# SYSC3303 Project Team 3 Iteration 3

## Table of Contents

## 0. Files

- README.pdf – Contains list of files, breakdown of responsibilities, timing diagrams, UML class diagram, test cases.
- test.txt – A 896-byte text file for running tests.
- src/Client.java – The Client.
- src/Server.java, SocketListener.java, ServerRequestThread.java – The Server classes.
- src/ErrorSimulator.java – The Error Simulator.
- src/TFTPHost.java – Interface containing all major constants.

HOW TO RUN:

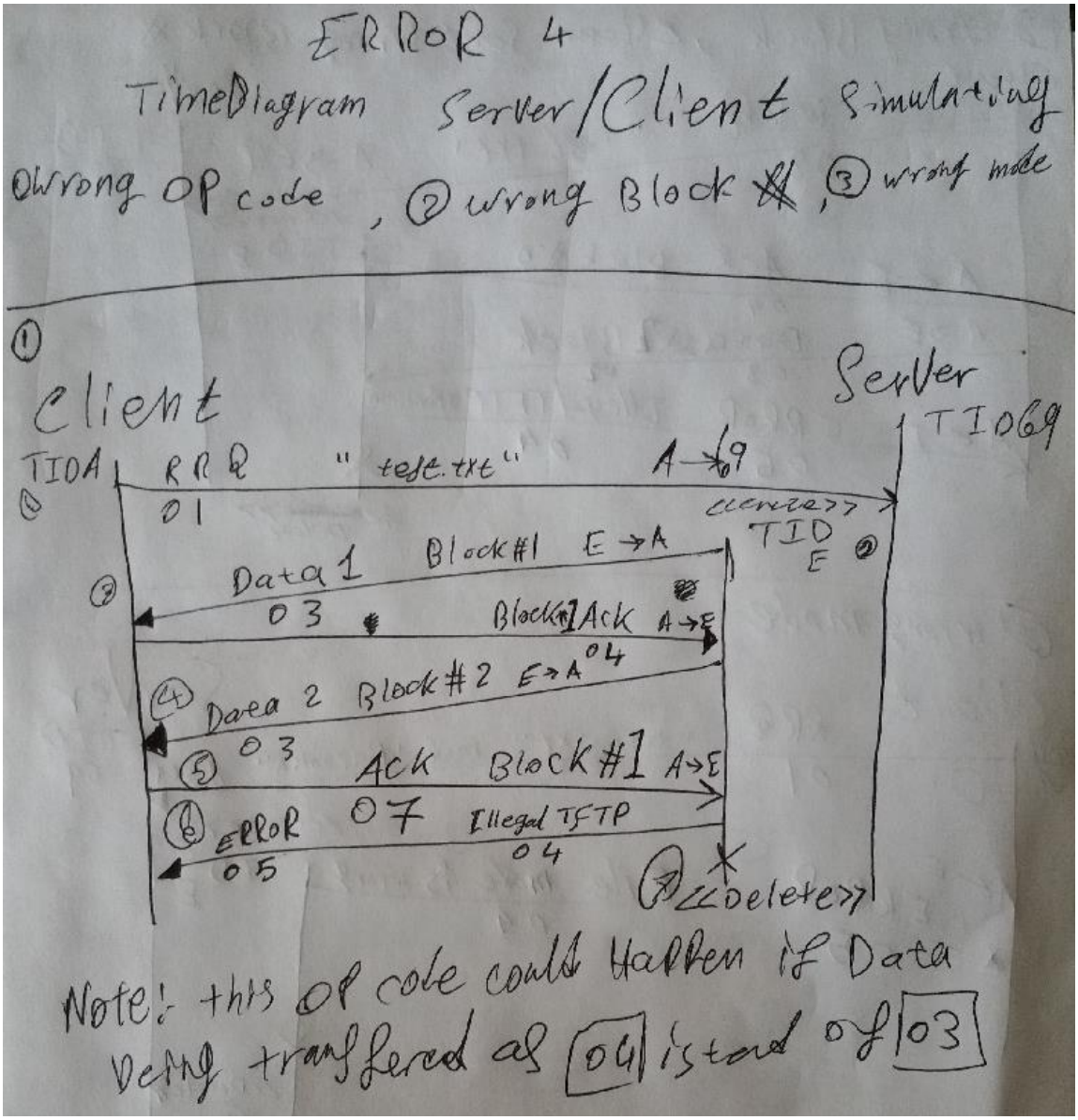
1. In Eclipse, create a new Java project and import all files in the src folder.
2. Go to C:/Users/yourname and create a new folder called "TeamWinners". Inside TeamWinners, create two new folders: "Server files" and "Client files". So you now have two new folders, one with path "C:/Users/yourname/TeamWinners/Server files" and one with path "C:/Users/yourname/TeamWinners/Client files".
3. Run Server.java. When prompted for your name, enter your CMAIL USERNAME. (E.g. when Jessica Morris runs this in the lab, she enters "jessicamorris".)
4. Run ErrorSimulator.java. For the server address, enter 127.0.0.1
5. Run Client.java. When prompted for your name, enter your CMAIL USERNAME.
6. Run any test cases in section 3.

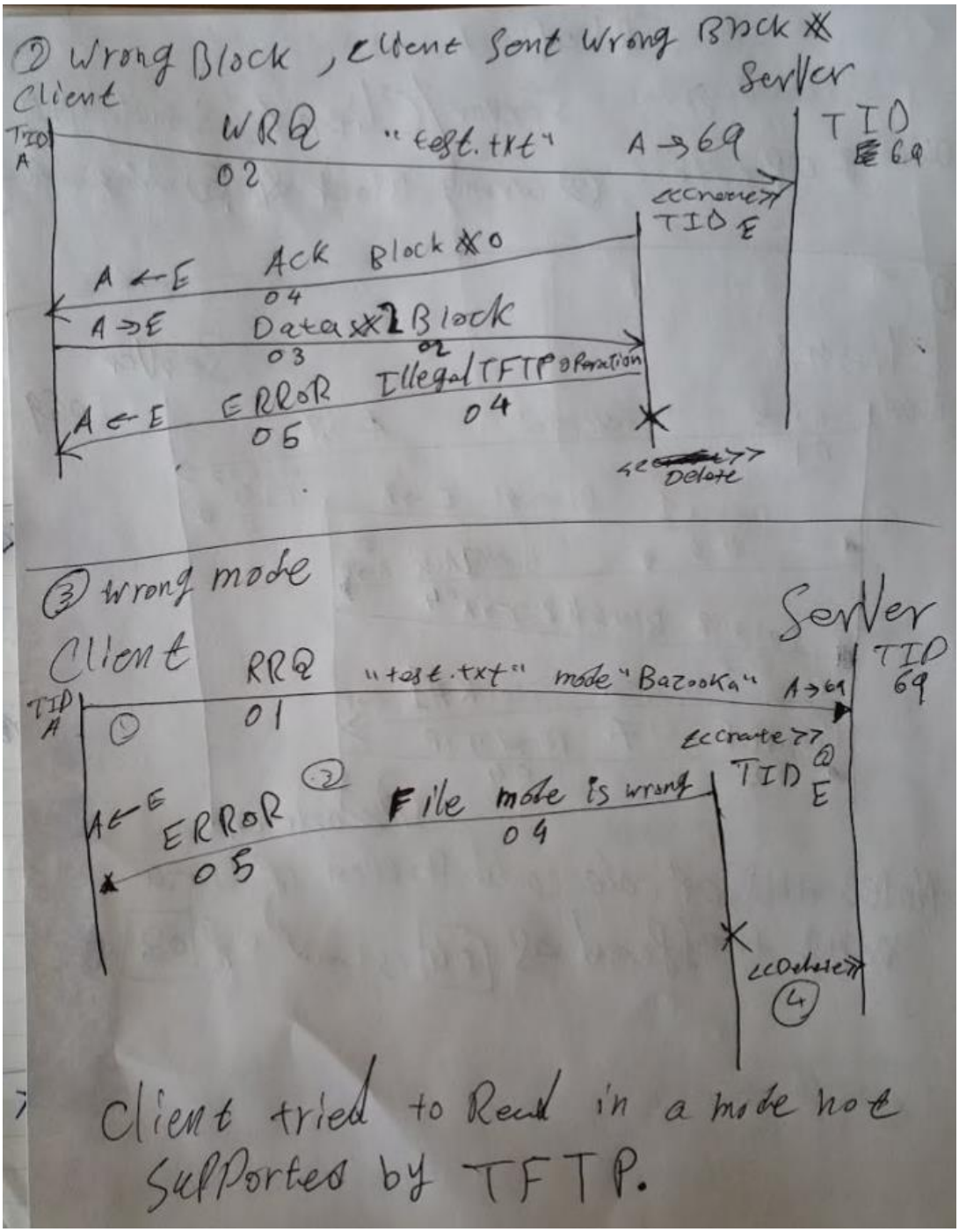
## 1. Breakdown of Responsibilities

| **Iteration 1** | |
|---|---|
| **Name** | **Responsibilities** |
| Mujtaba Alhabib | Client and Error Simulator portion of UCMs |
| Koen Bouwmans | Wrote Client.java |
| Andrew Dichabeng | Drew UML class diagram |
| Ahmad Holpa | ErrorSimulator.java |
| Tamer Kakish | Quality assurance |
| Jessica Morris | Server portion of UCMs, and wrote Server.java, ServerRequestThread.java, SocketListener.java |
| **Iteration 2** | |
| **Name** | **Responsibilities** |
| Mujtaba Alhabib | Coded and did timing diagram for Error 1 |
| Koen Bouwmans | Coded and did timing diagram for Error 3 |
| Ahmad Holpa | Coded and did timing diagram for Error 2 |
| Tamer Kakish | Coded and did timing diagram for Error 6 |
| Jessica Morris | Updated ErrorSimulator.java, tested all errors, wrote up document |
| **Iteration 3** | |
| **Name** | **Responsibilities** |
| Mujtaba Alhabib | Updated some of ErrorSimulator (handling WRQ), devised test cases |
| Koen Bouwmans | Coded and did timing diagram for Error 5 |
| Ahmad Holpa | Coded and did timing diagrams for Error 4 with Tamer |
| Tamer Kakish | Coded and did timing diagrams for Error 4 with Ahmad |
| Jessica Morris | Updated rest of ErrorSimulator (handling user input, handling RRQ), put everything together, devised test cases, wrote up document |

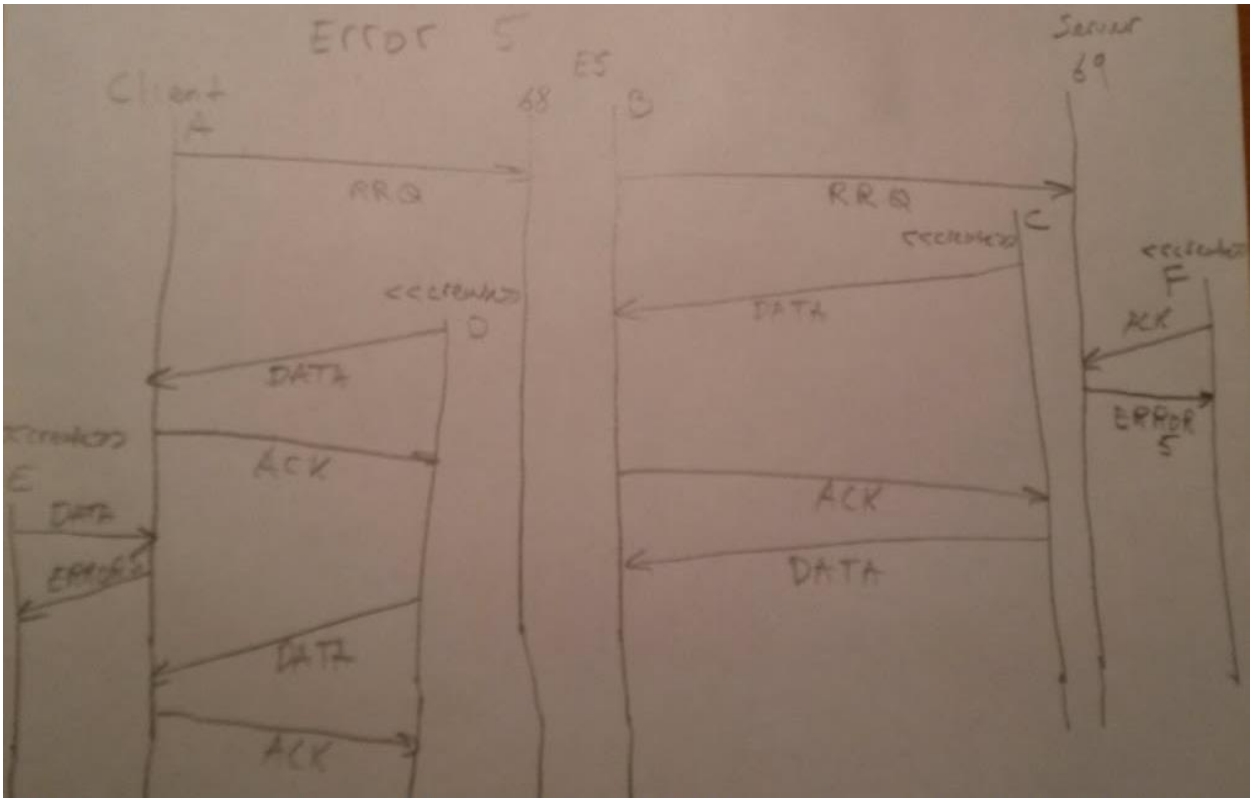
## 2. Timing Diagrams

### Error 04: Invalid TFTP Operation – Tamer and Ahmad

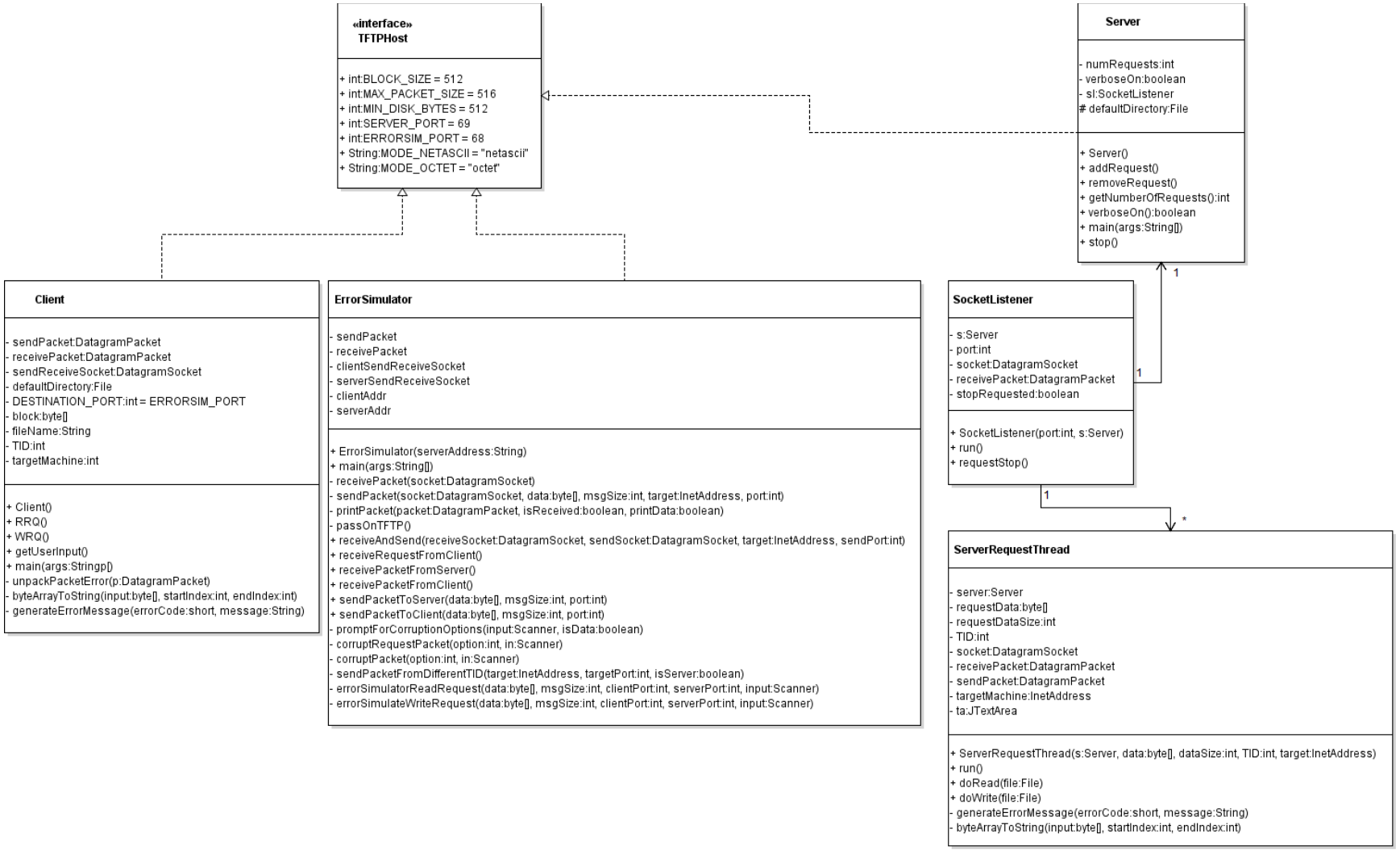



② Wrong Block, client sent Wrong Block #
Client
Server
TID A
WRQ
02
"test.txt"
A → 69
TID E
69
create
TID E
A ← E
ACK
04
Block # 0
A → E
DATA
03
# 2 Block
or
Illegal TFTP operation
04
A ← E
ERROR
05
Delete
③ wrong mode
Client
Server
TID A
RRQ
01
"test.txt" mode "Bazooka"
A → 69
TID 69
create
TID E
A ← E
ERROR
05
File mode is wrong
04
Delete
④
Client tried to Read in a mode not Supported by TFTP.

## Error 05: Invalid TID – Koen



# 3. UML Class Diagram



## 4. Breakdown of Test Cases – Jessica

Note: It is best to restart the Client, Server and ErrorSimulator after each test case to avoid issues with error packets lingering in sockets.

### General Request Tests

Setup: Each test cases says "read or write", so pick either read or write for all test cases. If you decide to go with read requests for each test case, place "test.txt" in the Server files folder. Otherwise, place "test.txt" in the Client files folder for write requests.

- **ERROR 4 Bad request opcode:**
  1. Set up a read or write request from the Client, filename "test.txt".
  2. When the ErrorSimulator receives the request, type "y" to simulate errors, then enter "1" for change opcode.
  3. Change the opcode to something Invalid, like 64 64.
  4. Server should send back error 4, for invalid opcode.
- **ERROR 4 Bad request format:**
  1. Set up a read or write request from the Client, filename "test.txt".
  2. When the ErrorSimulator receives the request, type "y" to simulate errors, then enter "2" to change packet format.
  3. The simulator will remove the 0 between filename and mode.
  4. Server should send back error 4, for invalid mode (because it can't find the mode if there isn't a 0 between the filename and mode).
- **ERROR 4 Invalid mode:**
  1. Set up a read or write request from the Client, filename "test.txt".
  2. When the ErrorSimulator receives the request, type "y" to simulate errors, then enter "3" and the simulator will switch the mode from "octet" to "BANANAS".
  3. Server should send back error 4, for invalid mode.

### Write Request Tests

Setup: Place "test.txt" in the folder "C:/Users/yourname/TeamWinners/Client files". If test.txt also exists in the Server files folder, delete it.

- **ERROR 4 Bad opcode during transfer:**
  1. Set up a write request from the Client, filename "test.txt".
  2. When the Error Simulator receives the request, enter "y" to simulate errors, then "4" to leave the request packet unchanged.
  3. When the Error Simulator receives ACK0 from the Server, enter "1" to change the opcode. Enter an invalid opcode, such as 64 64.
  4. The transfer should terminate on the Client's end and the Server's end.
  5. Repeat Steps 1-2. On Step 3, do NOT corrupt ACK0 from the Server. Enter "1" to change the opcode when the Client sends DATA1.
  6. The transfer should terminate on the Client's end and the Server's end.
- **ERROR 4 Bad block number:**
  1. Set up a write request from the Client, filename "test.txt".
  2. When the Error Simulator receives the request, enter "y" to simulate errors, then "4" to leave the request packet unchanged.
  3. When the Error Simulator receives ACK0 from the Server, enter "2" to change the block number. Enter an incorrect block number, such as 64 64.
  4. The transfer should terminate on the Client's end and the Server's end.

5. Repeat Steps 1-2. On Step 3, do NOT corrupt ACK0 from the Server. Enter "2" to change the block number when the Client sends DATA1.
6. The transfer should terminate on the Client's end and the Server's end.

- **ERROR 4 Bad packet size:**
  1. Set up a write request from the Client, filename "test.txt".
  2. When the Error Simulator receives the request, enter "y" to simulate errors, then "4" to leave the request packet unchanged.
  3. When the Error Simulator receives ACK0 from the Server, enter "3" to change the packet size. This will add an additional byte to ACK0.
  4. The transfer should terminate on the Client's end and the Server's end.
  5. Repeat Steps 1-2. On Step 3, do NOT corrupt ACK0 from the Server. Enter "3" to change the packet size when the Client sends DATA1. (n.b. Changing the packet size is only valid for DATA packets when the DATA is 512 bytes long.)
  6. The transfer should terminate on the Client's end and the Server's end.
- **ERROR 5 Invalid TID:**
  1. Set up a write request from the Client, filename "test.txt".
  2. When the Error Simulator receives the request, enter "y" to simulate errors, then "4" to leave the request packet unchanged.
  3. When the Server sends ACK0, in the ErrorSimulator, enter "5" to leave ACK0 unchanged.
  4. When the Client sends DATA1, in the Error Simulator, enter "4" to send DATA1 from the new TID. The packet will also be sent using the correct TID.
  5. In the Server popup for the request, it should say "Error 5 sent to IP address:port", followed by "DATA 1 received". In the ErrorSimulator console, it should say "Simulator: Received error 05 from Server when using different port."
  6. So now that the Server response has been verified, let's try the Client response. Back in the ErrorSimulator console, enter "4" to send ACK1 from a new TID.
  7. In the Client console, it should say "sent error message 5 to IP address:port", followed by "received ACK1". In the ErrorSimulator console, it should say "Simulator: Received error 05 from Client when using different port."
  8. Finish the request by entering "5" in the ErrorSimulator for each ACK/DATA packet until done.

## Read Request Tests

Setup: Place "test.txt" in "C:/Users/yourname/TeamWinners/Server files". If test.txt also exists in the Client files folder, delete it.

Repeat all test cases in Write Request tests, but set up read requests instead of write requests. Note that for the TID test, you MUST allow DATA1 to go from the Server to the Client normally in order for the test to be valid.