# Concurrent and Distributed Programming

# Homework 1

## Introduction

This document presents a detailed analysis of data transmission time and reliability regarding two fundamental Internet protocols: Tr**ansmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)**.

The following tests compares the performance impacts of using **streaming** and **stop-and-wait** mechanisms under various conditions, including different message sizes and numbers of clients.

## Setup

* **Hardware**: Tests were conducted on a server with an AMD Ryzen 7 processor, 16GB RAM, and a Gigabit Ethernet connection.
* **Software and Technologies**: The server and clients were implemented in C# .NET 8.0 and ran on a Linux operating system.
* **System architecture**: The project has a two-tier architecture, with direct communication between clients and the server, to minimize complexity.
* **Network**: A local area network (LAN).

## Project Structure

The project was organized into three separate but interconnected projects to facilitate ease of testing, and modularity.

**Server and Client Projects**

Both projects implement each protocol (TCP and UDP), encapsulated within its class (TCPClient, TCPServer, UDPClient, UDPServer). Additional logic was added to ensure the changes to the mechanism used.

**Test Project**

A separate project was dedicated to defining and executing the tests. This project references the client and server projects to initiate test scenarios and easily adjust test parameters, including varying message sizes, client counts, and protocol mechanisms.

## Testing

Each test was executed with the following parameters:

* Protocols: TCP and UDP
* Mechanisms: Streaming and Stop-and-Wait
* Message Sizes: Varied across tests from 16384 bytes to 65535 bytes
* Total bytes to send: Ranged from 1 MB to 1 GB
* Client Counts: Single and multiple clients (up to 10)

Tests measured transmission time, number of messages sent and received, total bytes sent and received.

## Results

## **TCP Transmission Times and Reliability**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mechanism | Nr of  Clients | Message  Size | Total Bytes  To Send | Transmission  Time (ms) | Messages  Sent | Bytes  Sent | Messages  Read | Bytes  Read |
| Streaming | 1 | 65535 | 1048576 bytes = 1 MB | 1 | 17 | 1114095 | 17 | 1114095 |
| Streaming | 1 | 65535 | 524288000  bytes = 500 MB | 464 | 8001 | 524345535 | 8098 | 524345535 |
| Streaming | 1 | 65535 | 1073741824  bytes = 1 GB | 917 | 16385 | 1073790975 | 16821 | 1073790975 |
| Stop-and-Wait | 1 | 65535 | 1048576 bytes  = 1 MB | 1 | 17 | 1114095 | 17 | 1114095 |
| Stop-and-Wait | 1 | 65535 | 524288000  bytes = 500 MB | 764 | 8001 | 524345535 | 8005 | 524345535 |
| Stop-and-Wait | 1 | 65535 | 1073741824  bytes = 1 GB | 1455 | 16385 | 1073790975 | 16388 | 1073790975 |
| Streaming | 10 | 65535 | 1048576 bytes  = 1 MB | Average  of 5 | 17  Per client | 1114095  Per client | 17  Per client | 1114095  Per client |
| Stop-and-Wait | 10 | 65535 | 1048576 bytes  = 1 MB | Average  of 30 | 17  Per client | 1114095 Per client | 17  Per client | 1114095  Per client |

## **UDP Transmission Reliability**

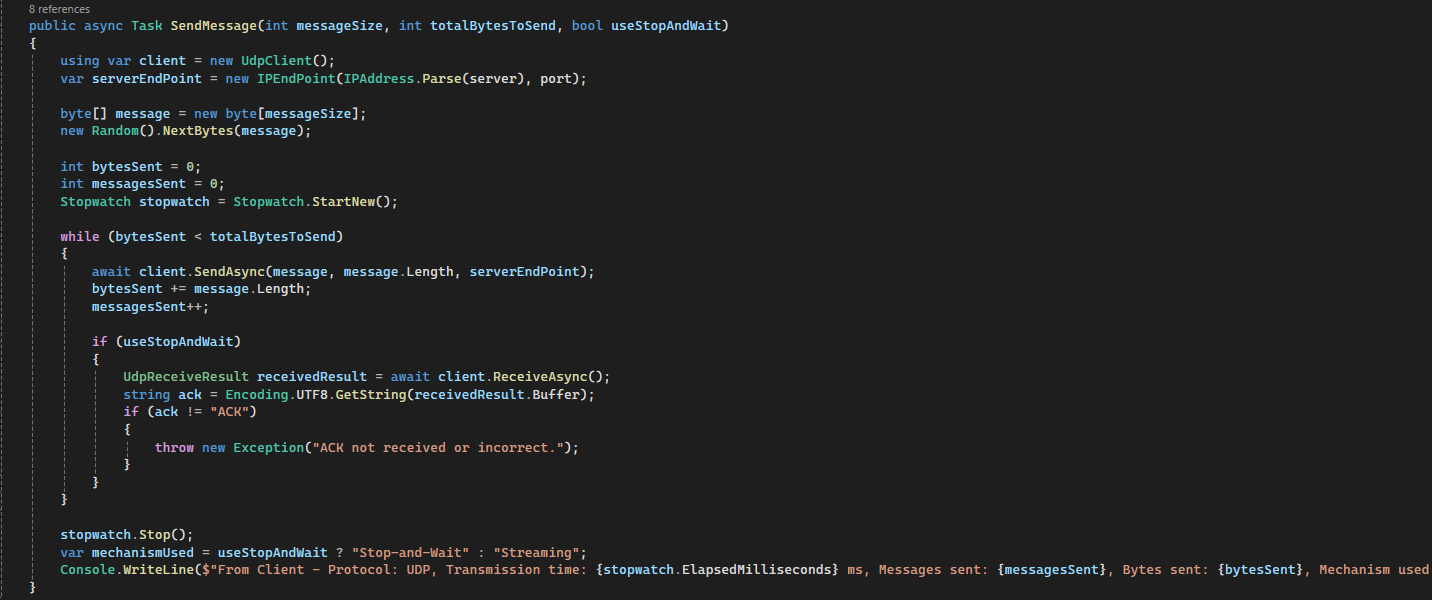
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mechanism | Nr of  Clients | Message  Size | Total Bytes  To Send | Transmission  Time (ms) | Messages  Sent | Bytes  Sent | Messages  Read | Bytes  Read |
| Streaming | 1 | 65535 | 1048576 bytes = 1 MB | 1 | 64 | 1048576 | 86751  From  Total of 97600 | 1421328384  From  Total of  1599078400 |
| Streaming | 1 | 65535 | 524288000  bytes = 500 MB | 842 | 32000 | 524288000 |
| Streaming | 1 | 65535 | 1073741824  bytes = 1 GB | 1716 | 65536 | 1073741824 |
| Stop-and-Wait | 1 | 65535 | 1048576 bytes  = 1 MB | 30 | 64 | 1048576 | 97600  From  Total of  97600 | 1599078400  From  Total of  1599078400 |
| Stop-and-Wait | 1 | 65535 | 524288000  bytes = 500 MB | 8288 | 32000 | 524288000 |
| Stop-and-Wait | 1 | 65535 | 1073741824  bytes = 1 GB | 15502 | 65536 | 1073741824 |
| Streaming | 10 | 65535 | 1048576 bytes  = 1 MB | Average  of 0 | 640 | 10485760 | 388 | 6356992 |
| Stop-and-Wait | 10 | 65535 | 1048576 bytes  = 1 MB | Average  of 62 | 640 | 10485760 | 640 | 10485760 |

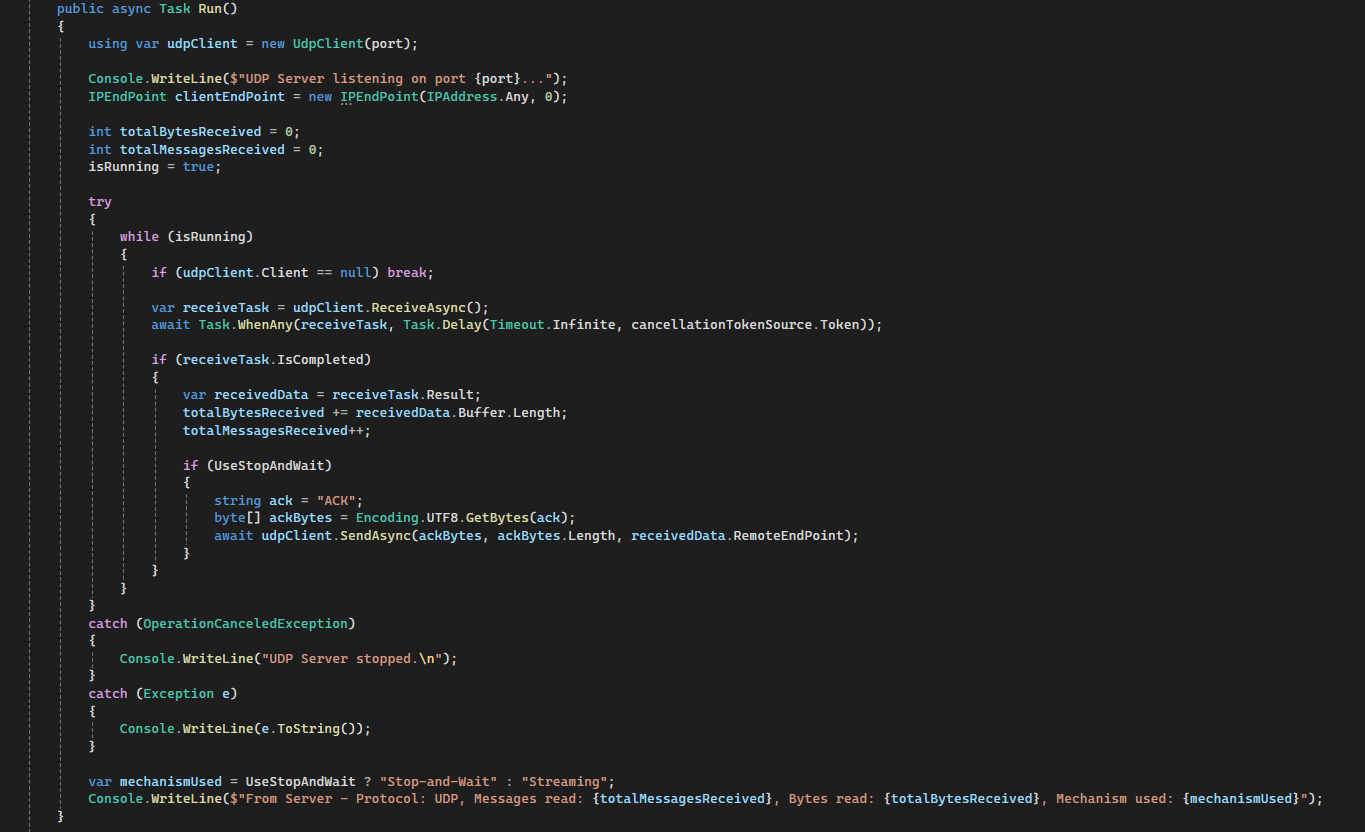
## Observations

* TCP vs. UDP: TCP consistently delivered all bytes sent, due to its built-in mechanisms for ensuring data integrity and order. UDP, while faster, showed data loss in streaming mode even when there was a single client involved and transmitted large amount of data.
* Streaming vs. Stop-and-Wait: For TCP, stop-and-wait increased transmission times as expected, due to waiting for acknowledgments. For UDP, stop-and-wait significantly improved reliability (it was able to read all bytes sent) at the cost of increased transmission time.
* Impact of Multiple Clients: TCP handled multiple clients without data loss, but increased client count led to slight increases in transmission time. UDP showed considerable data loss under streaming with multiple clients, which was eased by stop-and-wait.

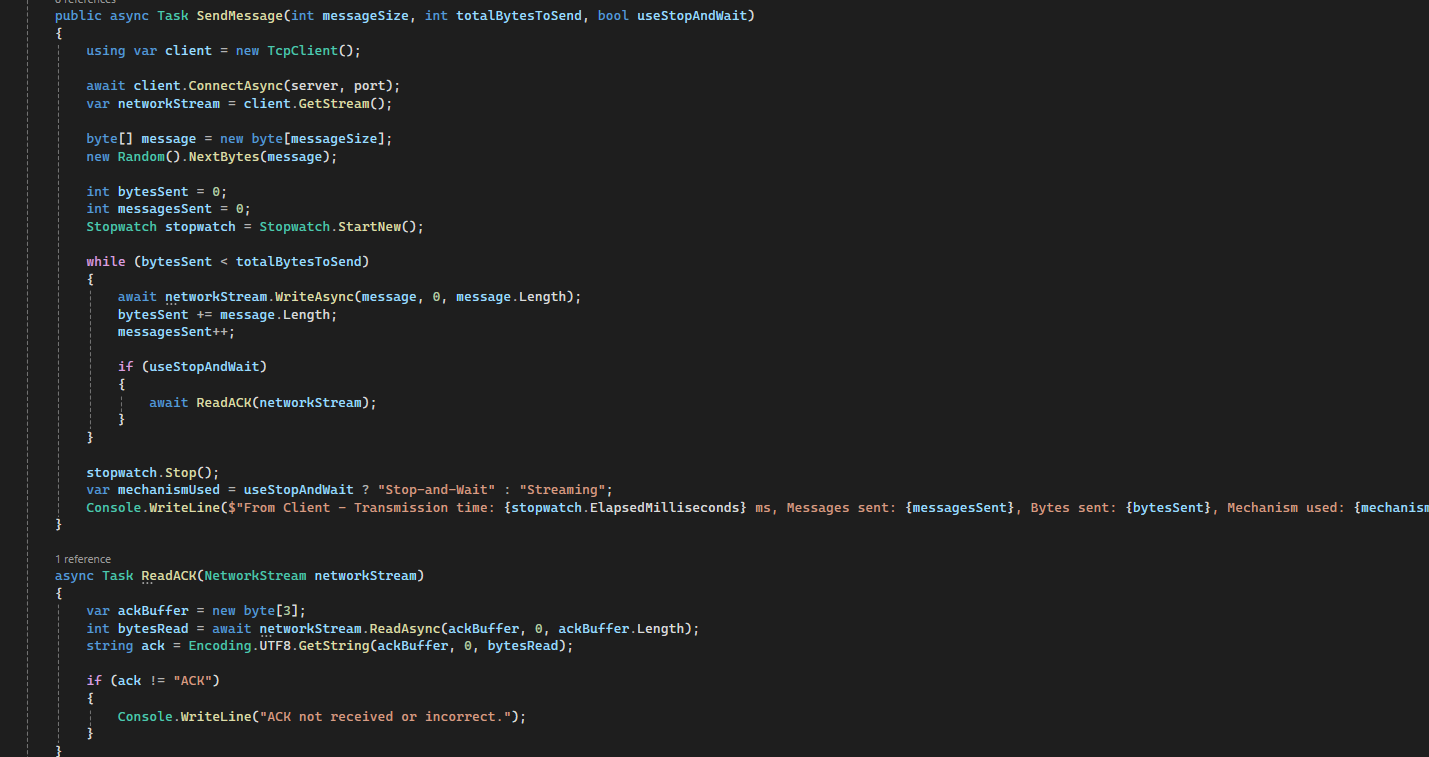
## Code Snippets

## UDP Client and Sever





## TCP Client and Server





## Conclusion

The tests show that TCP is very reliable for sending data where it's important that all data gets delivered correctly. Using stop-and-wait with UDP can also make it reliable but makes sending data slower. Whether to use TCP or UDP, and to use streaming or stop-and-wait, really depends on what you need more: speed, making sure data is received correctly, or managing many connections at once