SYSC 3303 TFTP File Transfer System

Group 4

Benjamin Bichel, Jake MacDonald, Andrew Vicencio,
Paul Hewson, Geoffrey Scornaienchi

Table of Contents

1 Responsibilities

1.1 Iteration 1

Server:

- Jacob MacDonald
- Benjamin Bichel

Logger:

- Jacob MacDonald

Packet:

- Jacob MacDonald
- Andrew Vicencio

Tools:

- Jacob MacDonald
- Paul Hewson
- Andrew Vicencio
- Geoffrey Scornaienchi
- Benjamin Bichel

Error Simulator:

- Geoffrey Scornaienchi

Client:

- Paul Hewson
- Andrew Vicencio

UML Class Diagram:

- Geoffrey Scornaienchi

UCM, Timing Diagrams

- Benjamin Bichel

1.2 Iteration 2

Server:

- Jacob MacDonald
- Benjamin Bichel

Logger:

- Jacob MacDonald

Packet:

- Jacob MacDonald
- Andrew Vicencio

Tools:

- Jacob MacDonald
- Paul Hewson
- Andrew Vicencio
- Geoffrey Scornaienchi
- Benjamin Bichel

Error Simulator:

- Geoffrey Scornaienchi

Client:

- Paul Hewson
- Andrew Vicencio

UML Class Diagram:

- Geoffrey Scornaienchi

UCM, Timing Diagrams

- Benjamin Bichel

1.3 Iteration 3

Server:

- Jacob MacDonald
- Benjamin Bichel

Logger:

- Jacob MacDonald

Packet:

- Jacob MacDonald
- Andrew Vicencio

Tools:

- Jacob MacDonald
- Paul Hewson
- Andrew Vicencio
- Geoffrey Scornaienchi
- Benjamin Bichel

Error Simulator:

- Geoffrey Scornaienchi

Client:

- Paul Hewson
- Andrew Vicencio

CommandLine:

- Andrew Vicencio
- Jake MacDonald

UML Class Diagram:

- Geoffrey Scornaienchi

UCM, Timing Diagrams

- Paul Hewson
- Geoffrey Scornaienchi

1.4 Iteration 4

Server:

- Jacob MacDonald
- Benjamin Bichel

Logger:

- Jacob MacDonald

Packet:

- Jacob MacDonald
- Andrew Vicencio

Tools:

- Jacob MacDonald
- Paul Hewson
- Andrew Vicencio
- Geoffrey Scornaienchi
- Benjamin Bichel

Error Simulator:

- Geoffrey Scornaienchi
- Paul Hewson

Client:

- Paul Hewson
- Andrew Vicencio

CommandLine:

- Andrew Vicencio
- Jake MacDonald

UML Class Diagram:

- Geoffrey Scornaienchi

1.5 Final Submission

Server:

- Jacob MacDonald
- Benjamin Bichel

Logger:

- Jacob MacDonald

Packet:

- Jacob MacDonald
- Andrew Vicencio

Tools:

- Jacob MacDonald
- Paul Hewson
- Andrew Vicencio
- Benjamin Bichel

Error Simulator:

- Jacob MacDonald

Client:

- Paul Hewson
- Andrew Vicencio

CommandLine:

- Andrew Vicencio
- Jake MacDonald

UML Class Diagram:

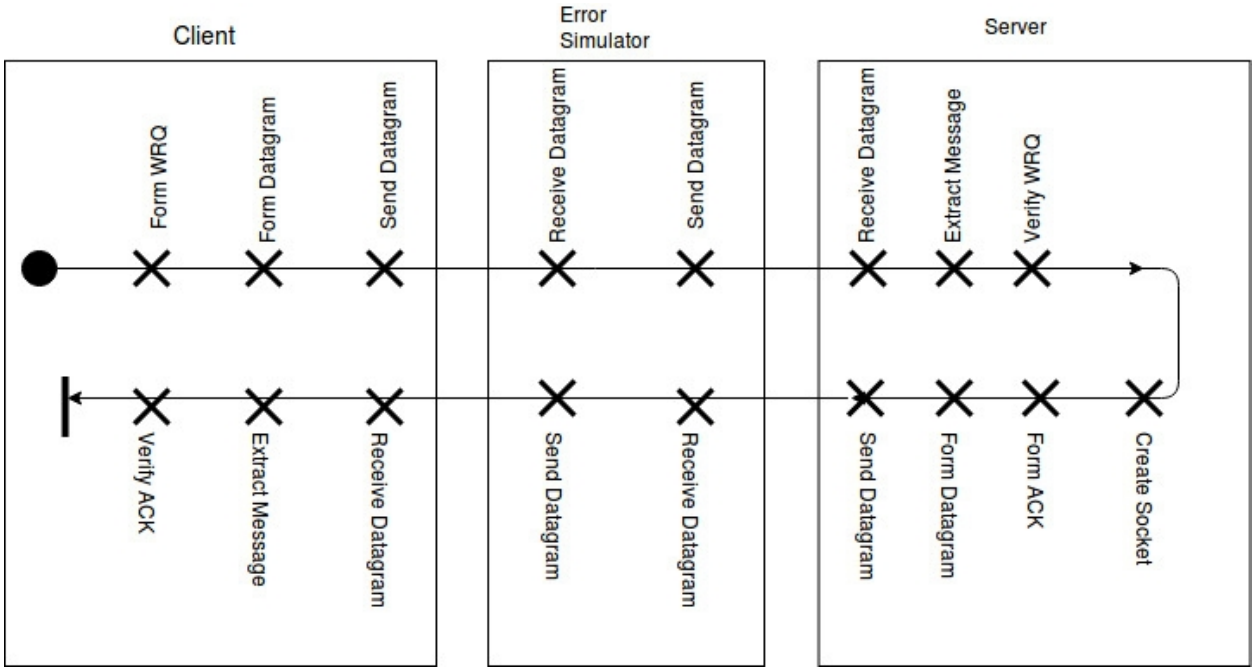
- Geoffrey Scornaienchi

Report

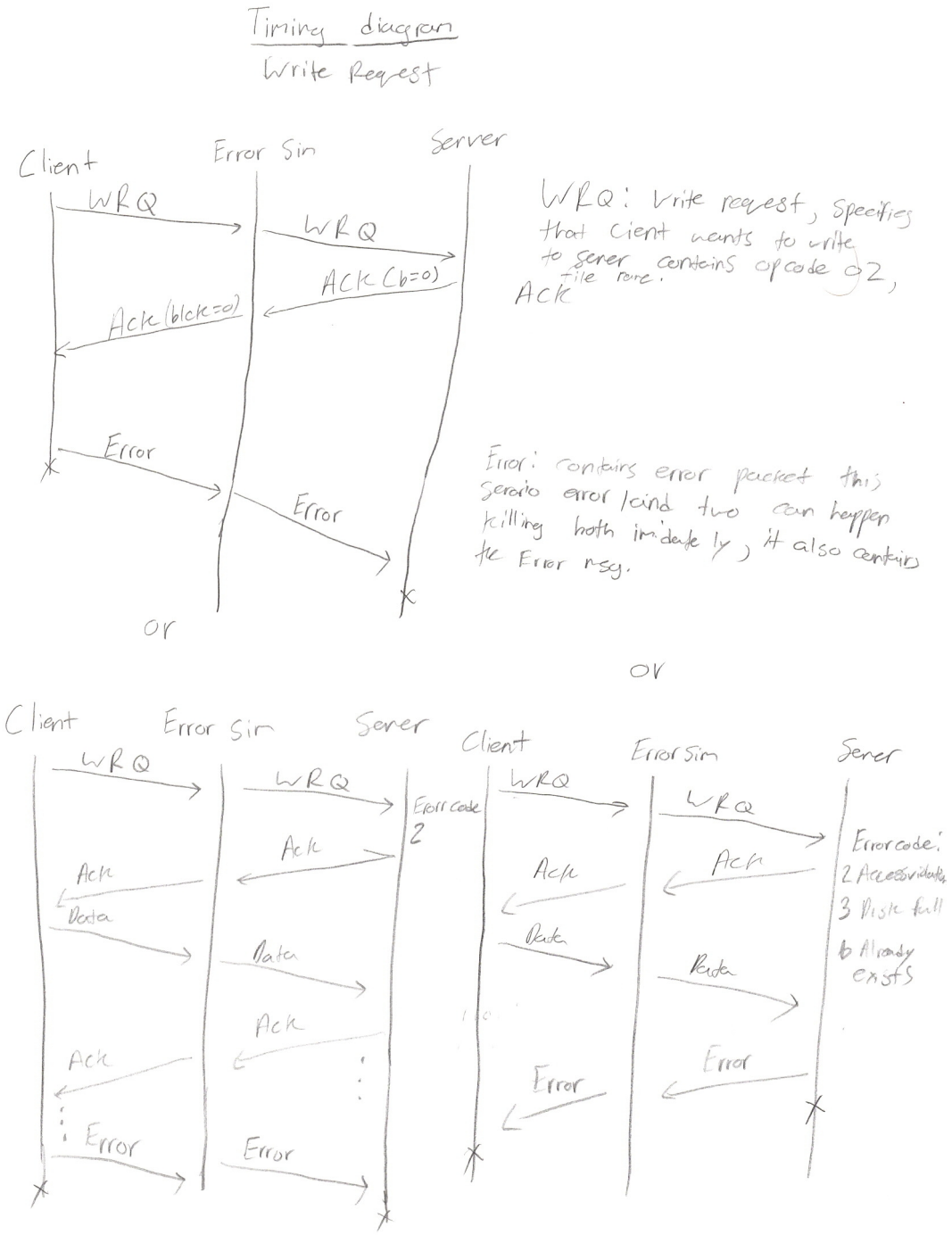
- Geoffrey Scornaienchi
- Paul Hewson

2 Diagrams

2.1 UCM Diagram

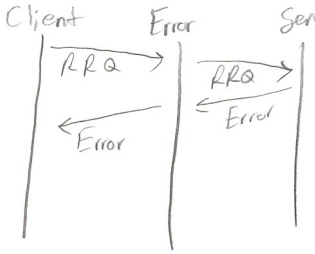


2.2 Timing Diagrams

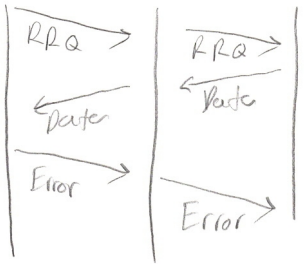


Timing diagram

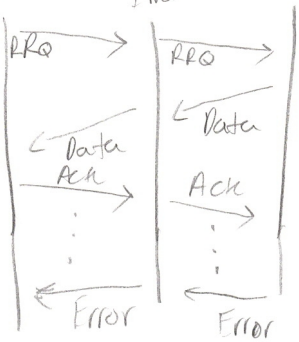
Read Request



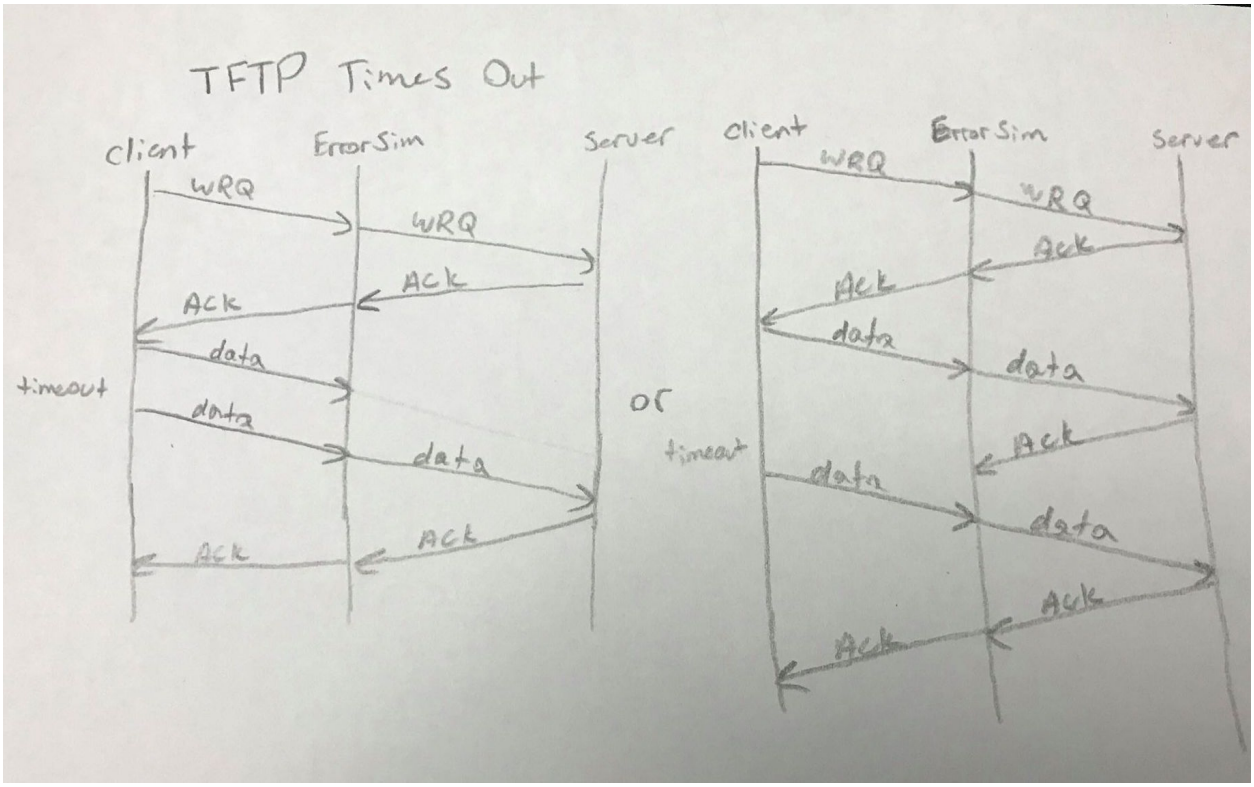
Error codes 1, 2, can occur at this time



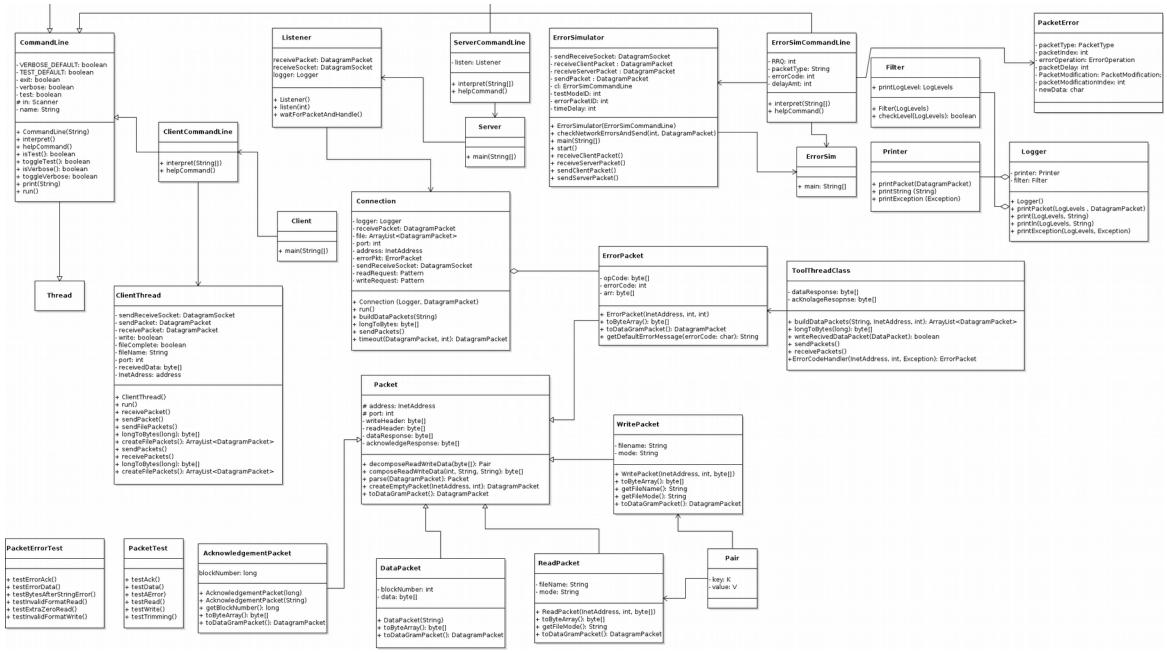
Error codes 2, 3, 6 can occur at this time



Error code 2 can happen at this time


TFTP Times Out
client
ErrorSim
server
WRQ
WRQ
ACK
ACK
data
timeout
data
data
ACK
ACK
or
client
ErrorSim
server
WRQ
WRQ
Ack
Ack
data
data
timeout
Ack
data
data
Ack
Ack

2.3 UML Class Diagram



3. User Manual

3.1 Setup

1. Open eclipse
2. Create new workspace
3. Import existing projects
4. Select root directory
5. Finish
6. Run -> Run Configurations
7. Double click JavaApplication
8. Change name to Server
9. Select path to project
10. Select path to Server class
11. Apply
12. Repeat steps 7-11 for Client and ErrorSim
13. Double click Launch Group
14. Change name to WithoutErrorSim
15. Add Client and Server
16. Repeat steps 13-15 with name WithErrorSim, and adding ErrorSim as well as Client and Server
17. Run WithoutErrorSim to run just Client and Server
18. Run WithErrorSim to run Client and Server with an Error Simulator

3.2

1. In Client's console, this prompt will show up:
   Client Command line ready.
   Verbose: true
   Test: true
   [--verbose], [--test], [HELP], [EXIT], or [CONTINUE]

   Inputs:
   --verbose will toggle verbose mode and display the next prompt.
   --test will toggle test mode and display the next prompt.
   HELP displays a help message.
   CONTINUE will go to the next prompt.
   EXIT exits the console.

2. Would you like to read or write?

   Inputs:
   read to read a file from the server.
   write to write a file to the server.

3. What file would you like to read/write?

   Inputs:
   Name of file to be sent, including file extension.

4. What address would you like to talk to? type local for local address

   Inputs:
   IP address of the device the server is running on.

Example:

Client Command line ready.
Verbose: true
Test: true
[--verbose], [--test], [HELP], [EXIT], or [CONTINUE]
>> --test
Would you like to read or write?
>> Write
What file would you like to write?
>> MobyDick.txt
What address would you like to talk too? type local for local address
>> 192.168.0.15
Client socket created.
Client Command line ready.
Verbose: true
Test: true
[--verbose], [--test], [HELP], [EXIT], or [CONTINUE]
Client - Sending packet to /192.168.0.15 Port 1069
Client - Packet sent.
Starting file transfer.
Waiting2.0
…
Send finished.