TFTP Project - Team 2000000

SYSC 3303 – A

**Members:**

Supriya Gadigone

Kshamina Ghelani

Daniel Graval

Bhavik Tailor

Tanzim Zaman

Contents

[Breakdown of Responsibilities 4](#_Toc468755202)

[UCM Diagrams 6](#_Toc468755203)

[UML Class Diagram 8](#_Toc468755204)

[Timing Diagrams 9](#_Toc468755205)

[Error 1 - File Not Found 9](#_Toc468755206)

[Error 2 - Access Violation 9](#_Toc468755207)

[Error 3 - Disk Full, Case 1 10](#_Toc468755208)

[Error 3 - Disk Full, Case 2 10](#_Toc468755209)

[Error 6 - File Already Exists, RRQ 11](#_Toc468755210)

[Error 6 - File Already Exists, WRQ 11](#_Toc468755211)

[Lost request 12](#_Toc468755212)

[RRQ, Lose DATA Packet 12](#_Toc468755213)

[RRQ, Lose ACK Packet 13](#_Toc468755214)

[WRQ, Lose ACK Packet 13](#_Toc468755215)

[WRQ, Lose DATA Packet 14](#_Toc468755216)

[RRQ, Duplicate ACK Packet 14](#_Toc468755217)

[RRQ, Duplicate DATA Packet 15](#_Toc468755218)

[WRQ, Duplicate DATA Packet 16](#_Toc468755219)

[WRQ, Duplicate ACK 16](#_Toc468755220)

[Delay Request 17](#_Toc468755221)

[RRQ, Delay DATA 17](#_Toc468755222)

[RRQ, Delay ACK 18](#_Toc468755223)

[WRQ, Delay DATA 19](#_Toc468755224)

[WRQ, Delay ACK 20](#_Toc468755225)

[Error 5 - Invalid  TID, Sent to Client 20](#_Toc468755226)

[Error 5 - Invalid  TID, Sent to Server 21](#_Toc468755227)

[Error 4, Invalid Request Packet 21](#_Toc468755228)

[Error 4, Invalid Opcode, to Server 22](#_Toc468755229)

[Error 4, Invalid Opcode, to Client 22](#_Toc468755230)

[Error 4, Invalid Packet Size, to Server 23](#_Toc468755231)

[Error 4, Invalid Packet Size, to Client 23](#_Toc468755232)

[Error 4, Invalid Block Number, to Server 24](#_Toc468755233)

[Error 4, Invalid Block Number, to Client 24](#_Toc468755234)

[Testing Instructions 25](#_Toc468755235)

[Disk Full 25](#_Toc468755236)

[Client 25](#_Toc468755237)

[Server 25](#_Toc468755238)

[File Already Exists 25](#_Toc468755239)

[Client 25](#_Toc468755240)

[Server 25](#_Toc468755243)

[File Not Found 25](#_Toc468755247)

[Client 25](#_Toc468755248)

[Server 25](#_Toc468755251)

[File Access Denied 25](#_Toc468755254)

[Illegal TFTP 26](#_Toc468755255)

[Unknown Transfer ID 26](#_Toc468755256)

[Intermediate Host – Normal, Lost, Delay, Duplicate 26](#_Toc468755259)

[Intermediate Host – Manipulating Error Packet 26](#_Toc468755260)

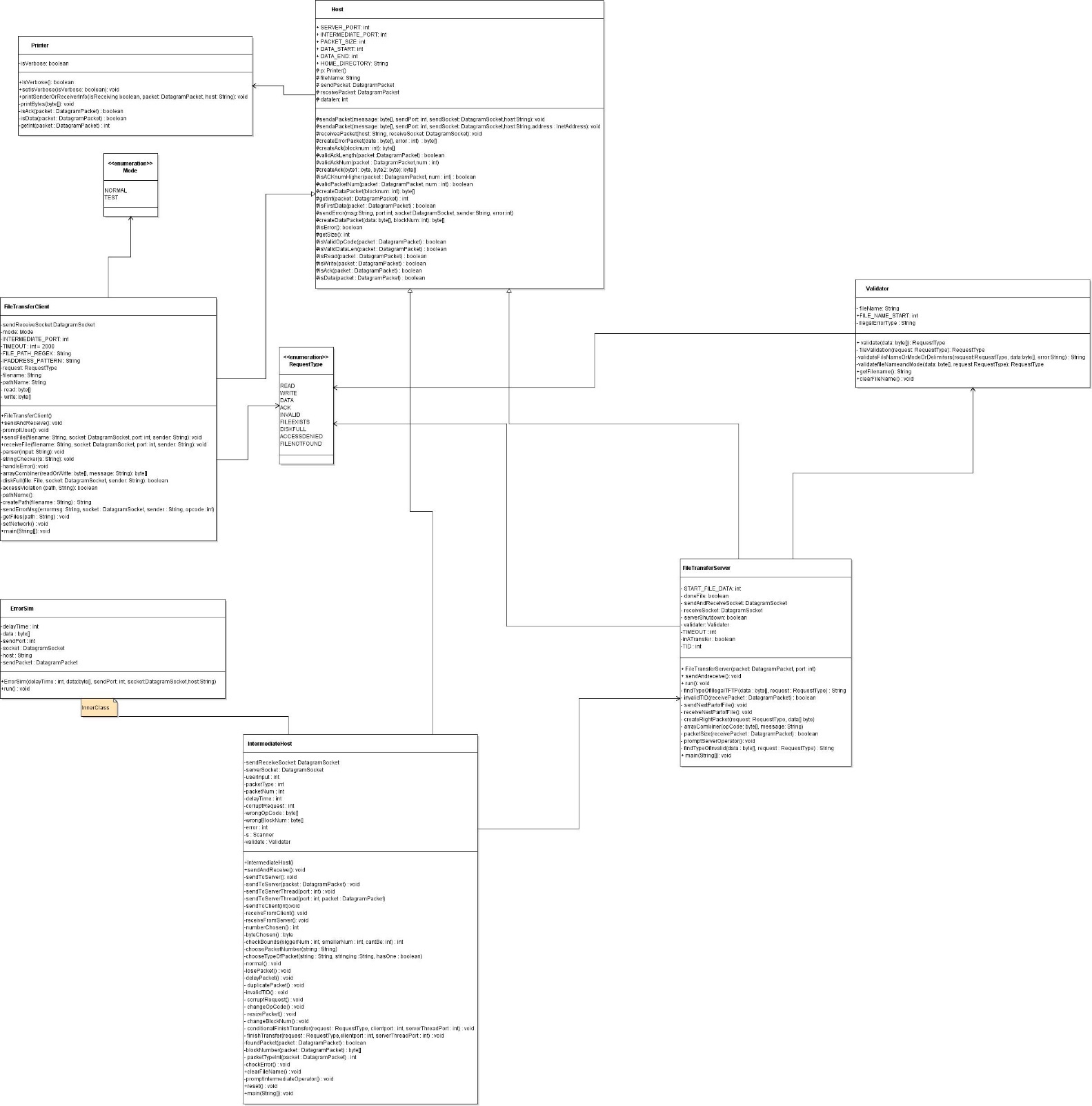
# Breakdown of Responsibilities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I# | Group Member | | | | |
| Supriya Gadigone | Kshamina Ghelani | Daniel Gravel | Bhavik Tailor | Tanzim Zaman |
| 1 | * Made JavaDocs comments for all methods * Adjusted methods within Printer class * Helped debug FileTransferClient | * UML Class Diagrams * UCM Diagrams * README.md | * FileTransferClient.java * Host.java * receiveFile * sendFile * createAck * createDataPacket * Printer * Verbose/Quiet | * FileTranserServer.java * Made method to receive file data from client via intermediate * Adjusted IntermediateHost to accommodate larger file transfers * Shutdown in client | * FileTransferServer.java * Receiving files * Sending files * Validation and what to do with requests/data/acks * Host.java * Debugged * FileTransferClient.java * Debugged |
| 2 | * Prompt for file path in client * Client timeout * UML Class Diagram * General debugging and testing (Client Side) | * FileTransferServer.java * error handling, server side * server shutdown * debugging/testing * Printer.java * Worked on fixing bug in verbose mode (Server side) * timing diagrams | * Testing different errors * FileTransferClient.java * receiveFile * sendFile * user prompt * server error handling * client error handling | * Made methods for handling errors in client class * Adjusted UCM Diagrams * Debugging client side parser * General Debugging | * Decoupled server class, moved validation to own class: Validater.java * Moved RequestType to own class * FileTransferServer.java * Debugged * Updated all aspects with regards to feedback an Iteration 2 * Host.java * Debugged, added classes |
| 3 | * Timing diagrams * Handled duplicate packets | * UML Class Diagram * Fixed block numbers from client for ack/data * IntermediateHost.java * Debugged * Wrote prompt for different network errors * Implemented losing and delaying packets | * Added ability to change directories * Added various new commands(ls, pwd) * Debugged IntermediateHost, client, and Host * Helped implement lost packets client side | * Handled lost packets on server side * Handled multiple ACKs on server | * Aided in implementation of duplicating, delaying, losing packets * Added changing op code feature |
| 4 | * Implemented server side TID and invalid tftp * Debugged server side issues | * Timing Diagrams * Added error simulations for invalid TID, corrupt request, invalid packet size, change block number * Debugged Intermediate Host and Host | * Added client side error handling for TID,invalid TFTP * Helped debug host, intermediate host | * Updated UML Diagrams * Updated ReadME * Helped debug server side issues | * Added error simulation: alter opcode * Debugged Intermediate Host * Tested Error Sim scenarios |
| 5 | * Debugged and tested on server side * Fixed illegal TFTP error on server side * Filename * Mode * Delimiter | * Fixed IT4 issues in Intermediate Host * Fixed sending and receiving packets using correct length * Debugged and tested IH and Server * Put together final hard copy report | * Fixed IT4 issues in client and Host * Debugged and tested * Implemented sending across a network * Diagrams * Fixed server shutdown |  | * Implemented quiet mode * Tested all cases * Fixed corrupting packet/opcode bugs * Added and aided error implementations (losing, delaying, etc) * Prompt fixed to also accept errors |

# UCM Diagrams

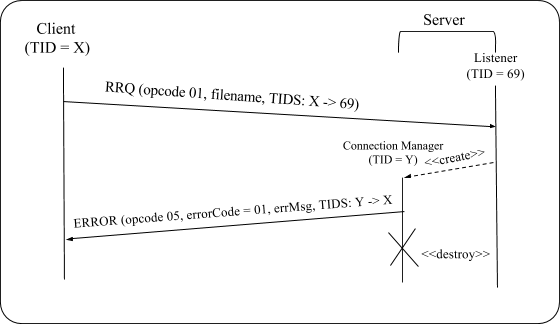


# UML Class Diagram

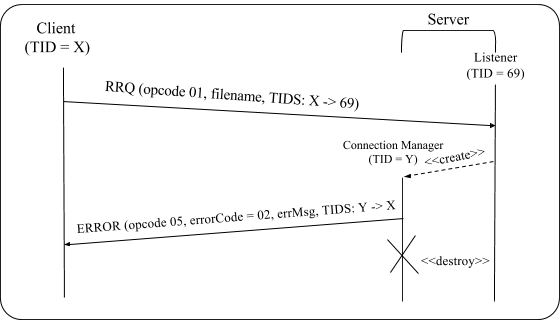


# Timing Diagrams

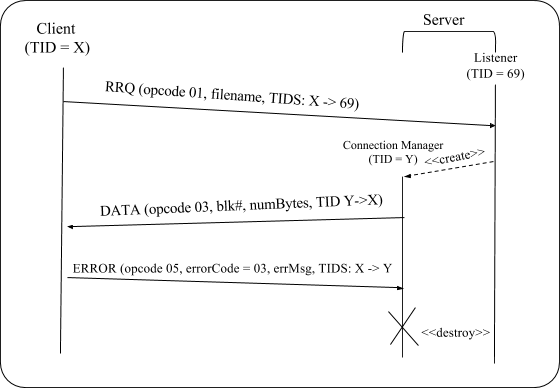
## Error 1 - File Not Found



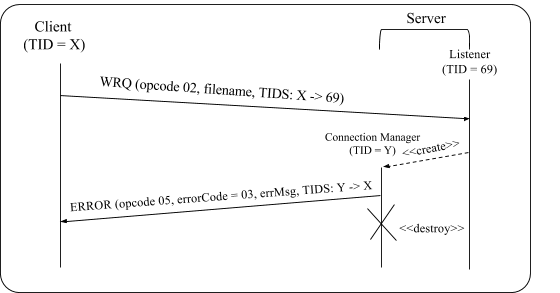
## Error 2 - Access Violation



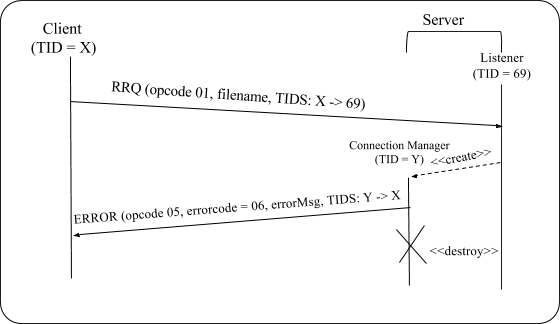
## Error 3 - Disk Full, Case 1



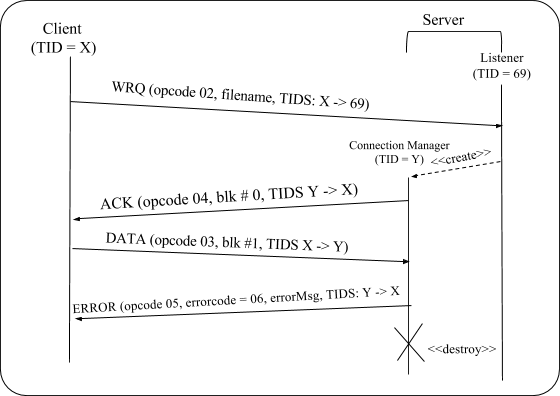
## Error 3 - Disk Full, Case 2



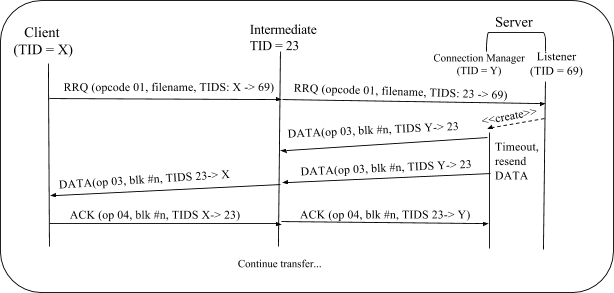
## Error 6 - File Already Exists, RRQ



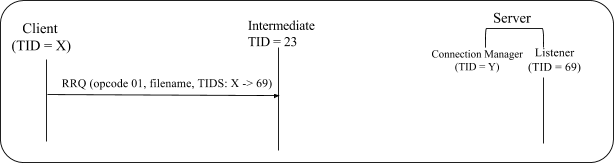
## Error 6 - File Already Exists, WRQ



## Lost request

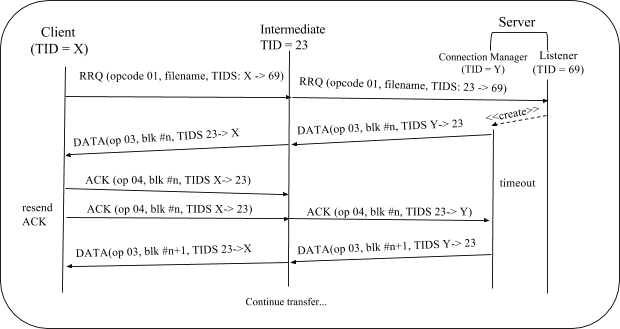


## RRQ, Lose DATA Packet

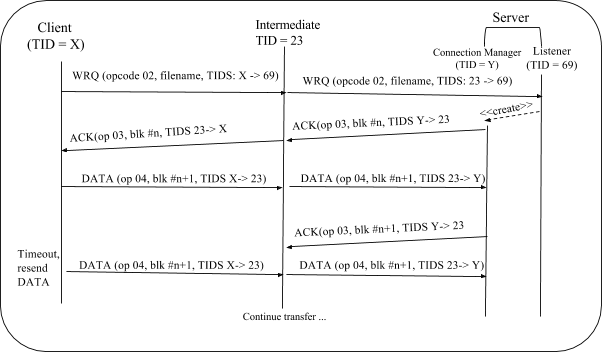


\*\* client will timeout and re prompt user for a new request (same for WRQ)

## RRQ, Lose ACK Packet



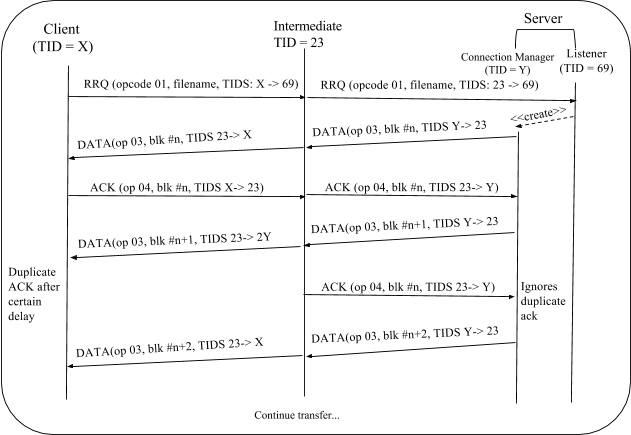
## WRQ, Lose ACK Packet



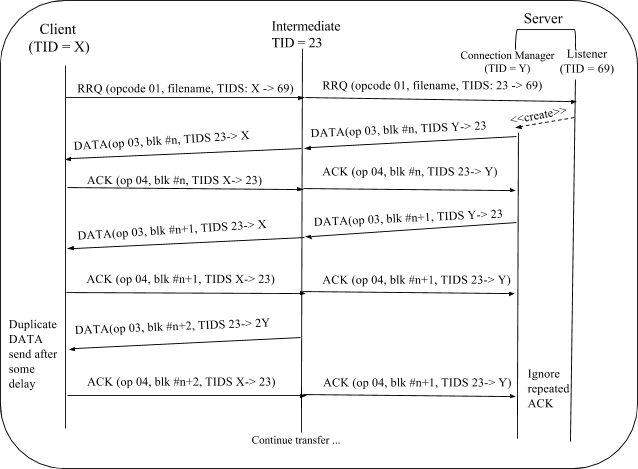
## WRQ, Lose DATA Packet



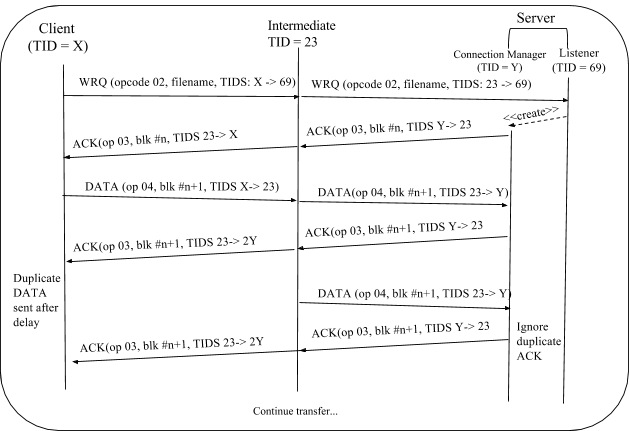
## RRQ, Duplicate ACK Packet



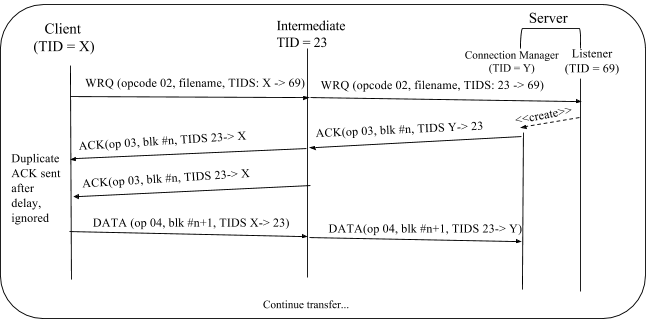
## RRQ, Duplicate DATA Packet



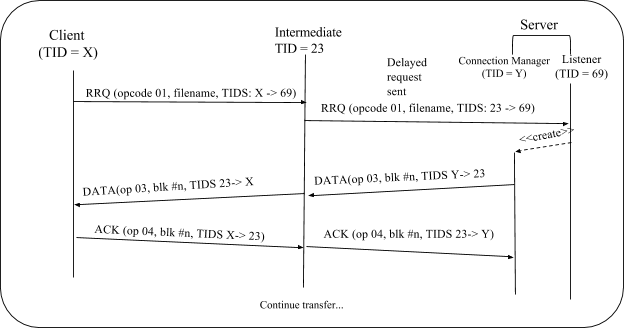
## WRQ, Duplicate DATA Packet



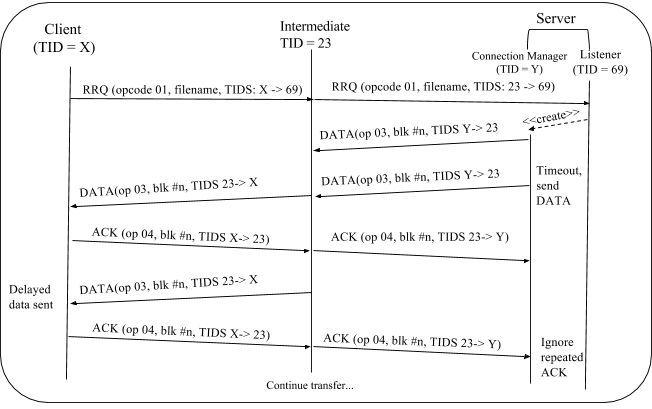
## WRQ, Duplicate ACK

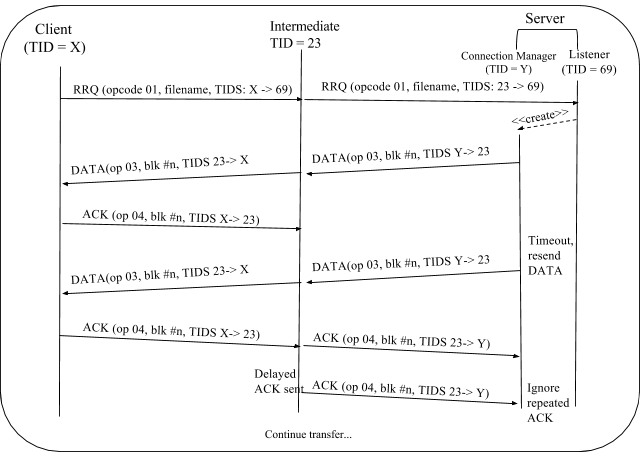


## Delay Request



\*\* if client times out, client will reprompt the user for a new request (same for WRQ)

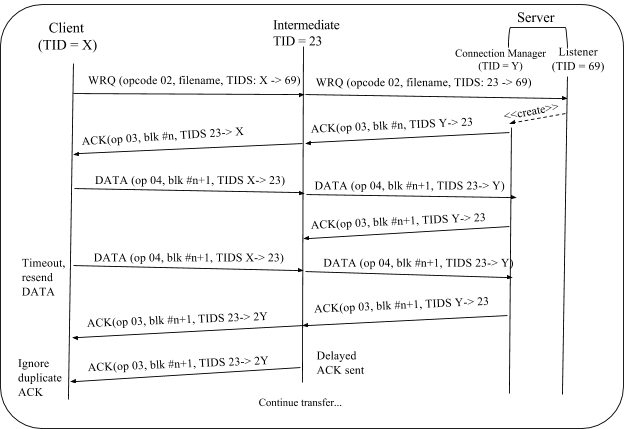
RRQ, Delay DATA  


RRQ, Delay ACK  
  


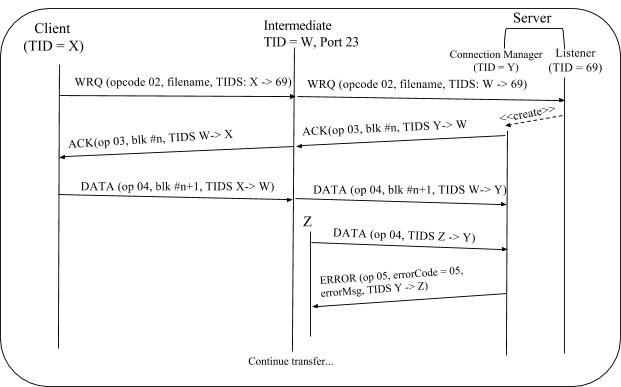
## WRQ, Delay DATA



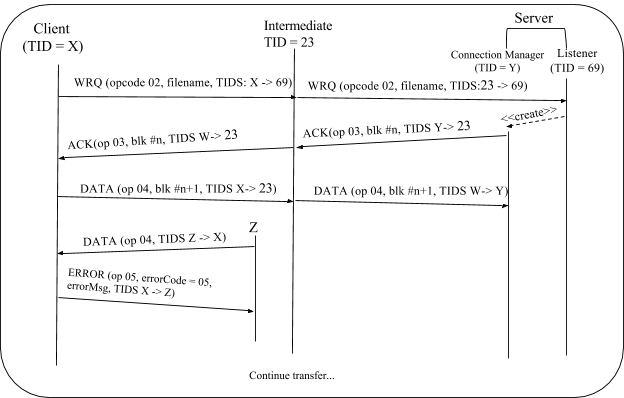
## WRQ, Delay ACK



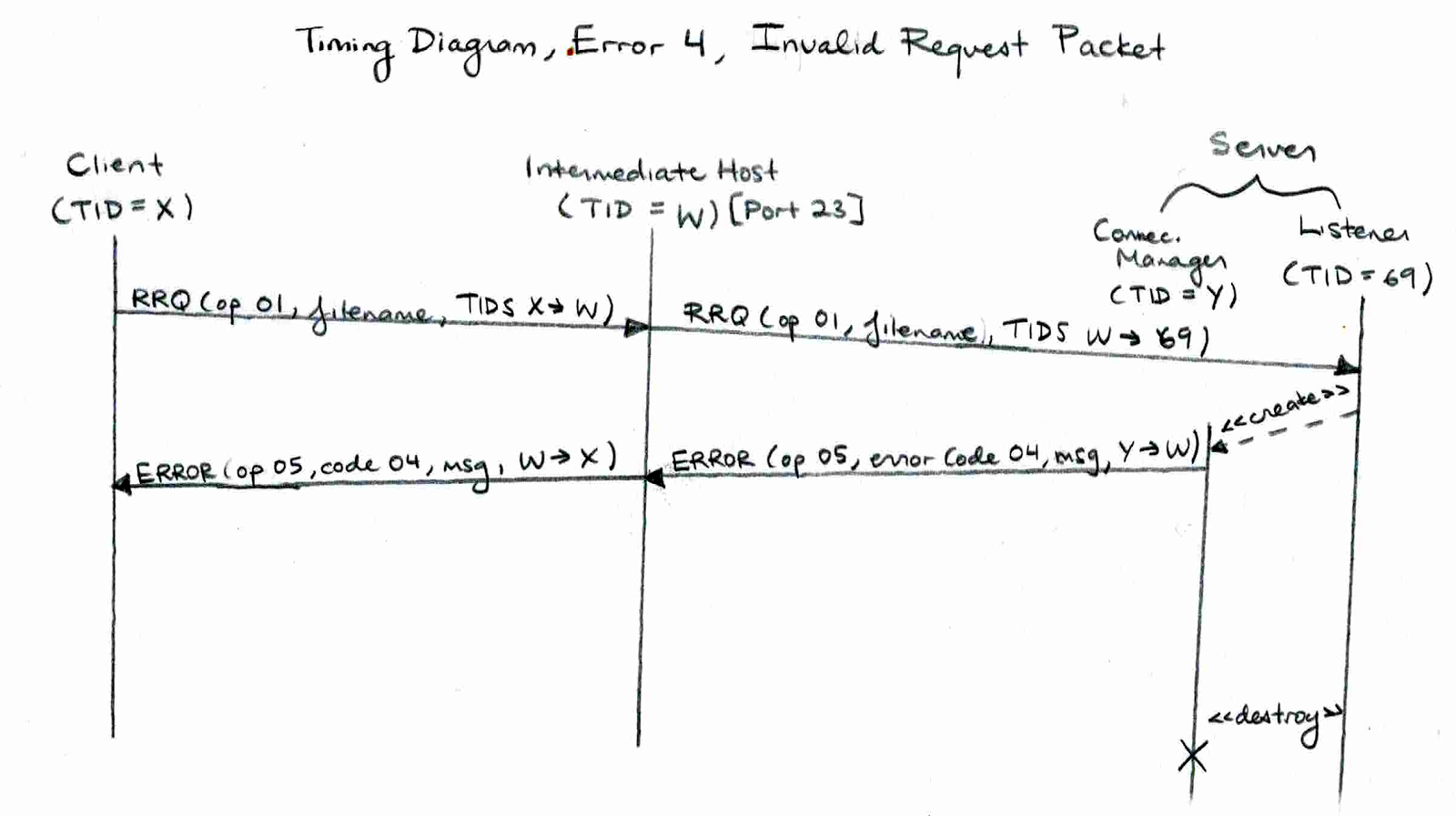
## Error 5 - Invalid  TID, Sent to Client



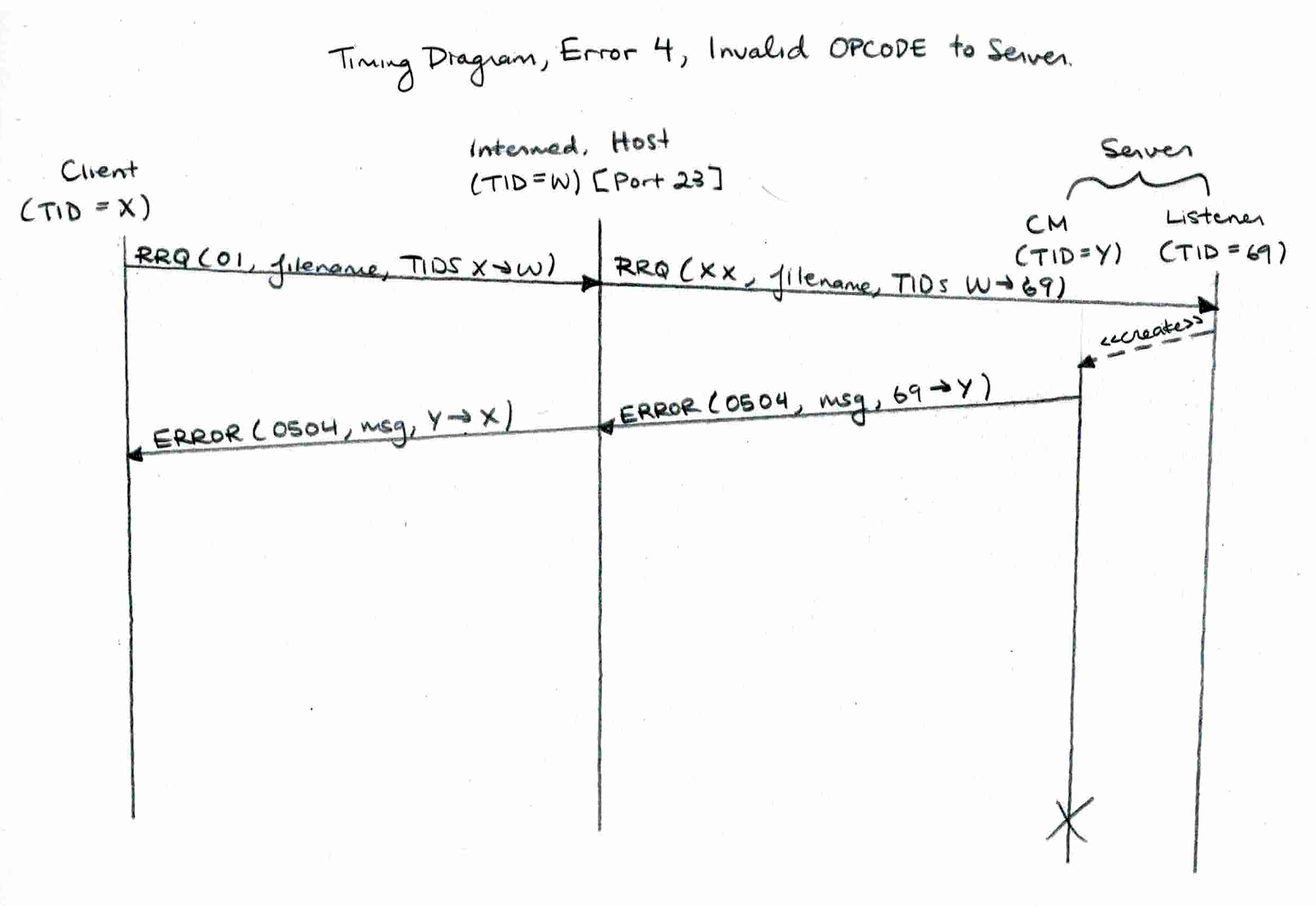
## Error 5 - Invalid  TID, Sent to Server



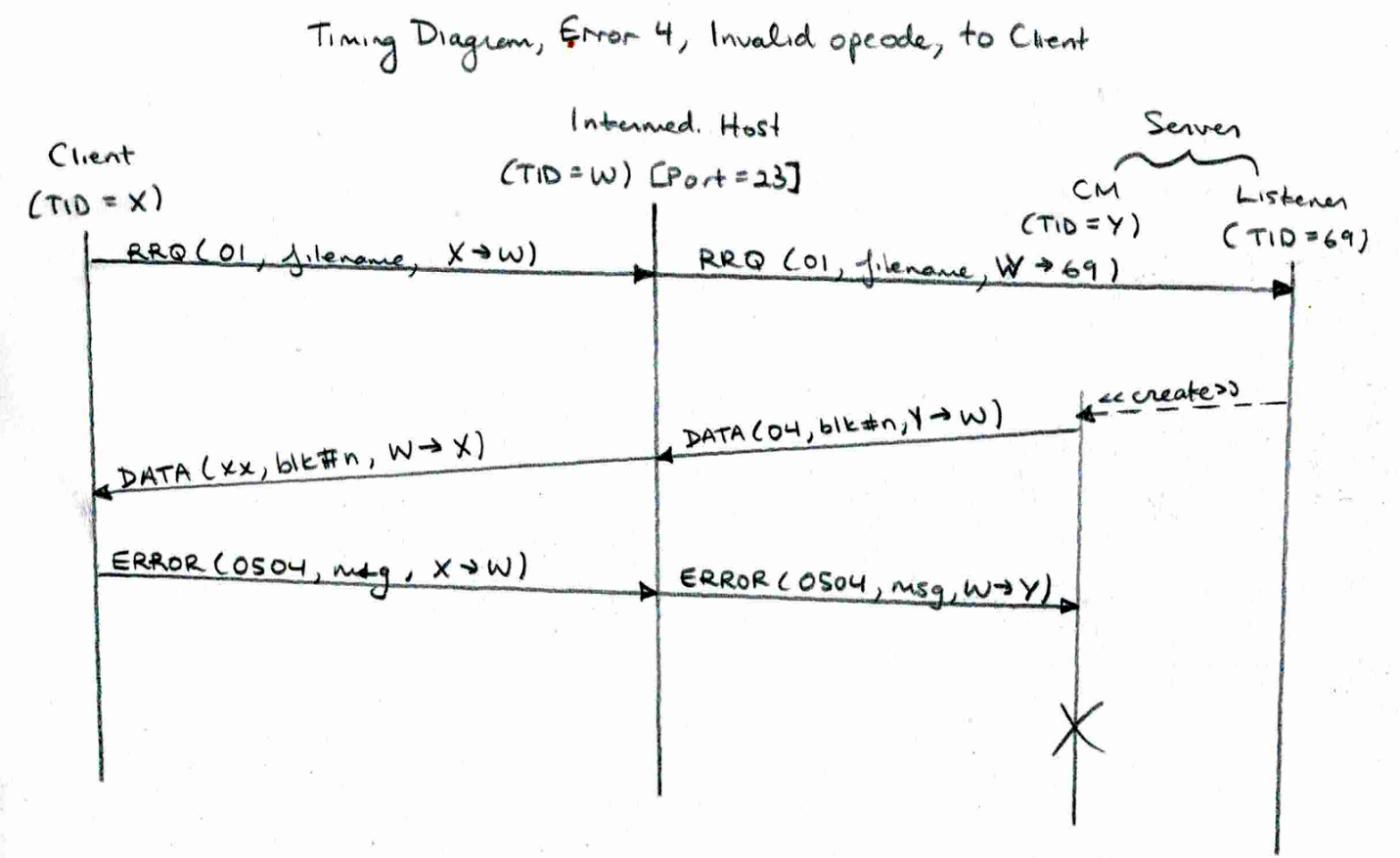
## Error 4, Invalid Request Packet



## Error 4, Invalid Opcode, to Server

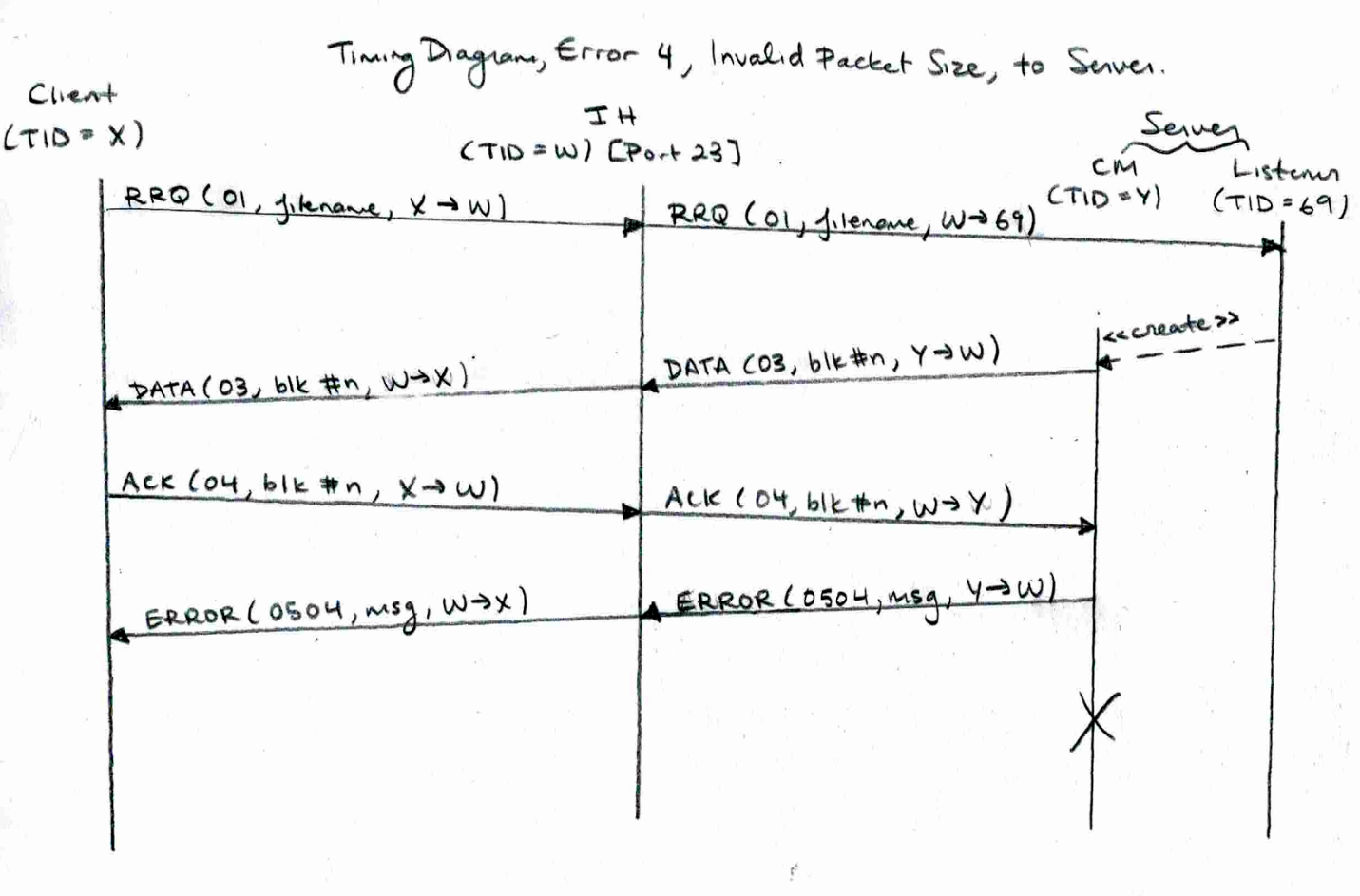


## Error 4, Invalid Opcode, to Client



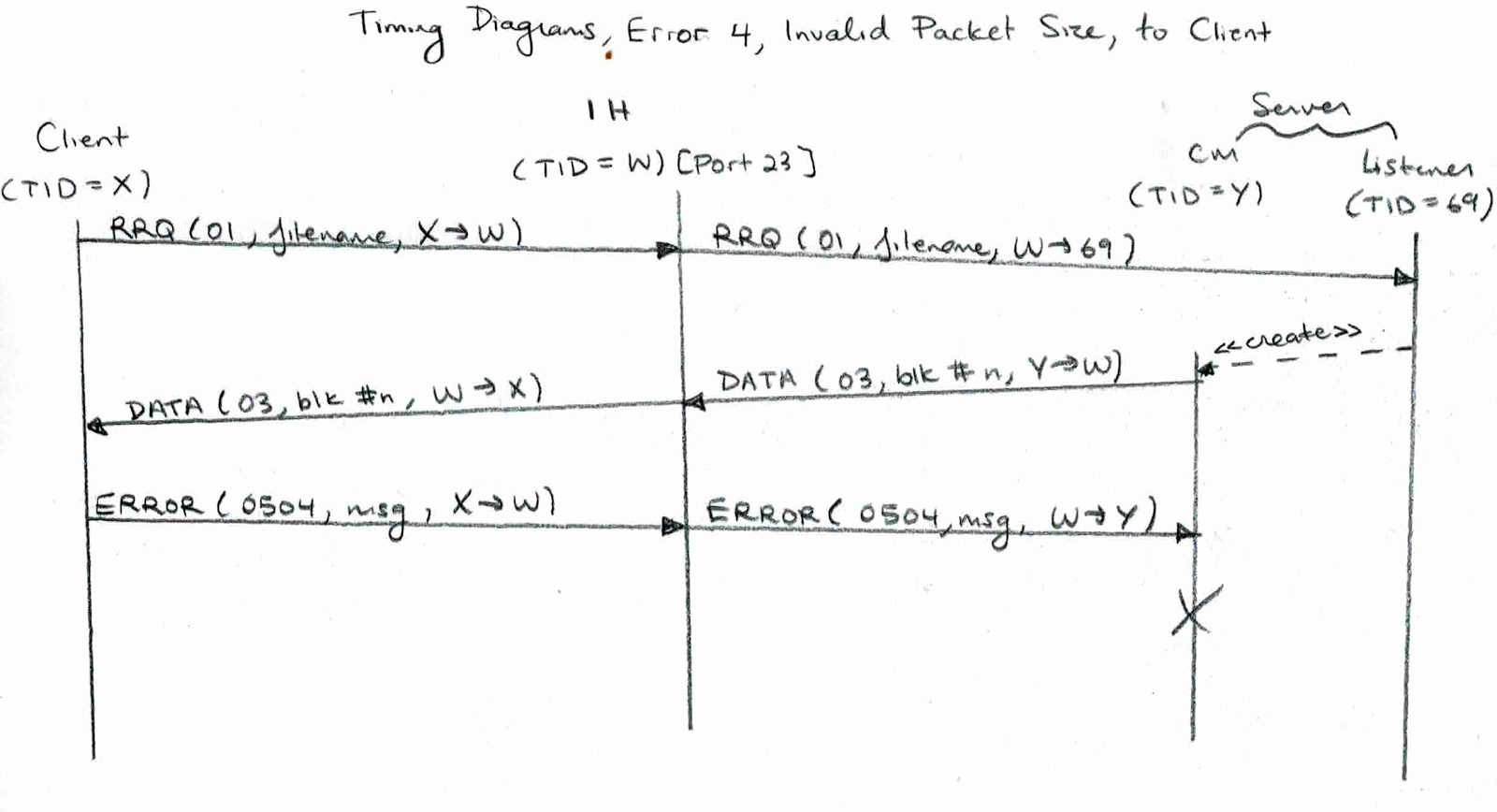
\*\* if RRQ changed to WRQ, there will be a file already exists error

\*\* WRQ to READ, file already exists error  
\*\*DATA to ACK or ACK to DATA, invalid opcode error

Error 4, Invalid Packet Size, to Server  
  


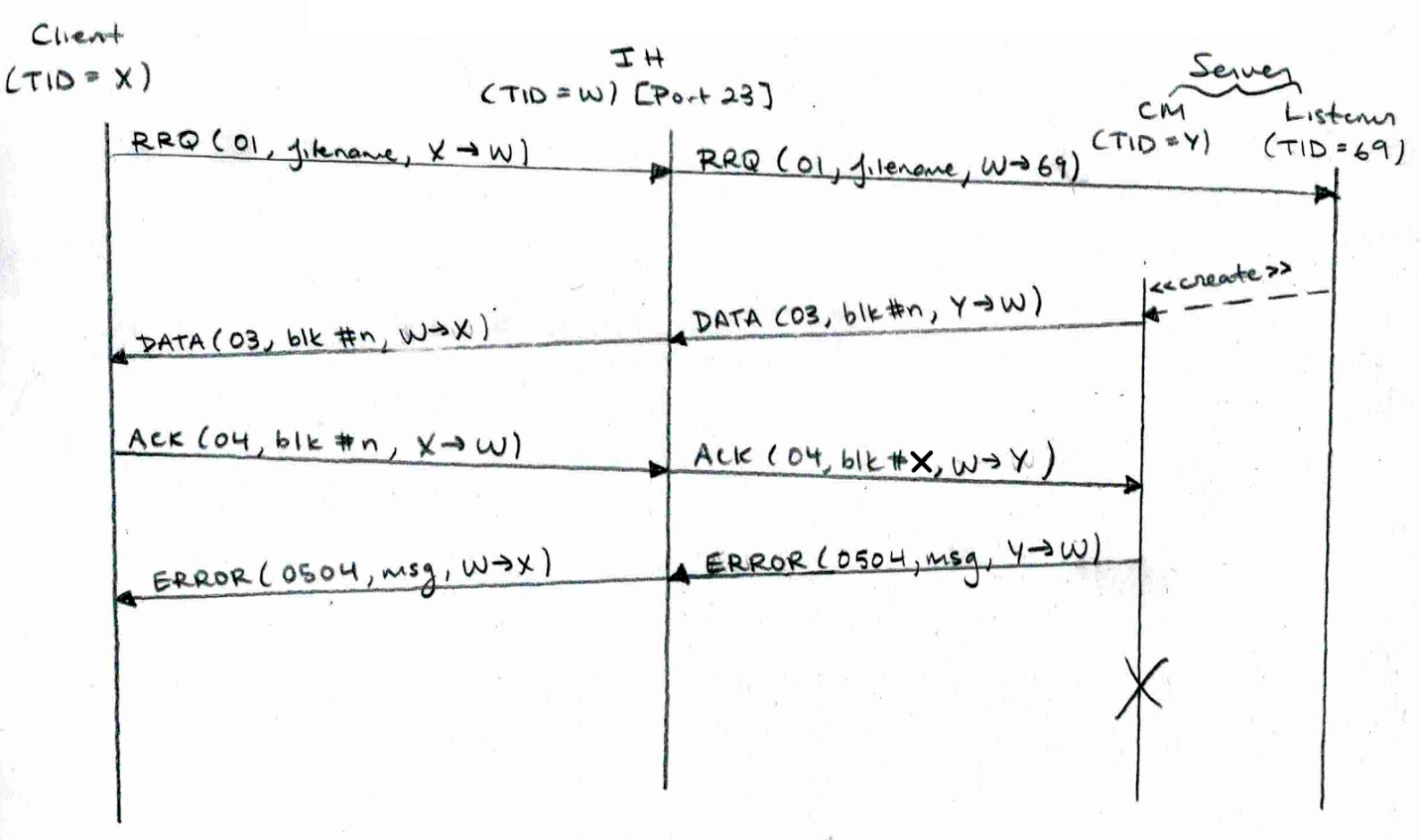
\*\* ACK made too big

Error 4, Invalid Packet Size, to Client



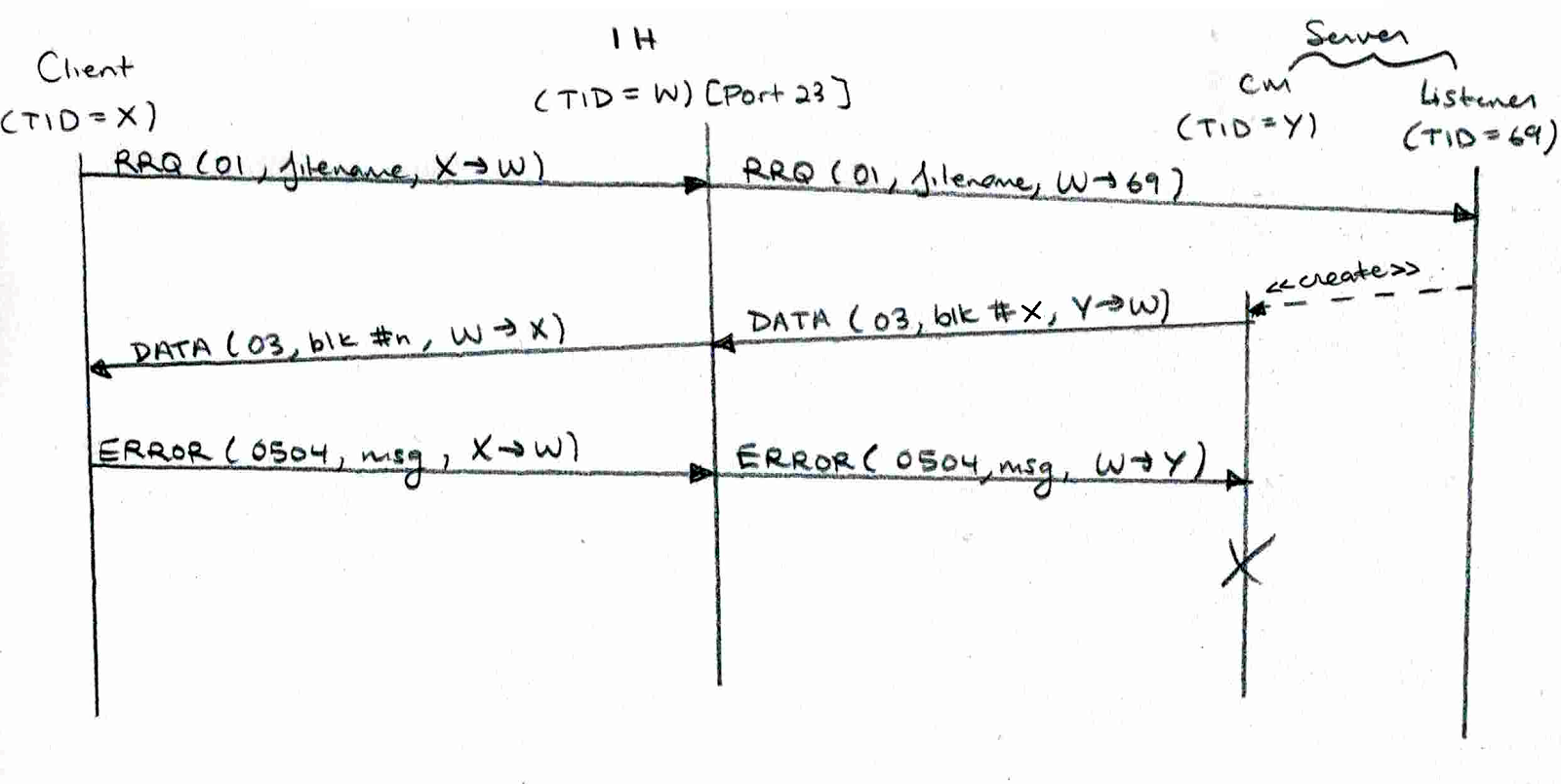
\*\* DATA packet made too large

## Error 4, Invalid Block Number, to Server



\*\* if repeated ACK, server will ignore

## Error 4, Invalid Block Number, to Client



\*\* if repeated DATA, client will send ACK and server will ignore repeated ACK

# Testing Instructions

## Disk Full

* obtain a USB disk that is basically full
* you can do this through the command line by running this command where path is the path of your USB and size is how much you want to fill your USB
  + Windows: fsutil file createnew <PATH> <SIZE>
  + Linux: mkfile <SIZE> <PATH>

Client

* set the path of the client to the USB
* Make a RRQ for a file in Server, and the file should stop reading as soon as the disk is full
* client will send an error packet back to client saying the disk is full

Server

* the path of the server directory must be changed to the USB’s path within the code
* server will send an error packet back to client saying the disk is full

## File Already Exists

Client

* run a read request on a file that is already if the client’s folder
* the client will re-prompt the user saying that the file already exists

Server

* run a write request on a file that is already in src/serverFiles
* the server will send an error packet back to client saying the file already exists
* client will be re-prompt

## File Not Found

Client

* run a read request on a file that does not exists in src/serverFiles
* the server will send an error to client saying that the file does not exist

Server

* run a write request on a file that does not exists in the client folder
* the client will re-prompt because the file doesn't exist

## File Access Denied

* Take the file that needs to be read or written and change its access to read-only
* this will cause the error to occur on both client and server side

## Illegal TFTP

* An illegal TFTP operation can be performed using the intermediate host
* The options in the Intermediate host that provide this error are:
  + Corrupt Request Packet
  + Change OpCode
  + Invalid Packet Size
  + Change the Block Number (if block is too large, an error will be send, if it’s a repeated number, it will be treated accordingly)

Follow all prompts accordingly to create an Illegal TFTP.

Unknown Transfer ID

* An unknown transfer ID can also be performed through the intermediate host
* Use Change TID prompt and follow the instructions accordingly to create the error

Intermediate Host – Normal, Lost, Delay, Duplicate

* Use the intermediate host to perform these actions
  + Normal: passes packets through IH without changing them
  + Lost: will lost the specified packet
  + Delay: will delay the specified packet for a user inputted time
  + Duplicate: will duplicate a packet after a user inputted time

Follow the prompts that intermediate host gives.

Intermediate Host – Manipulating Error Packet

* Use intermediate host to perform these actions
  + Lose an error packet
  + Delay an error packet
  + Duplicate an error packet
  + Send an error packet from an invalid TID
  + Change the opcode of an error packet
  + Change the code of an error packet

\*\* Note this can only be performed on a “naturally error inducing situation”, e.g. Write with file not found, file does not exist, access denied, disk full and read with disk full.