

UDP - Reliable

Contributors

- **Ahmed Essam Hassan Abd-ElRazek - 19015289**
- **Mohammed Ayman Saaed Mahmoud - 19016250**

Overview

Suppose you've a file and you want to send this file from one side to the other (server to client). You will need to split the file into chunks of data of fixed length and add the data of one chunk to a UDP datagram packet in the data field of the packet. You need to implement TCP with congestion control.

UDP - Server

Pseudocode

1. Listen for requests on the predefined port.
2. Get client IP address and port number.
3. If it's a new connection, fork a new child to handle it. Otherwise, route the received packet to the appropriate child via IPC.
 - a. Open the file to be sent.
 - b. Receive ack 0 from the client which is equivalent to ack a handshake.
 - c. Transmit all the window determined by CWND.
 - d. Wait for the window to be completely acknowledged.
 - e. Go to step (c) if more bytes are remaining.

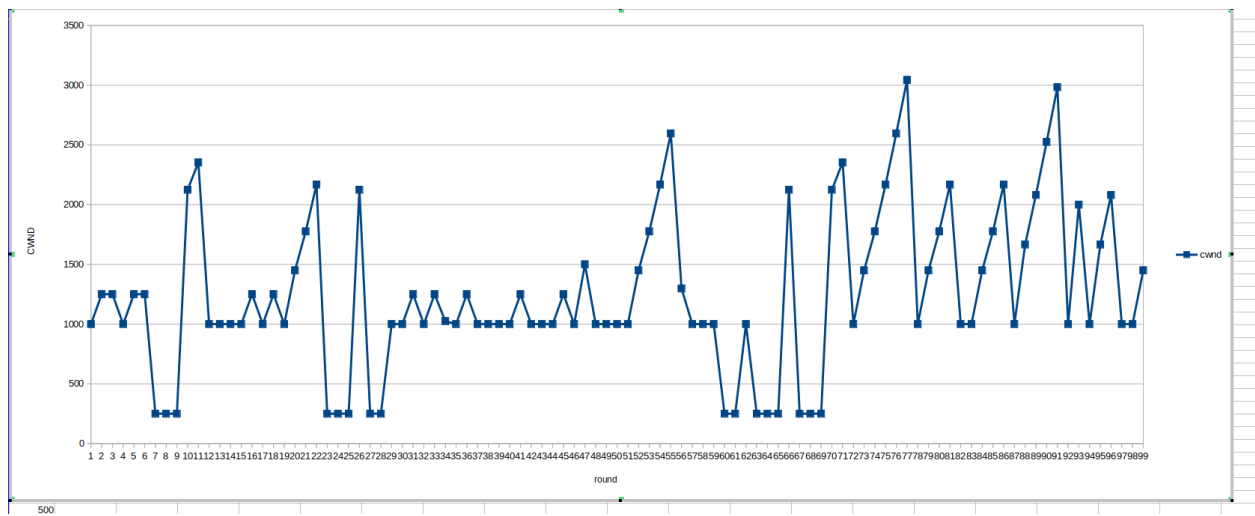
UDP - Client

Pseudocode

1. Open socket
2. Connect with server with ip address taken from arguments
3. Send packet that initialize the new connection
4. Receive packet from server that have file-size
5. Maintain variable last ack have first value = 1 to request using ack
6. Receive packets sent by server and save it and maintain sent array that tell us that this packet is sent
7. Always ack using last ack
8. Update last ack when seq no received is equal to last ack
9. Update until unsent packet
10. Do until last packet

file size = 5076501

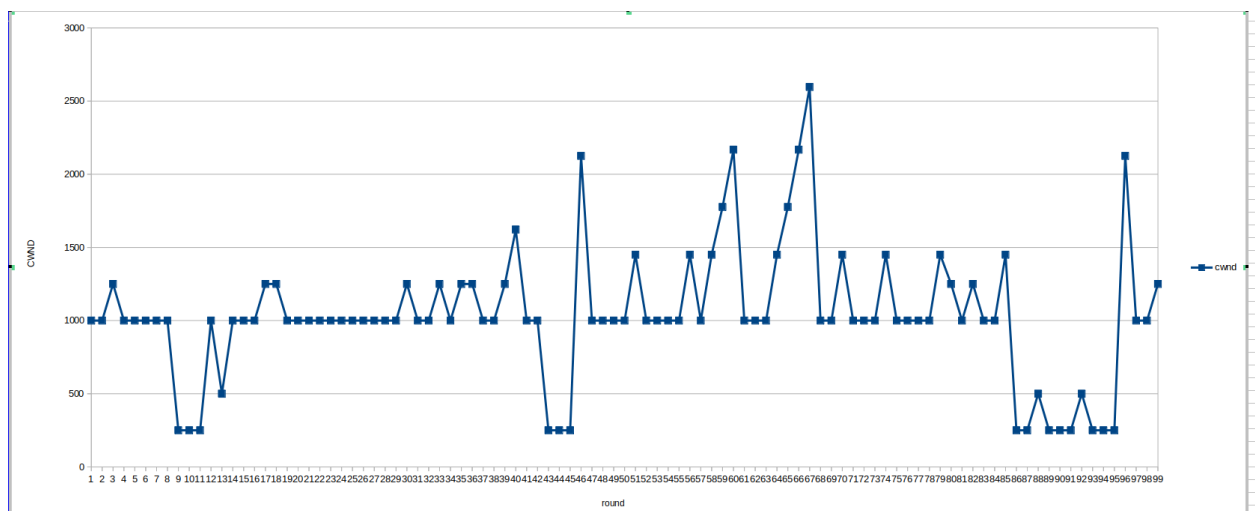
0.1



time taken = 12.47758

throughput = 406849.805811704 byte/sec
= 813.699611623 packet/sec

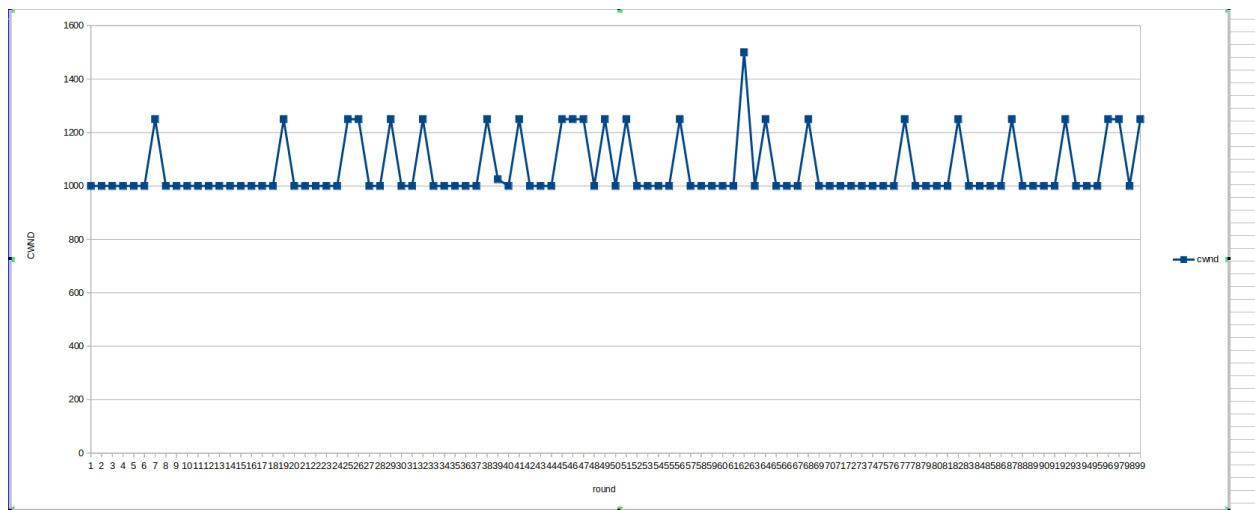
0.3



time taken = 11.238258

throughput = 451716.004384309 byte/sec
= 903.432008769 packet/sec

0.01

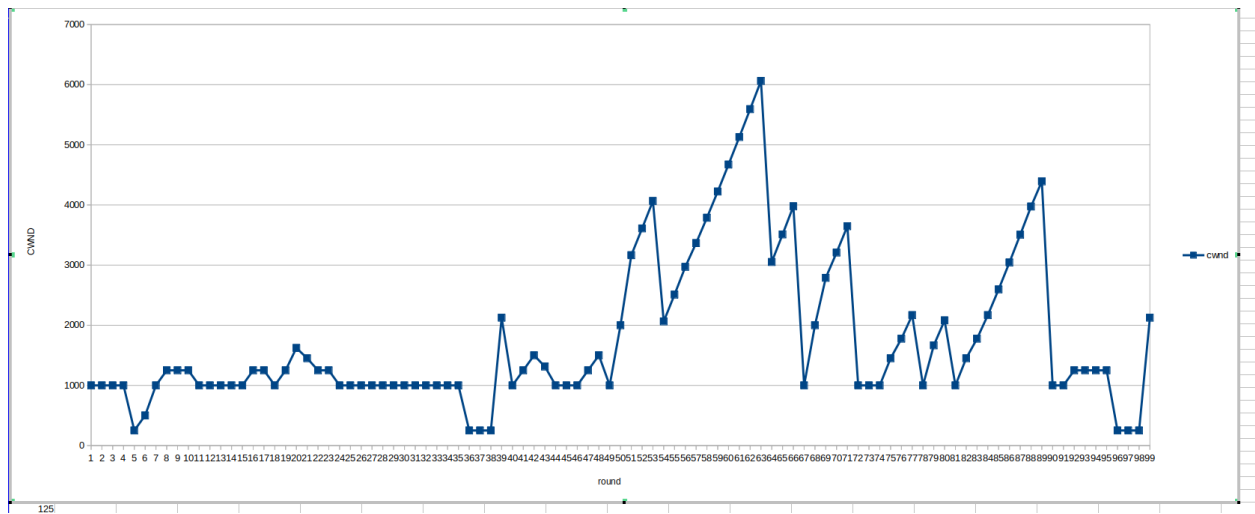


time taken = 0.460081

throughput = 11033928.808188123 byte/sec

= 22067.857616376 packet/sec

0.05



time taken = 12.34854

throughput = 411101.312381869 byte/sec

= 822.202624764 packet/sec

