**ISTANBUL TECHNICAL UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**BLG 433E**

# COMPUTER COMMUNICATIONS TERM PROJECT REPORT

**GROUP MEMBERS:**

|  |  |  |
|---|---|---|
| 150200060 | : | ÖMER & YILDIZ |
| 150210079 | : | SÜLEYMAN CEYHUN & DEMİR |

**SPRING 2024-2025**

## Contents

## 1 Introduction

This report details the implementation and performance evaluation of a reliable data transfer protocol over UDP, utilizing the Selective Repeat ARQ method. The protocol simulates a reliable communication channel over an unreliable network, where packets, including control packets (such as Handshake and ACK packets), may be lost. The goal is to implement a reliable transport protocol on top of UDP and evaluate its performance under various conditions, including packet loss rates and window sizes. The server uses Selective Repeat to send packets, and both client and server must handle lost packets through timeouts and retransmissions. The evaluation focuses on the impact of different error rates and window sizes on transmission efficiency.

## 2 Protocol Overview

The protocol follows a connection-oriented model, where both the client and the server establish a logical connection before sending data packets. The server sends segments of a text file to the client, handling packet loss by retransmitting lost packets. The client sends acknowledgment (ACK) packets for each received segment. The Selective Repeat scheme allows pipelining of multiple segments before requiring an acknowledgment for each segment.

### 2.1 Key Features

- **Error Handling:** Packet loss is controlled by the unreliableSend function, which simulates the loss of data and control packets with specified error rates.
- **Selective Repeat Protocol:** The server sends multiple packets at once (up to the window size), and the client acknowledges each packet. Lost packets are retransmitted after a timeout.
- **Handshake and FIN Protocol:** A triple handshake is used to establish the connection, and a FIN packet indicates the end of data transmission.
- **Timeouts and Retransmissions:** The server detects lost packets using a timeout mechanism. If an ACK for a data packet is not received within the specified timeout period, the packet is retransmitted.

## 3 Implementation Details

The implementation consists of two main Python scripts: `client.py` and `server.py`, which simulate the client-server communication based on the Selective Repeat protocol.

### 3.1 Client Implementation

The client is responsible for initiating the connection, handling the handshake, and acknowledging the receipt of data packets. The client performs the following tasks:

1. Sends the filename to the server to initiate the connection.
2. Receives data packets from the server, sends ACKs for each successfully received packet.
3. Handles packet loss by waiting for a timeout and retransmitting lost ACK packets if necessary.
4. Completes the session by receiving a FIN packet from the server and responding with an ACK and FIN packet.

### 3.2 Server Implementation

The server manages the connection, sends data packets, and handles retransmissions for lost packets. The server performs the following tasks:

1. Waits for the client's handshake request and validates the filename.
2. Splits the file into segments and sends them using the Selective Repeat protocol, maintaining a sliding window of packets.
3. Detects lost packets by checking for timeouts and retransmits any packets that were not acknowledged within the timeout period.
4. Sends a FIN packet to the client to signal the end of transmission once all data has been sent.

## 4 Experimental Setup

The experiments were conducted under different conditions to evaluate the performance of the protocol. The following parameters were used:

- **Error Rate:** The probability of packet loss during transmission. The error rate takes values of 1%, 5%, 10%, and 20%.
- **Window Size:** The number of packets sent by the server before requiring an acknowledgment. The window size takes values of 1, 10, 50, and 100.

### 4.1 Performance Metrics

The following metrics were used to evaluate the performance of the protocol:

- **Total Transmission Time:** The total time taken to successfully transmit all packets.
- **Retransmissions:** The number of packets that were retransmitted due to timeouts or errors.

## 5 Results and Analysis

The experiments were run for different error rates and window sizes. The results are presented below.

### 5.1 Effect of Error Rate

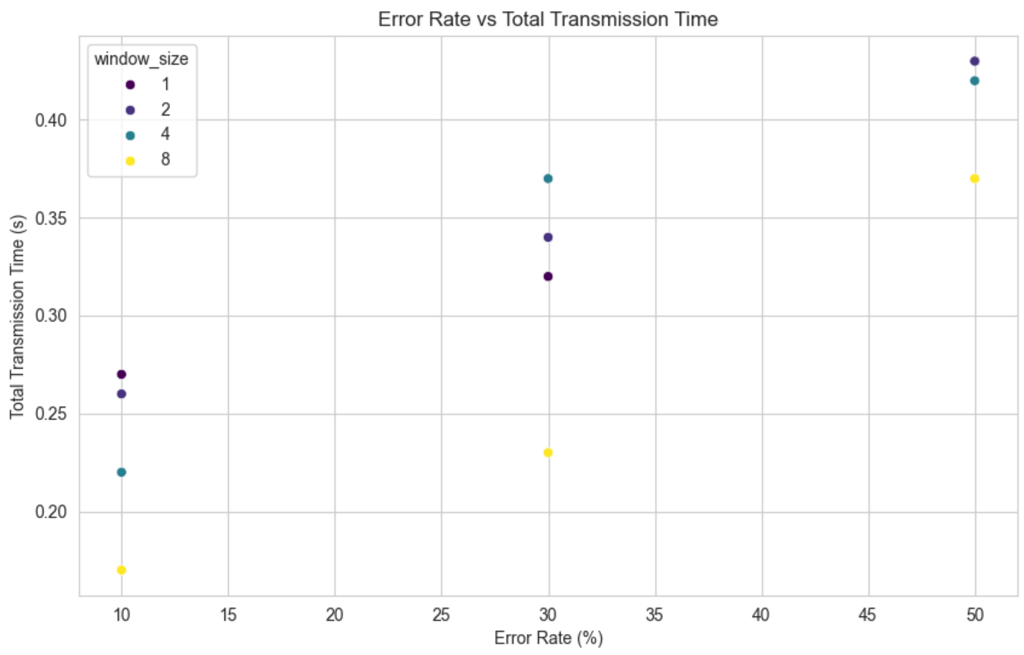

Figure 1: Total Transmission Time vs. Error Rate

As shown in Figure 1, the total transmission time increases as the error rate increases. Higher error rates result in more lost packets and, consequently, more retransmissions, which increases the total transmission time.

## 5.2 Effect of Window Size

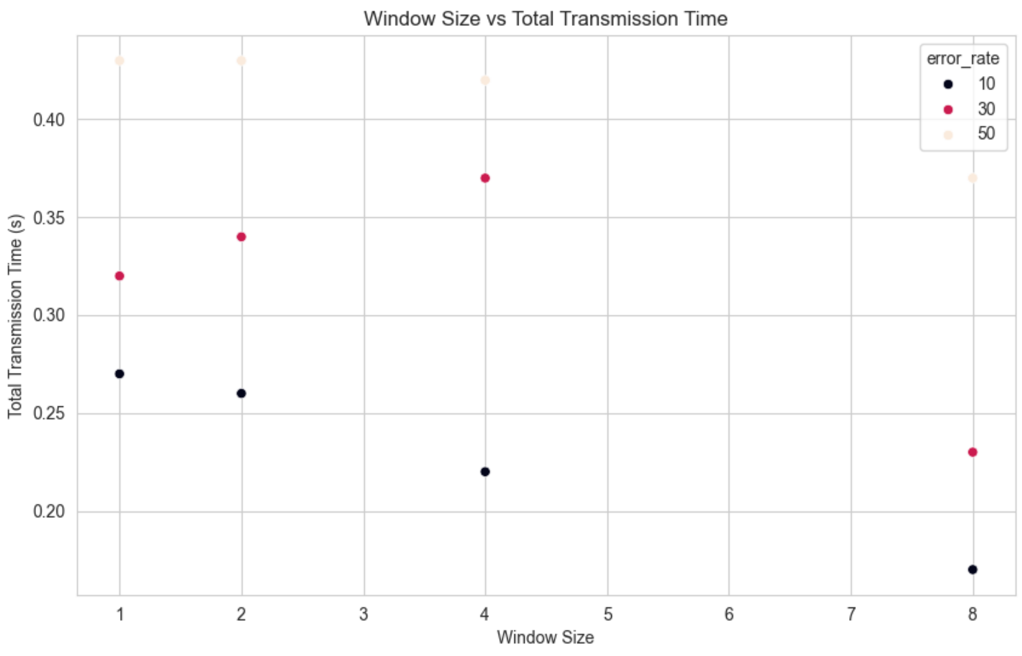

Figure 2: Total Transmission Time vs. Window Size

Figure 2 illustrates that larger window sizes lead to improved transmission efficiency. With a larger window, the server can send more packets without waiting for an acknowledgment, reducing the idle time and speeding up the transmission.

## 5.3 Retransmissions

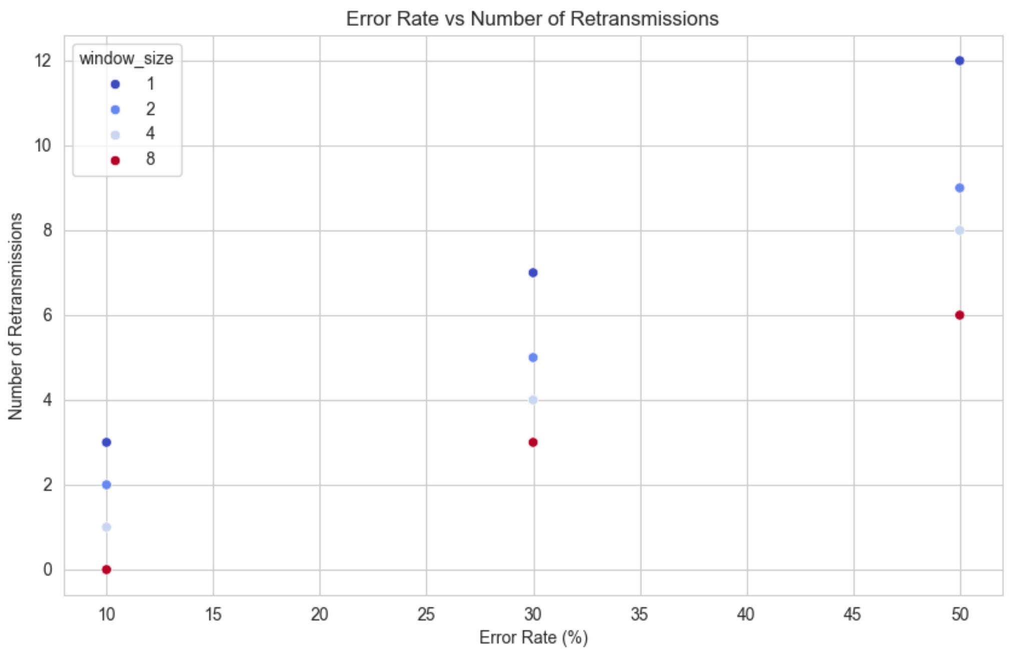

Figure 3: Retransmissions vs. Error Rate

Figure 3 shows that the number of retransmissions increases with the error rate. This is expected, as more packet losses require retransmissions, which impacts the efficiency of the protocol.

# 6 Conclusion

The implementation of the Selective Repeat ARQ protocol over UDP successfully handles packet loss and provides reliable data transmission. The performance of the protocol is significantly affected by the error rate and window size. Higher error rates result in more retransmissions and longer transmission times. Optimizing the window size can help improve efficiency, especially in networks with higher error rates. Further optimizations could focus on improving the timeout mechanism and handling varying network conditions more dynamically.