanism also uses sliding window protocol. It even works in noisy channel and resends only the damaged frame. It also has the concept of negative acknowledgement (NACK) that reports sequence of damaged frames before timer expires. The sliding window’s size for both transmitter and receiver are 2m/2. Rest of the working is same as Go Back-N.

1. Conclusions

The Go Back-N, Selective Repeat and Stop and Wait ARQ mechanisms were successfully learned and simulated via program written in C.