**SYSC 3303 Project Report**

**Group 2**

Ali Farah 100937214

Lava Tahir 100933449

Tosin Oni 100961759

Mohamed Dahrouj 100951843

Vanja Veselinovic 100946517

April 11 2018

Contents

[Breakdown of Responsibilities: 3](#_Toc511052336)

[Diagrams: 5](#_Toc511052337)

[Set up instructions: 13](#_Toc511052338)

# Breakdown of Responsibilities:

For Iteration 1:

|  |  |
| --- | --- |
| Team Member | Contributions |
| Ali Farah | Client Code, Helper Classes |
| Lava Tahir | Server Code, Helper Classes |
| Tosin Oni | Client Code, Helper Classes, UCM Diagrams |
| Mohamed Dahrouj | Server Code, Helper Classes |
| Vanja Veselinovic | Server Code, Helper Classes, UML Diagrams, Error Simulator |

For Iteration 2:

|  |  |
| --- | --- |
| Team Member | Contributions |
| Ali Farah | Error Code 1, 2 I/O error handlers |
| Lava Tahir | Error Code 1, 2 I/O error handlers |
| Tosin Oni | Error Code 3, 6 I/O error handlers |
| Mohamed Dahrouj | Error Code 1, 2 I/O error handlers, Timing Diagrams for all error scenarios |
| Vanja Veselinovic | Error Code 3, 6 I/O error handlers |

For Iteration 3:

|  |  |
| --- | --- |
| Team Member | Contributions |
| Ali Farah | Sorceror’s Apprentice fix, lost packet, delayed packet, timeout timing diagram |
| Lava Tahir | Sorceror’s Apprentice fix, delayed packet, duplicated packet, retransmit timing diagram |
| Tosin Oni | Test menu error simulator, space between duplicates |
| Mohamed Dahrouj | Delayed packet, duplicated packet |
| Vanja Veselinovic | Delayed packet, duplicated packet, updated UML |

For Iteration 4:

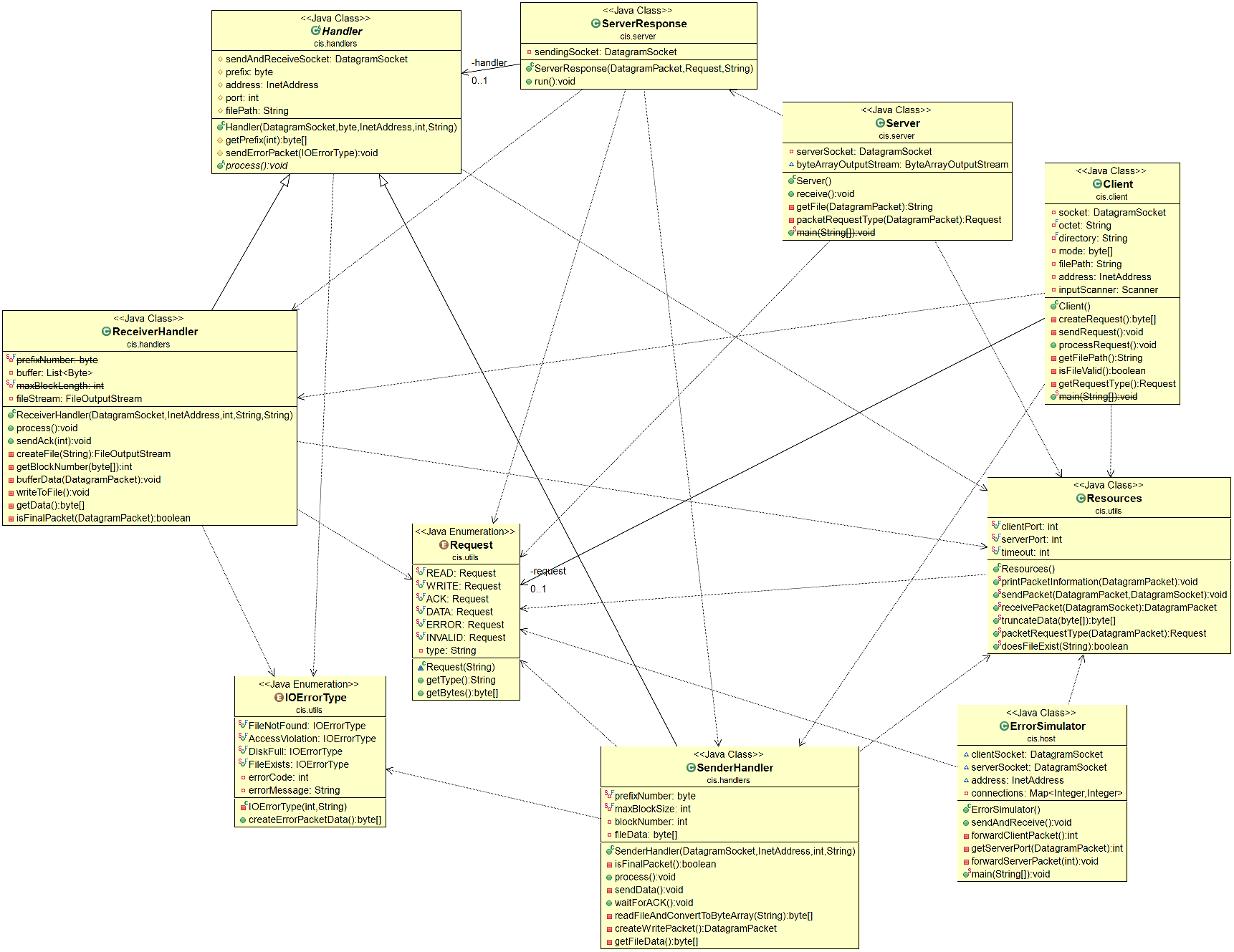
|  |  |
| --- | --- |
| Team Member | Contributions |
| Ali Farah | Error code 4, error code 5, timing diagrams error 5 |
| Lava Tahir | Error code 4, error code 5, updating UML diagrams |
| Tosin Oni | Updating test menu, error simulator |
| Mohamed Dahrouj | Error code 4, error code 5, timing diagrams error 4 |
| Vanja Veselinovic | Updating test menu, error simulator |

For Iteration 5:

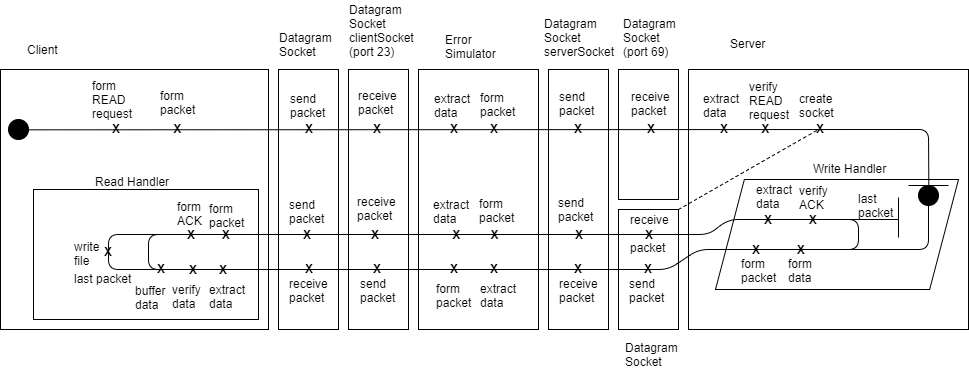
|  |  |
| --- | --- |
| Team Member | Contributions |
| Ali Farah | Updating Server Host Address |
| Lava Tahir | Change UI to update Server Host Address |
| Tosin Oni | Change UI to update Server Host Address |
| Mohamed Dahrouj | Updating Server Host Address |
| Vanja Veselinovic | Testing, Updating Server Host Address |

# Diagrams:

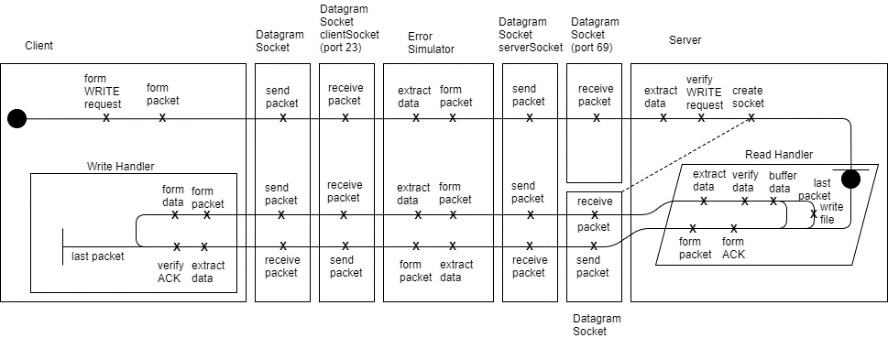
UML Class Diagram:



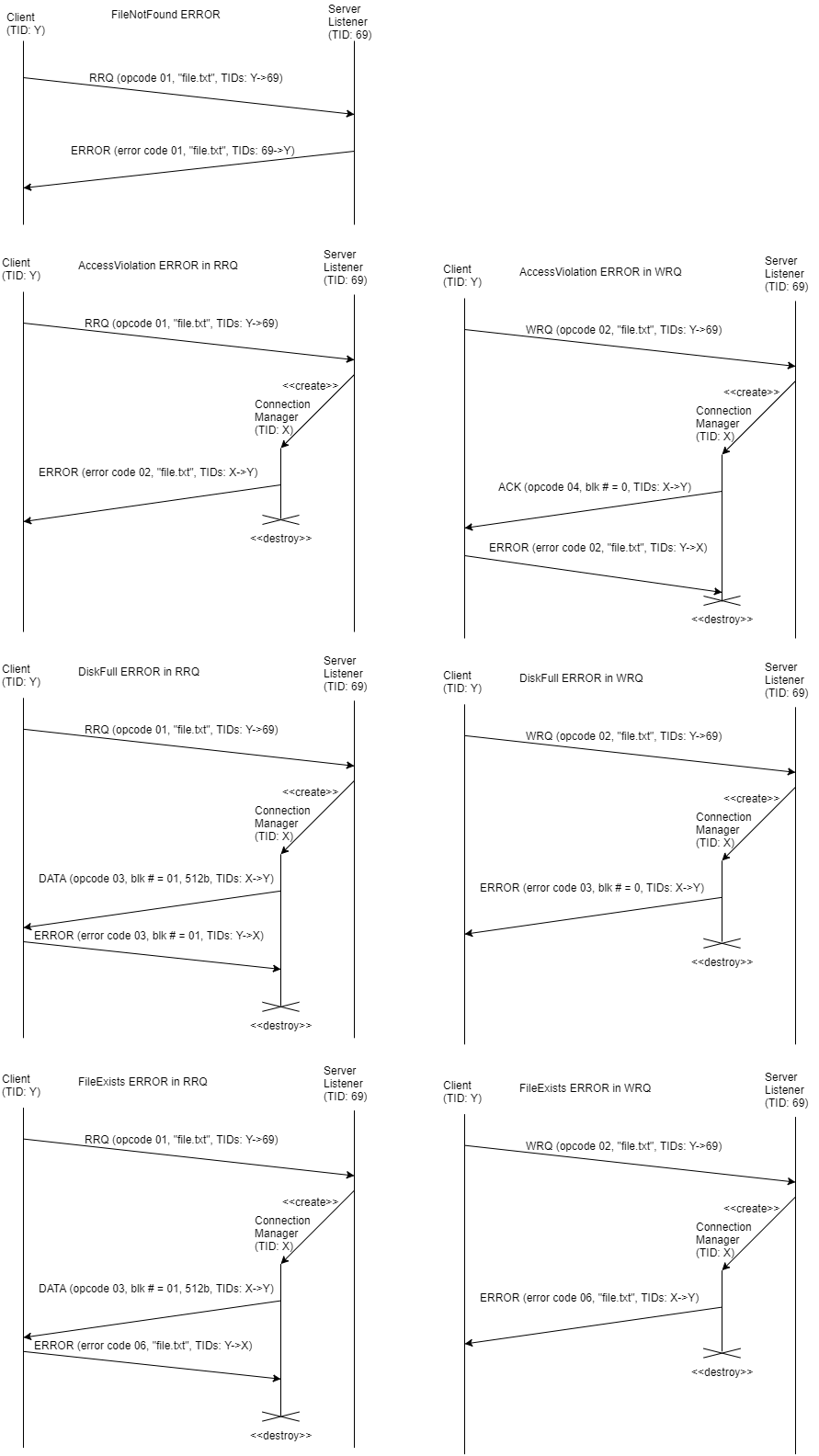
UCM Read:



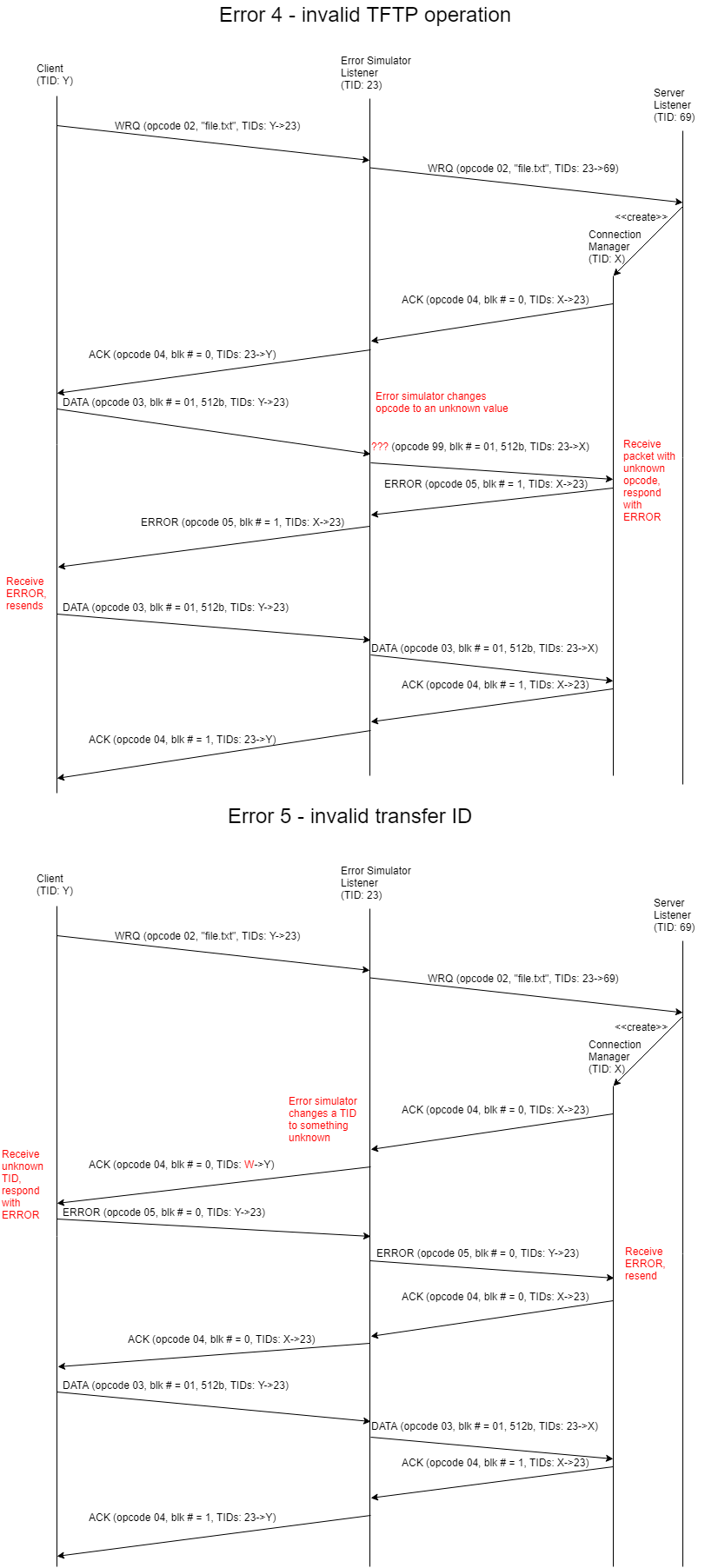
UCM Write:



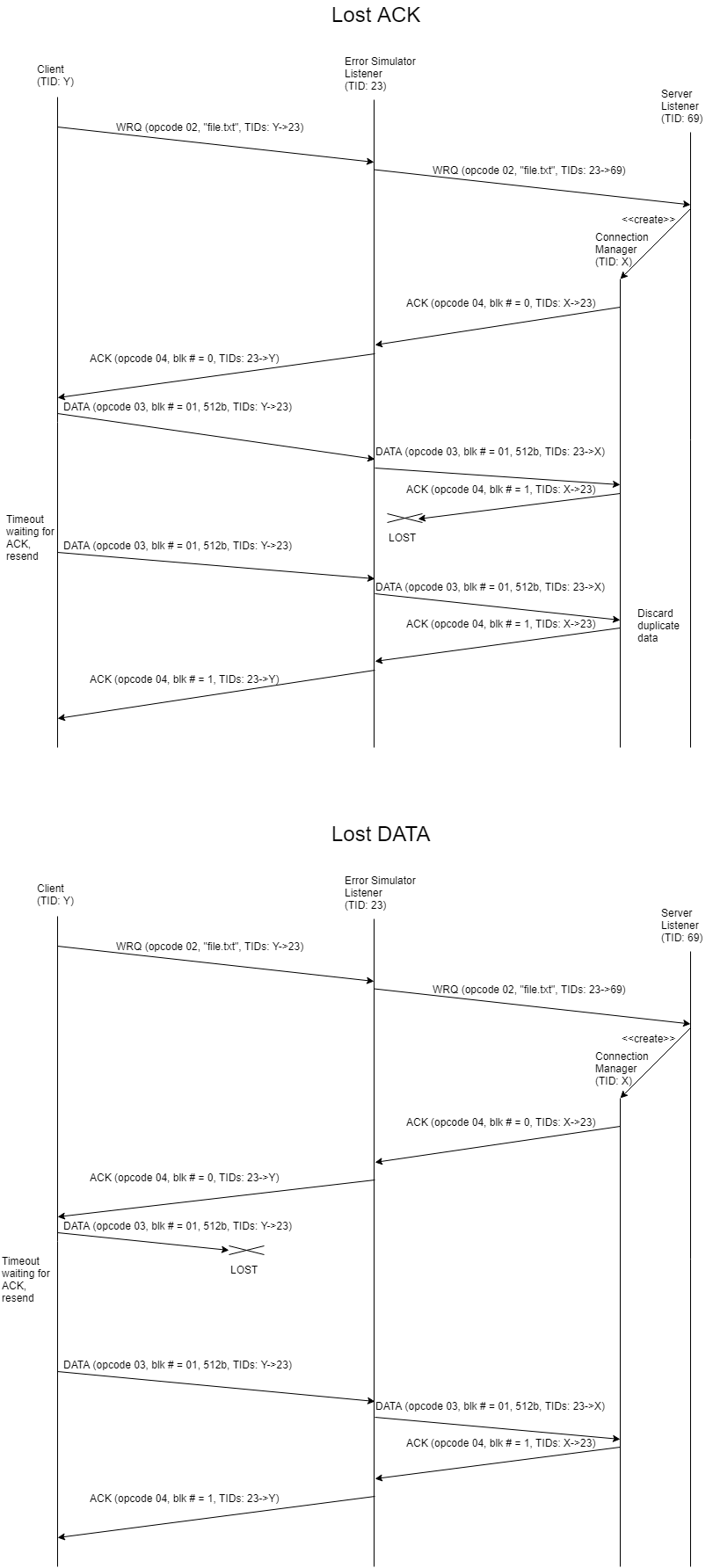
Timing Diagrams for FNF, AV, DF, FE:



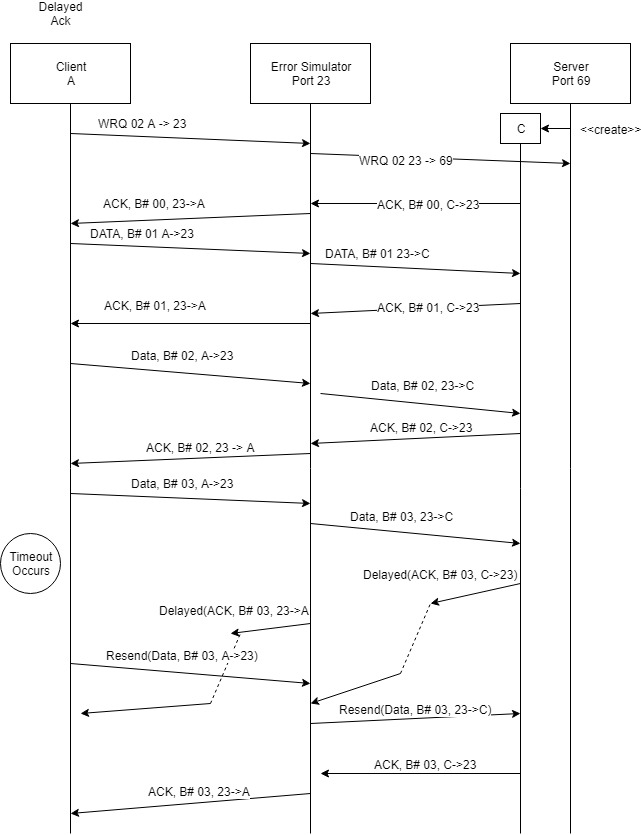
Timing Diagrams for invalid TFTP, transfer ID:



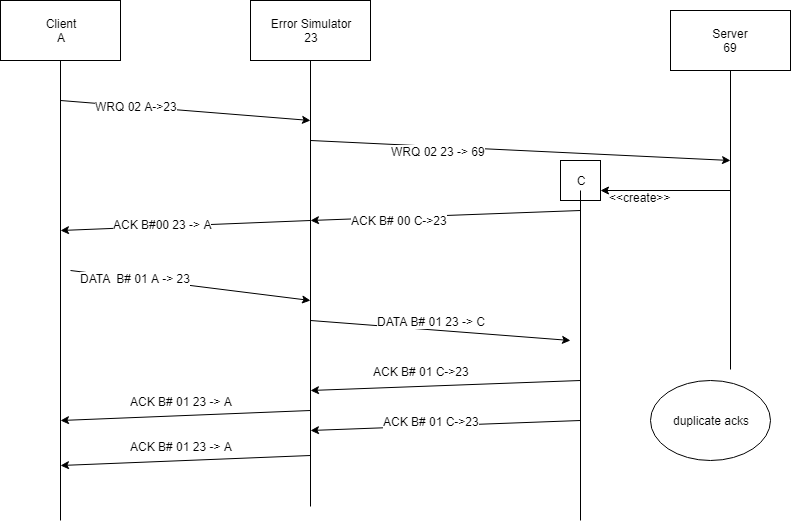
Timing Diagrams for lost ACK and Data packets:



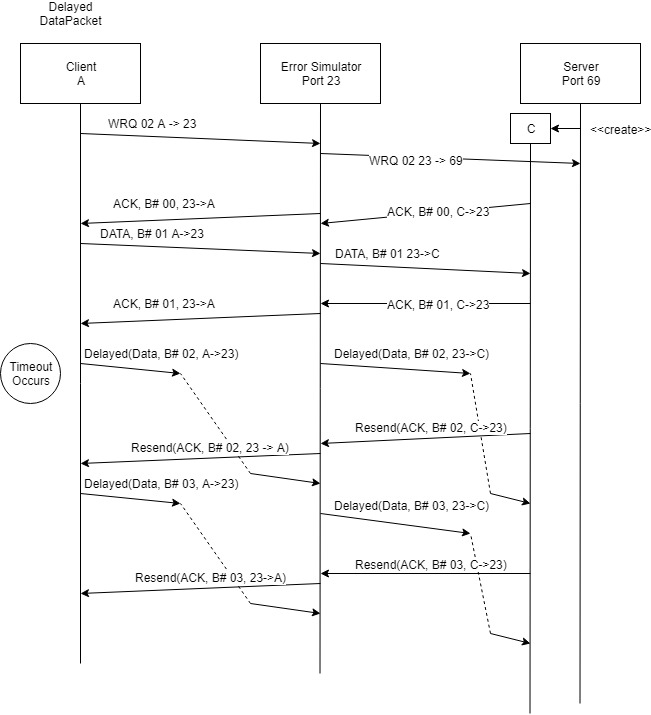
ACK Delay:



ACK Duplicate:



Packet Delay:



# Set up instructions:

1. Execute the Server.java in server folder (main)
2. Execute the ErrorSimulator.java in host folder (main)
3. Execute the Client.java in client folder (main)
4. After starting the Error Simulator you will be prompted to enter whether or not you want to delay/duplicate/ lose packet or send a illegal packet.
5. After starting the Client you will be prompted to enter the request type. Type R for a read request or W for a write request.
6. Then you will be prompted to enter the path of a file. 7a. If you enter a read request then the contents of the file that you specified will be transferred to a file of the same name in the Client folder. 7b. If you enter a write request then the contents of the file that you specified will be transferred to a file of the same name in the Server folder.
7. The client and the server response will timeout after 1s.
8. The server and the ErrorSimulator will shutdown after 300s of inactivity.

There are 5 folders within the source code.

Client:

Client.java : Represents the client class which interacts with the system

Host:

ErrorSimulator.java : Represents the intermediate host which interacts between client and server and simulates errors.

Server:

Server.java : Represents the server which sends and receives to the intermediate host.

Handlers:

Handler.java: Is the base class of the WriteHandler and ReadHandler.

WriteHandler.java: This class will send the contents of a file to the receiver.

ReadHandler.java: This class will receive data from a file and write it to a local file.

Utils:

Request.java: Is an enumerated class that represents the request types of the packets in the simulation

Resources.java : Is a common class that is used by all the Client, ErrorSimulator and Server to print, send and receive packets.

IOErrorType.java : A class that enumerates the different IO error types.