# Sliding Window Protocol

By Joseph Collora

Table of Contents

[Sliding Window Protocol 1](#_Toc96596767)

[Code Documentation 1](#_Toc96596768)

[*Stop and Wait* 1](#_Toc96596769)

[*Sliding Window* 1](#_Toc96596770)

[*Case 4 Packet Dropping* 2](#_Toc96596771)

[Execution Output 2](#_Toc96596772)

[Performance Evaluation 2](#_Toc96596773)

[Discussion 2](#_Toc96596774)

## Code Documentation

There are four files included that use two distinct algorithms that have been evaluated when transferring 20 thousand packets on 1Gbps networks. This section will review the implementation of these two algorithms below, starting with stop and wait:

### *Stop and Wait*

The client will be discussed first; the name of the algorithm is self-explanatory. For every package the client sends out, there will be a wait for a response. When an acknowledgement comes, the next packet is sent. Once the timer runs out and a timeout occurs, the packet is retransmitted which is waited on again. The server functions buy sending an acknowledgement to the client only when the sequence matches from the client sent package to the server’s internal count of what the incoming sequence number is.

Below are diagrams for both client and server side, “stop and wait” algorithms:

### *Sliding Window*

Secondly, the sliding window algorithm. The client for this protocol will continue to send out messages until the last item in the current “window” is sent. While this is happening a timer is running, if it runs out and the number of unacknowledged sends is still within the window the client resends. If a correct acknowledgement is received, then the current “window” is moved once, and the number of unacknowledged packets goes down.

Below are diagrams for both client and server side, “sliding window” algorithms:

### *Case 4 Packet Dropping*

Within the driver source file, “hw3case4.cpp,” the code was changed to generate a random chance to drop the data or not. Within the modified udp source file for case 4, the server was changed to use the new dropP (percent) parameter instead of window size to drop the data.

## Execution Output

(Server Above, Client Below)

## Performance Evaluation

The included performance figures will be explored in the discussion section below.

Figure 1:

## Discussion

The difference in performance between “stop-and-wait” and “sliding window”

Influence of window size on “sliding window”

Differences in the number of re-transmits are between the two algorithms

The effects of drop rate on window size 1 and 30