SYSC3303 TFTP: Team 4

Final Submission

Naol Gushu: 100911600

Melaku Semaw: 101059910

Keith Ko: 100973372

Chengyang Liu: 101011773

Raymond Wu: 100938326

August 8, 2018

Content

[1. Descriptions 1](#_Toc521568581)

[1.1 Client Side 2](#_Toc521568582)

[1.1.1 Client can send read requests to read exists files in server side. 2](#_Toc521568583)

[1.1.2 Client can send write requests to write into files in server side. 2](#_Toc521568584)

[1.1.3 Client is permitted to write while receiving an acknowledge packet from server. 2](#_Toc521568585)

[1.2 Server Side 2](#_Toc521568586)

[1.2.1 Server will respond with a file when received a valid read request. 2](#_Toc521568587)

[1.2.2 Server will respond with an acknowledgement file when received a valid write request. 2](#_Toc521568588)

[1.2.3 Server will committee the valid writing received from client. 2](#_Toc521568589)

[1.3 Error Simulator 2](#_Toc521568590)

[1.2.1 Error simulator will capture all the packet sent from client or server. 2](#_Toc521568591)

[1.2.2 Error simulator will monitor various errors which will happen in transaction. 2](#_Toc521568592)

[1.2.3 Server will committee the valid writing received from client. 2](#_Toc521568593)

[2.Instruction 2](#_Toc521568594)

[2.1 Run the TftpErrorSimulator.java to set up error simulator. 2](#_Toc521568595)

[2.2 Run the TftpServer.java to set up TFTP server. 2](#_Toc521568596)

[2.3 Run the Client.java to set up client. 2](#_Toc521568597)

[2.4 Type the command in console to decide client behavior. 3](#_Toc521568598)

[2.5 Type the command in console to decide error simulator behavior. 3](#_Toc521568599)

[3.Code Structure 3](#_Toc521568600)

[3.1 The whole program contains totally 11 java source files and 1 txt test file. The more specific structure can refer the UML(Part 4). 3](#_Toc521568601)

[3.2 File Contents 3](#_Toc521568602)

[4.UML diagram 5](#_Toc521568603)

[5.UCM diagram 5](#_Toc521568604)

[6.Timing diagram 5](#_Toc521568605)

# 1. Descriptions

This system is a designed for transferring files between 2 computers with TFTP protocol. It basically contains 2 entities: server and client. And we add a error simulator to simulating when errors happened, how will the system handles them. The main features will be listed below.

## 1.1 Client Side

### 1.1.1 Client can send read requests to read exists files in server side.

### 1.1.2 Client can send write requests to write into files in server side.

### 1.1.3 Client is permitted to write while receiving an acknowledge packet from server.

## 1.2 Server Side

### 1.2.1 Server will respond with a file when received a valid read request.

### 1.2.2 Server will respond with an acknowledgement file when received a valid write request.

### 1.2.3 Server will committee the valid writing received from client.

## 1.3 Error Simulator

### 1.2.1 Error simulator will capture all the packet sent from client or server.

### 1.2.2 Error simulator will monitor various errors which will happen in transaction.

### 1.2.3 Server will committee the valid writing received from client.

# 2.Instruction

## 2.1 Run the TftpErrorSimulator.java to set up error simulator.

## 2.2 Run the TftpServer.java to set up TFTP server.

## 2.3 Run the Client.java to set up client.

## 2.4 Type the command in console to decide client behavior.

Command List:

read filename - read file from server

write filename - write file to server

quit - stops the client

mode - Toggles between quiet and verbose mode

dir - prints the current directory for file transfers

test - Toggles between normal and test mode

host - Outputs the server address and port

server - set the ip or hostname of the server

## 2.5 Type the command in console to decide error simulator behavior.

Command List:

(0): normal operation

(1): request packet

(2): data packet

(3): ack packet

(10): lose a packet

(11): delay a packet

(12): duplicate a packet

(13): Invalid TID

(any other value): quit

# 3.Code Structure

## 3.1 The whole program contains totally 11 java source files and 1 txt test file. The more specific structure can refer the UML(Part 4).

## 3.2 File Contents

.DS\_Store

Client$Opcode.class

Client$TftpAck.class

Client.class

Client.java

ErrorSimulator.java

LIST.TXT

Proxy.class

TftpAck.class

TftpAck.java

TftpClientConnectionThread.class

TftpClientConnectionThread.java

TftpData.class

TftpData.java

TftpDelayThread.java

TftpError.java

TftpException.java

TftpPacket.class

TftpPacket.java

TftpRequest.class

TftpRequest.java

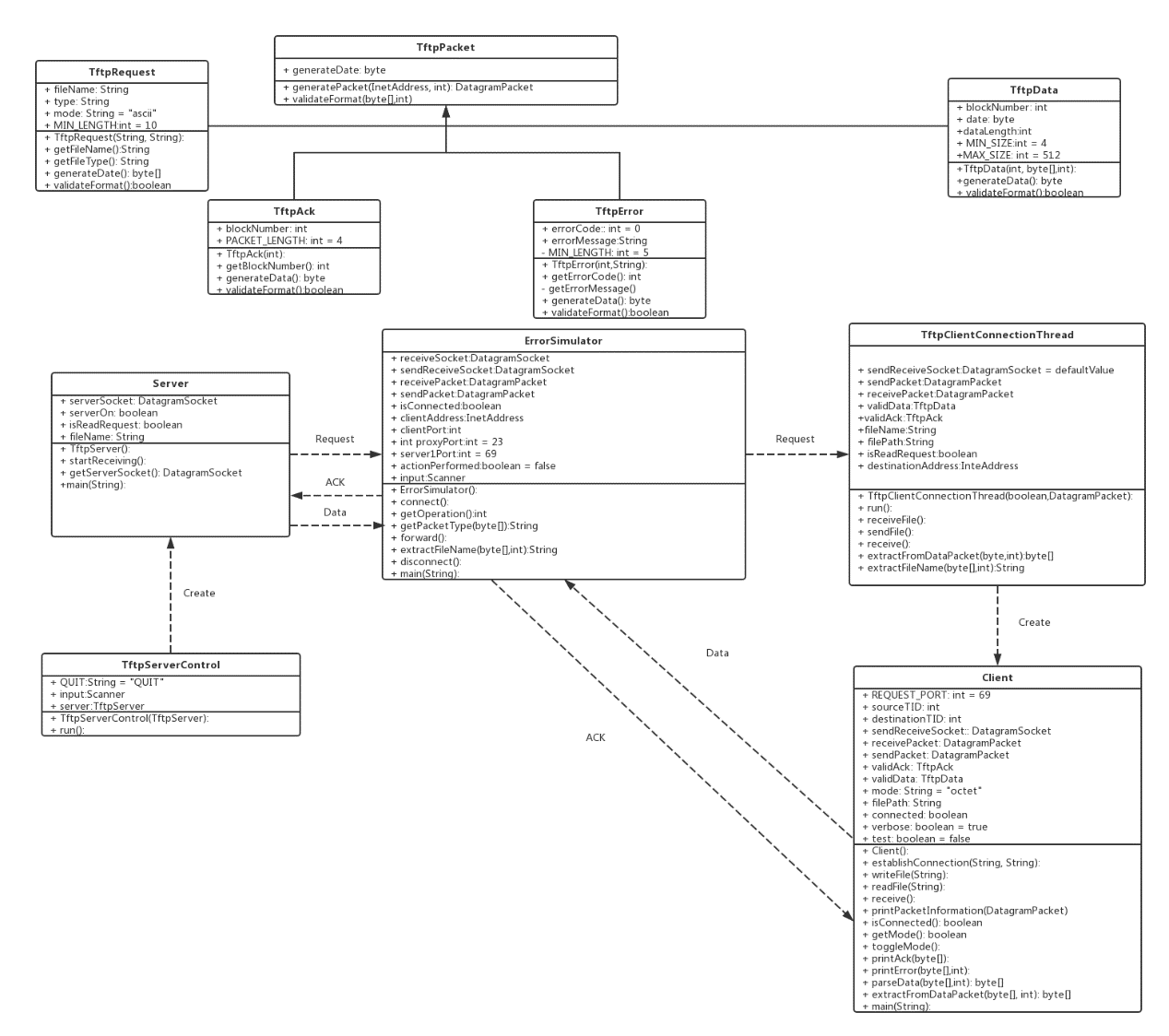
TftpServer.class

TftpServer.java

TftpServerControl.class

TftpServerControl.java

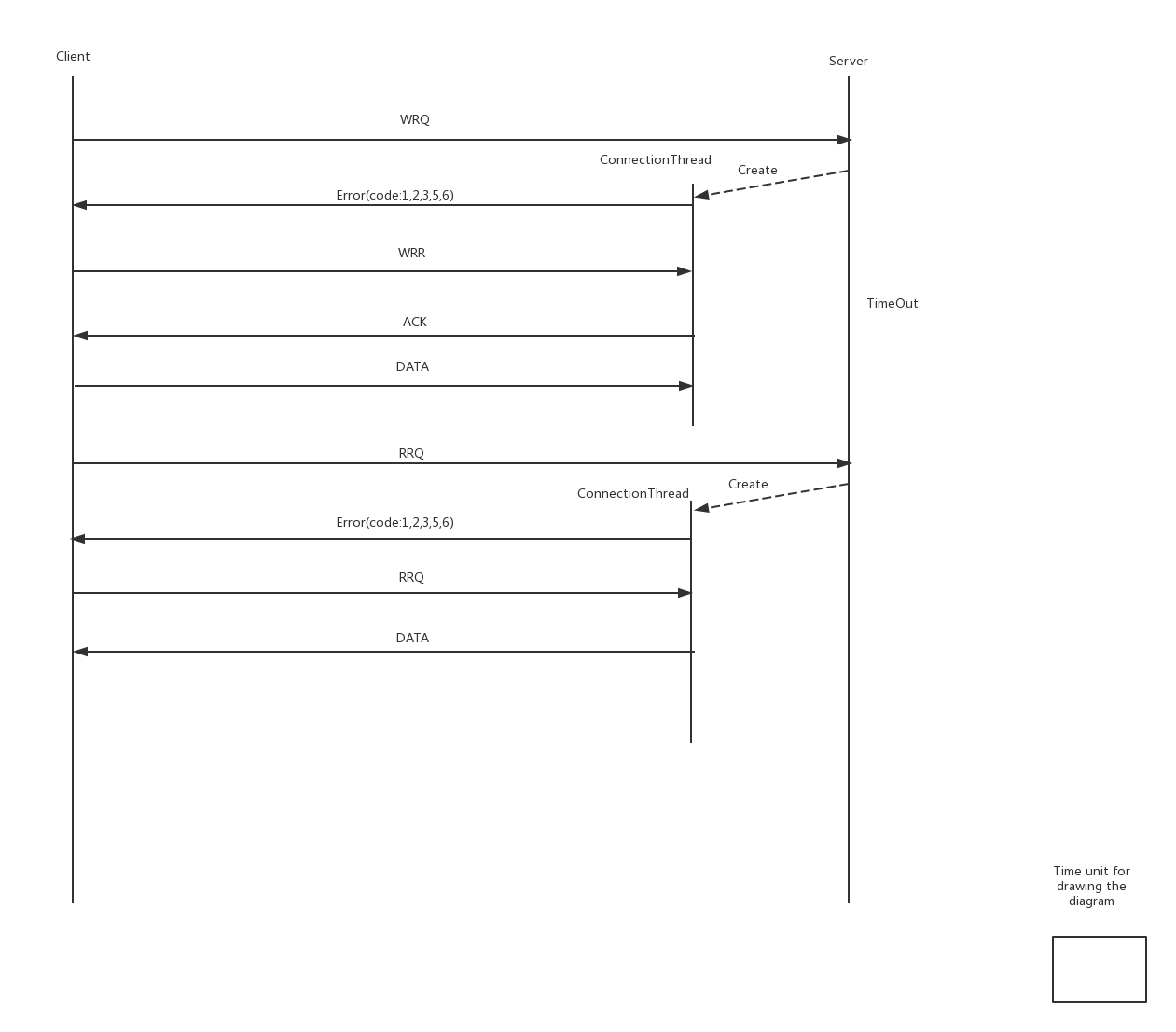
# 4.UML diagram



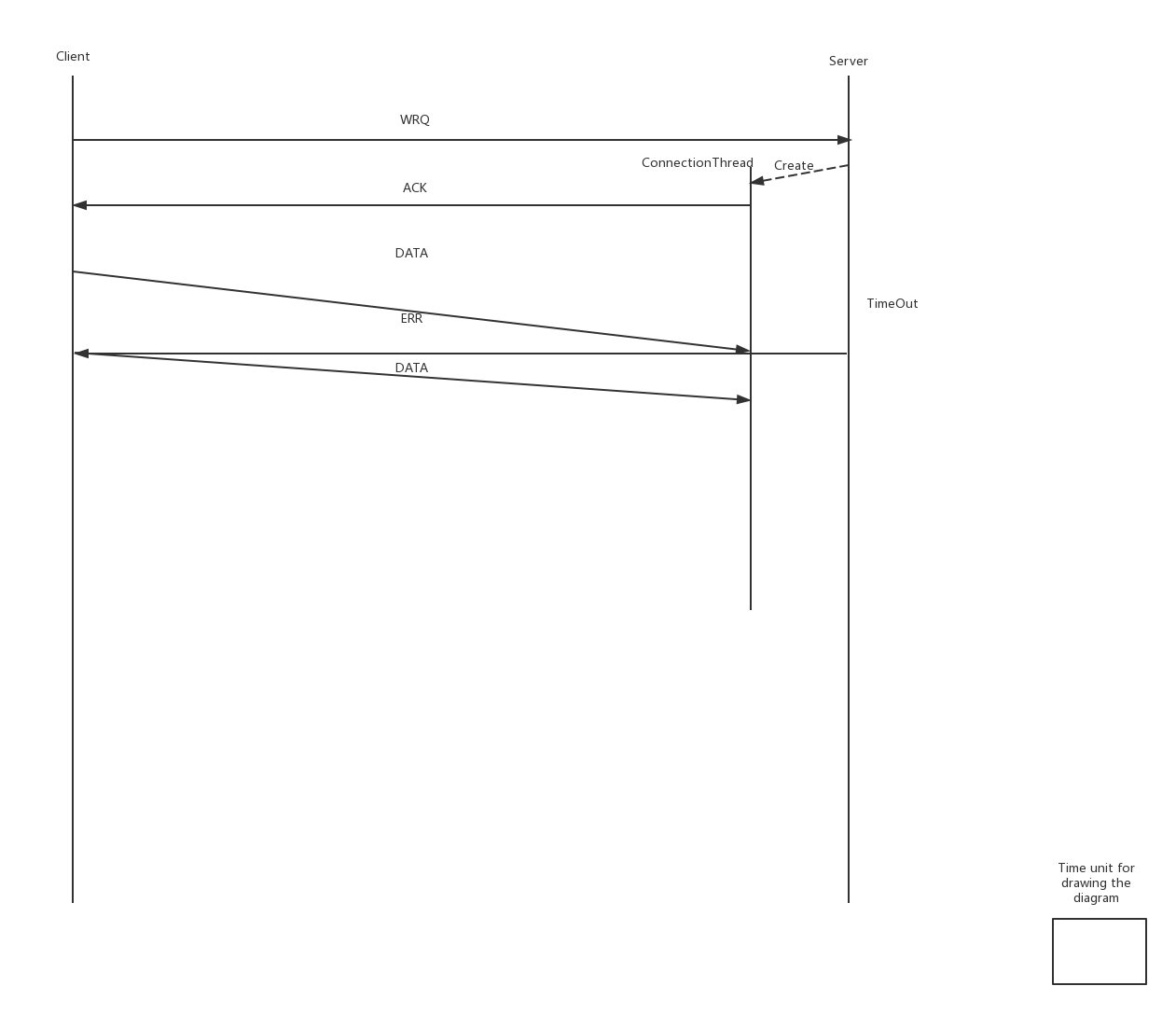
# 5.UCM diagram

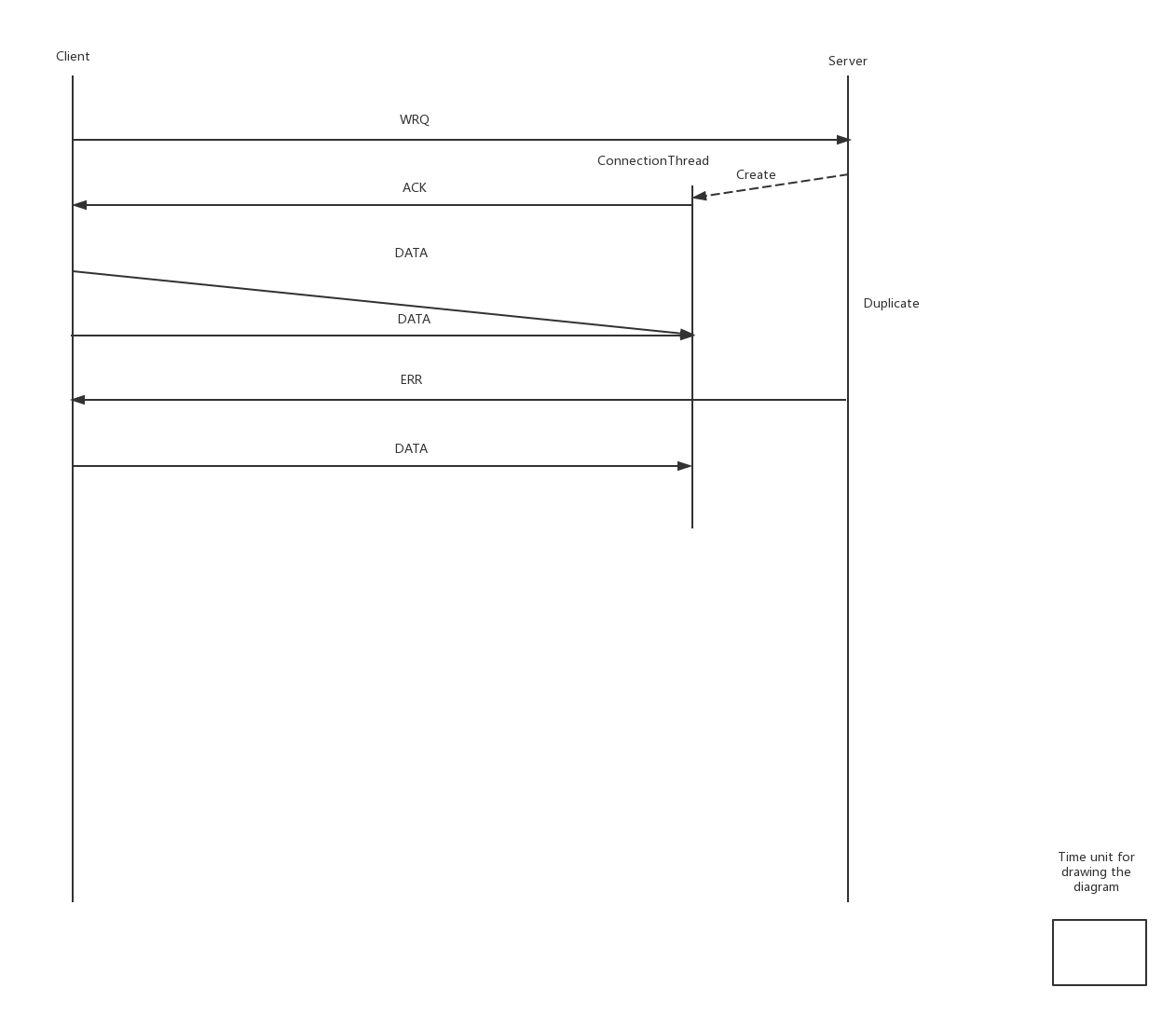
# 6.Timing diagram

ErrorCode1，2，3，5，6

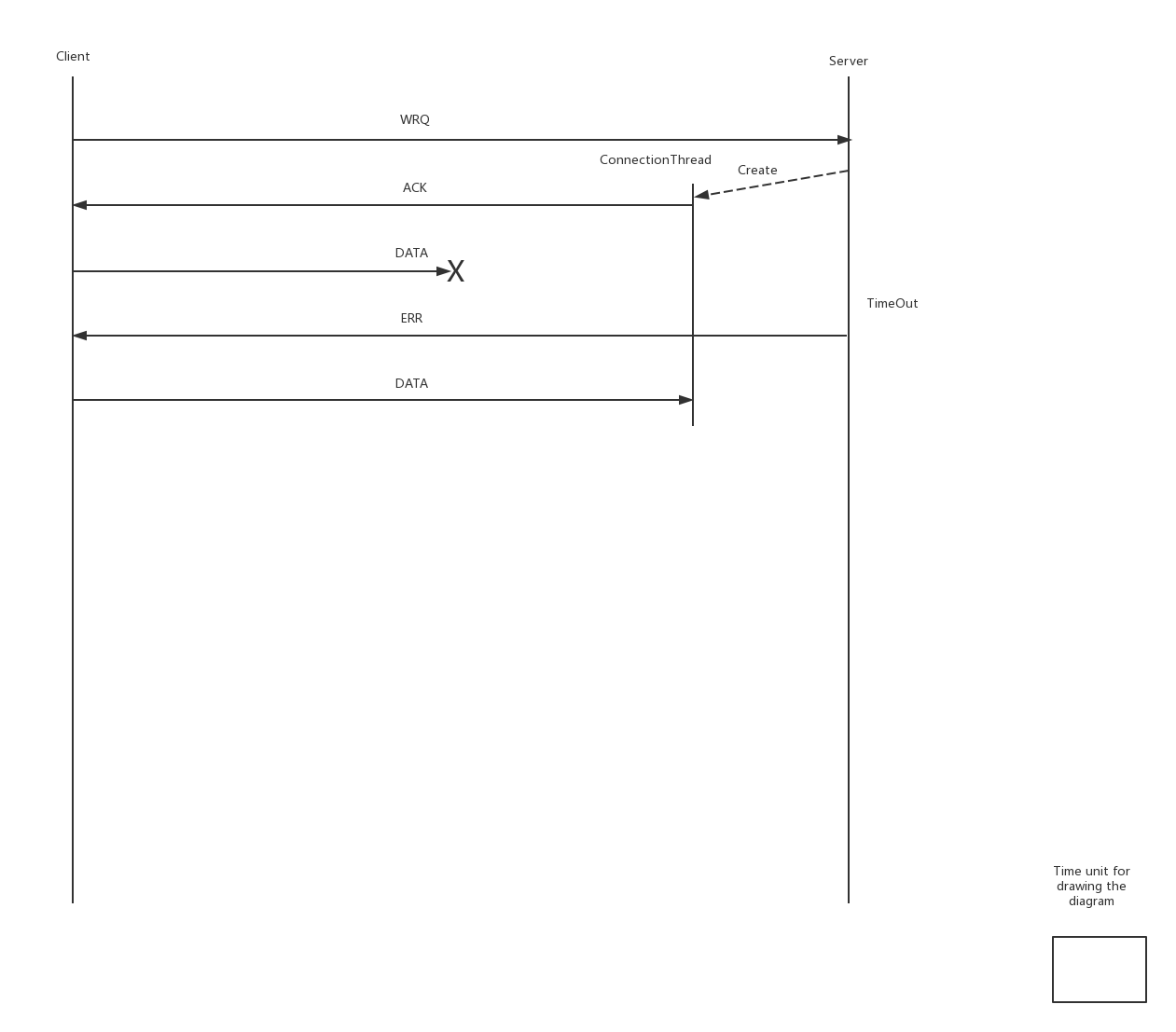


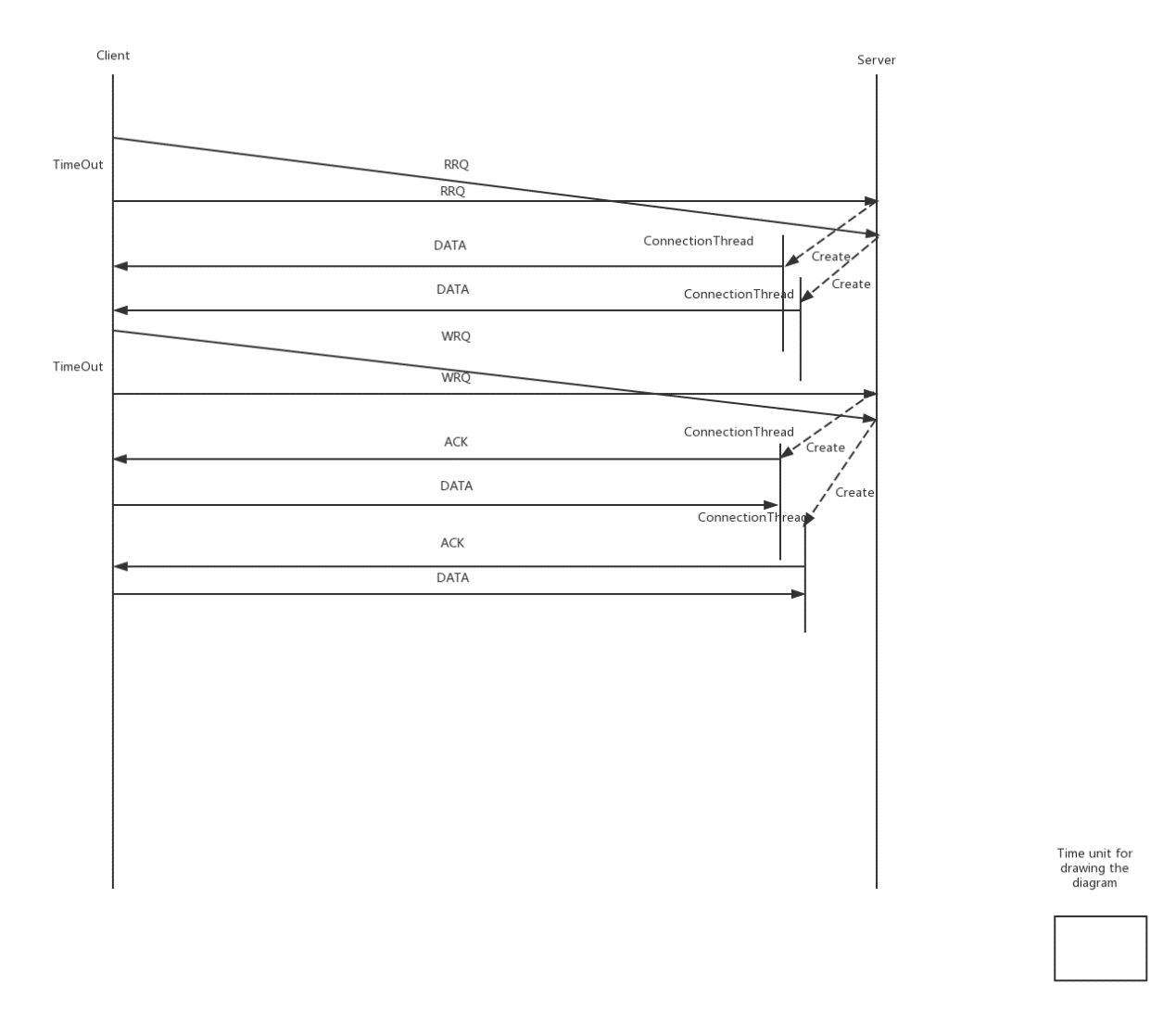
Client-DATA-Delay

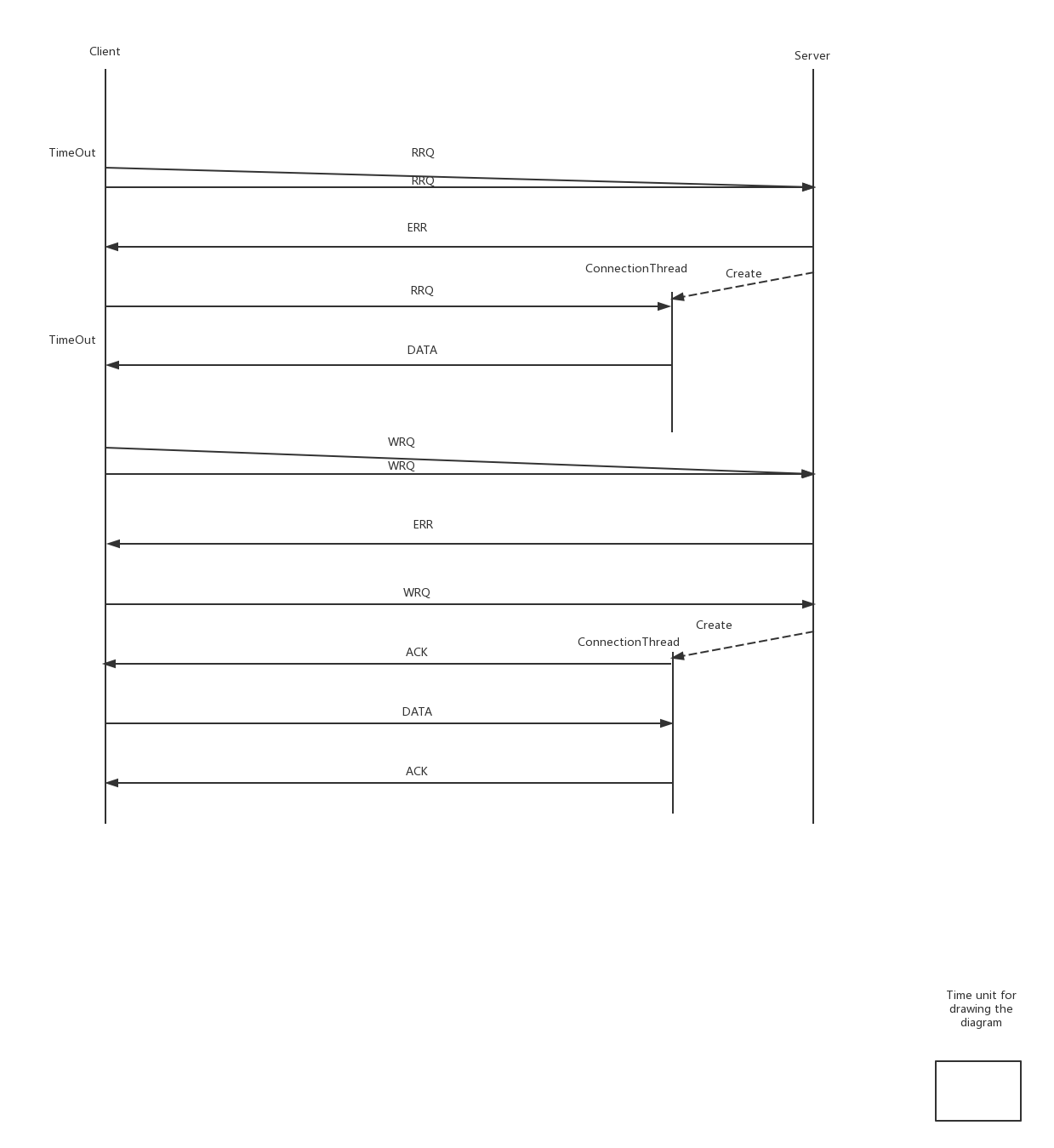


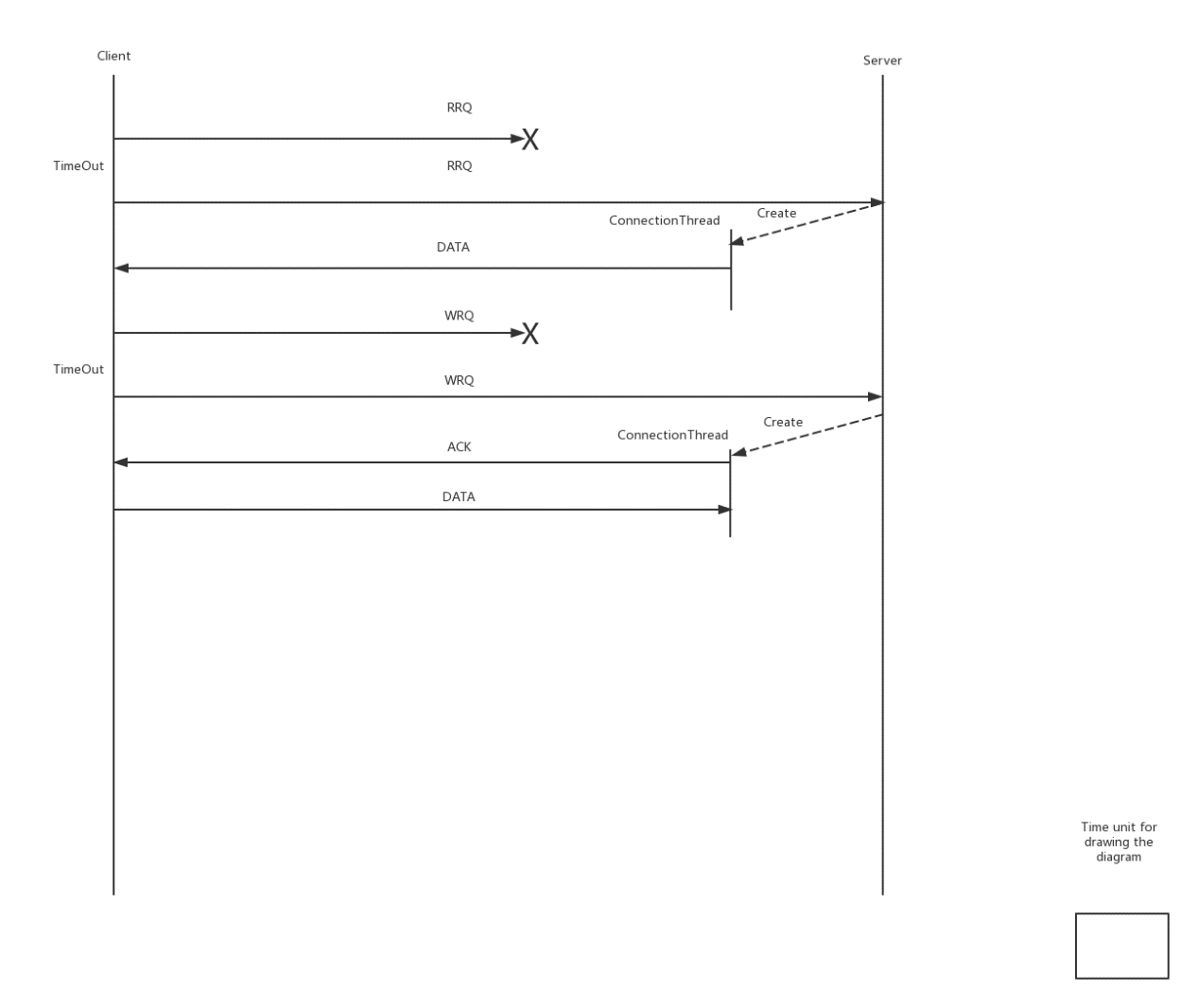
Client-DATA-Duplicate

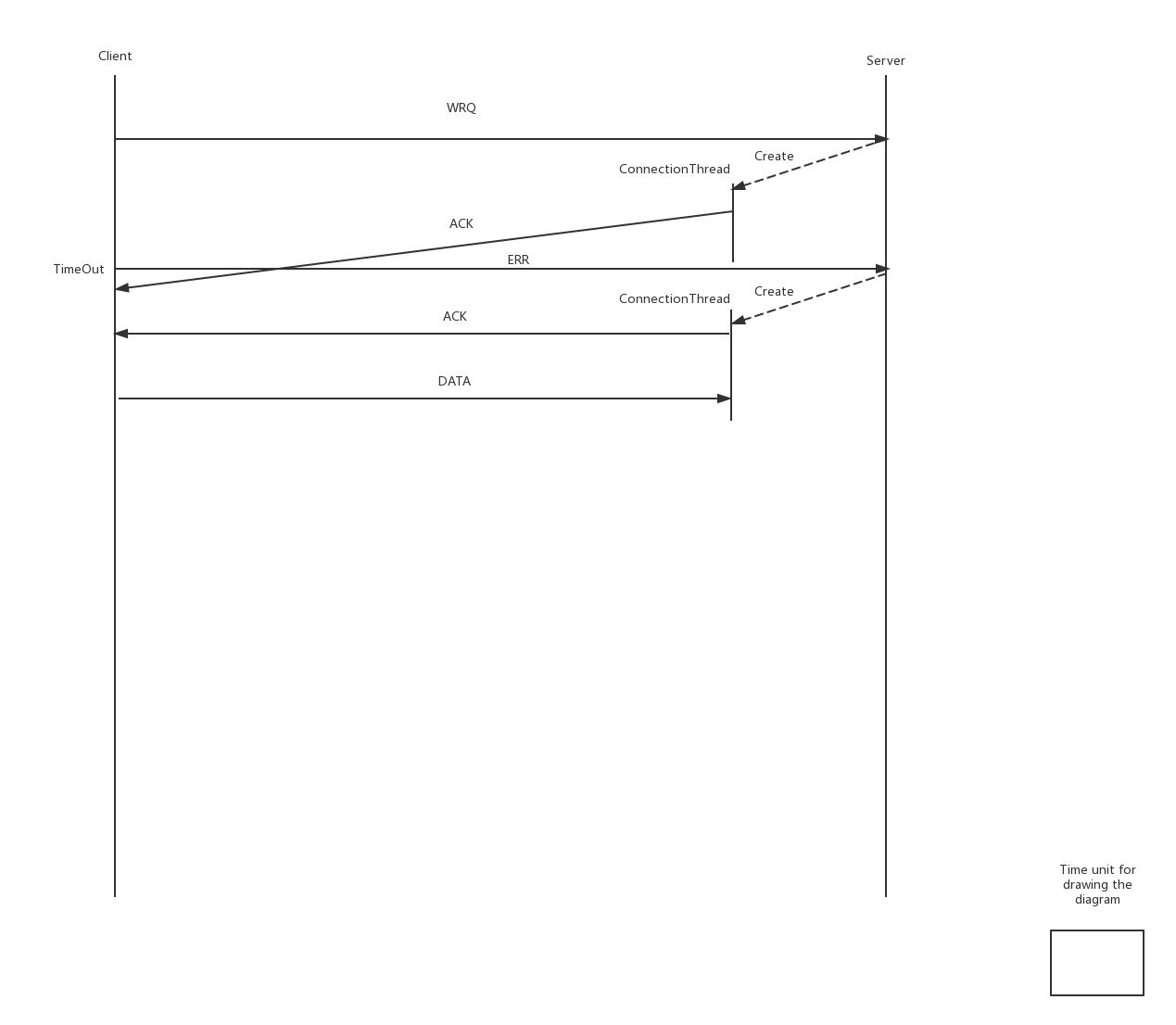
Client-DATA-Lost

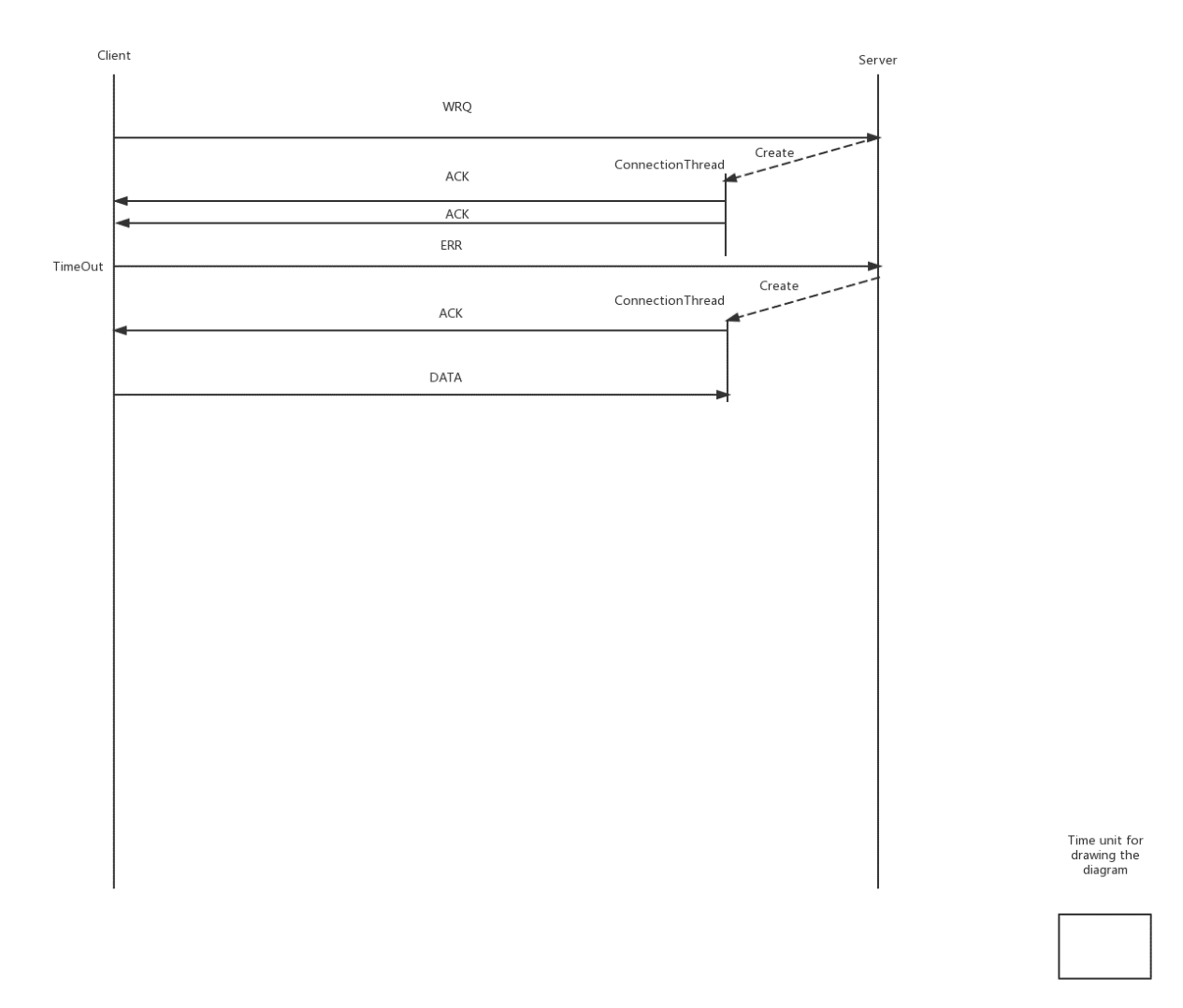


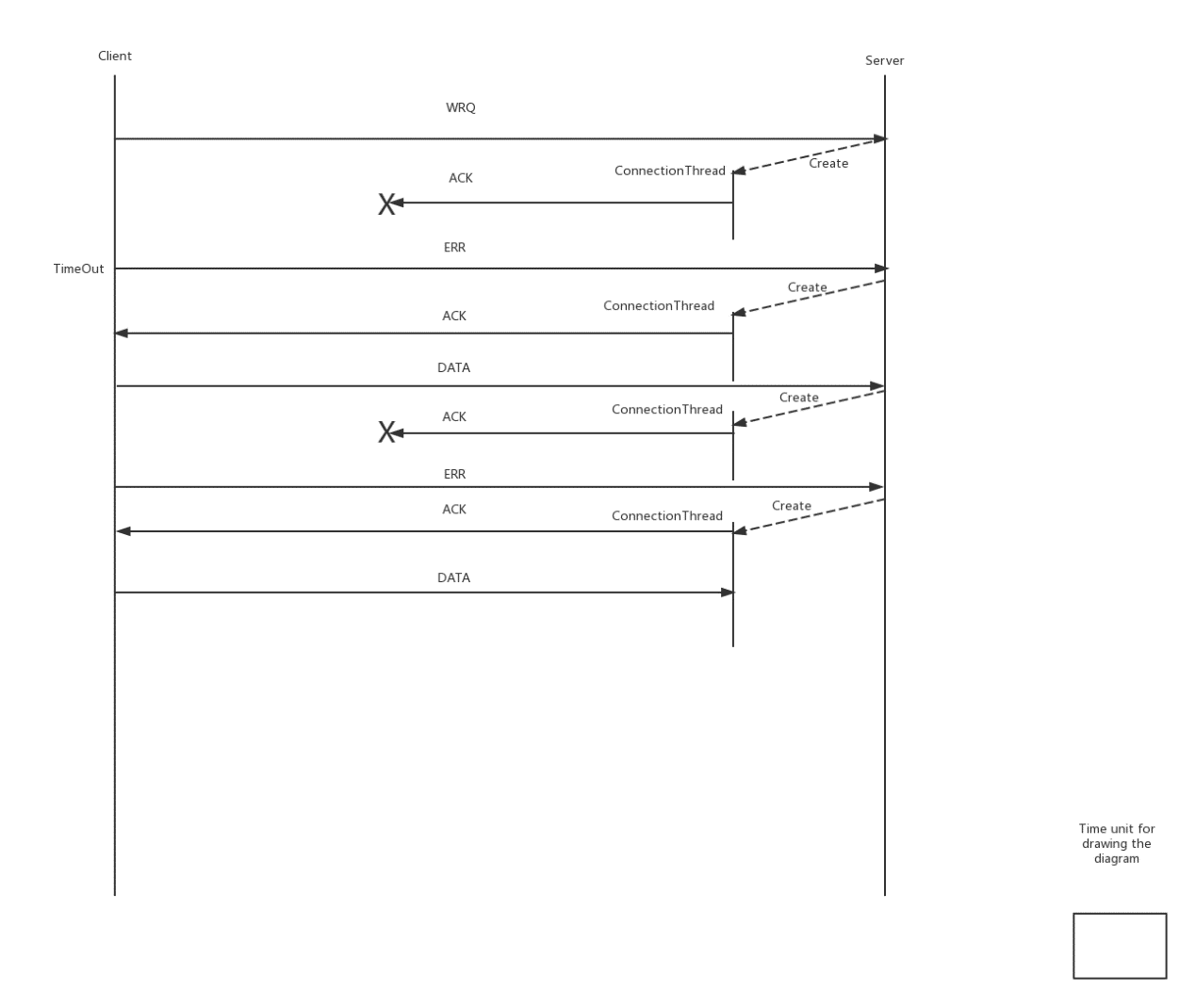
Client-Request-Delay

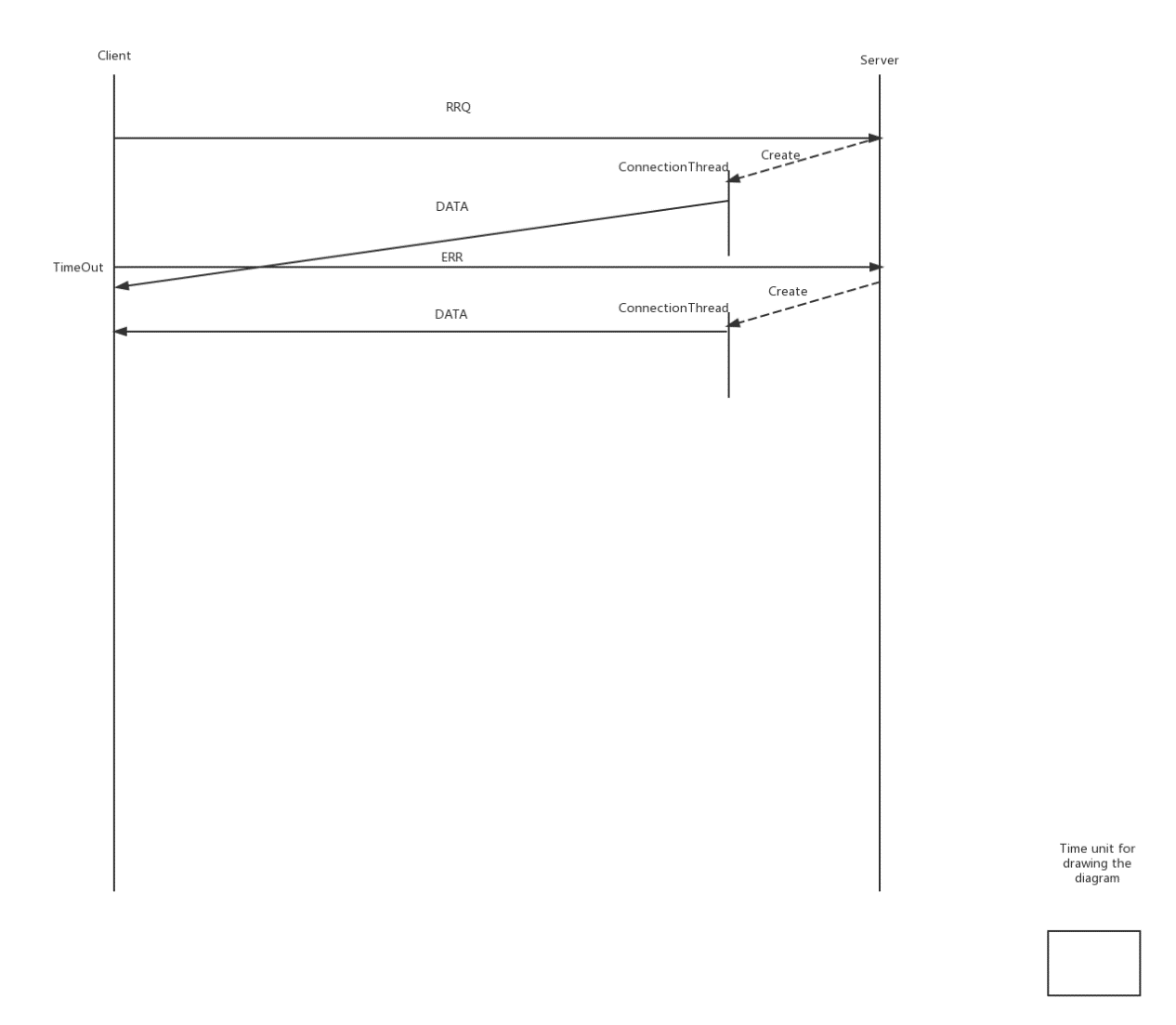
Client-Request-Duplicate

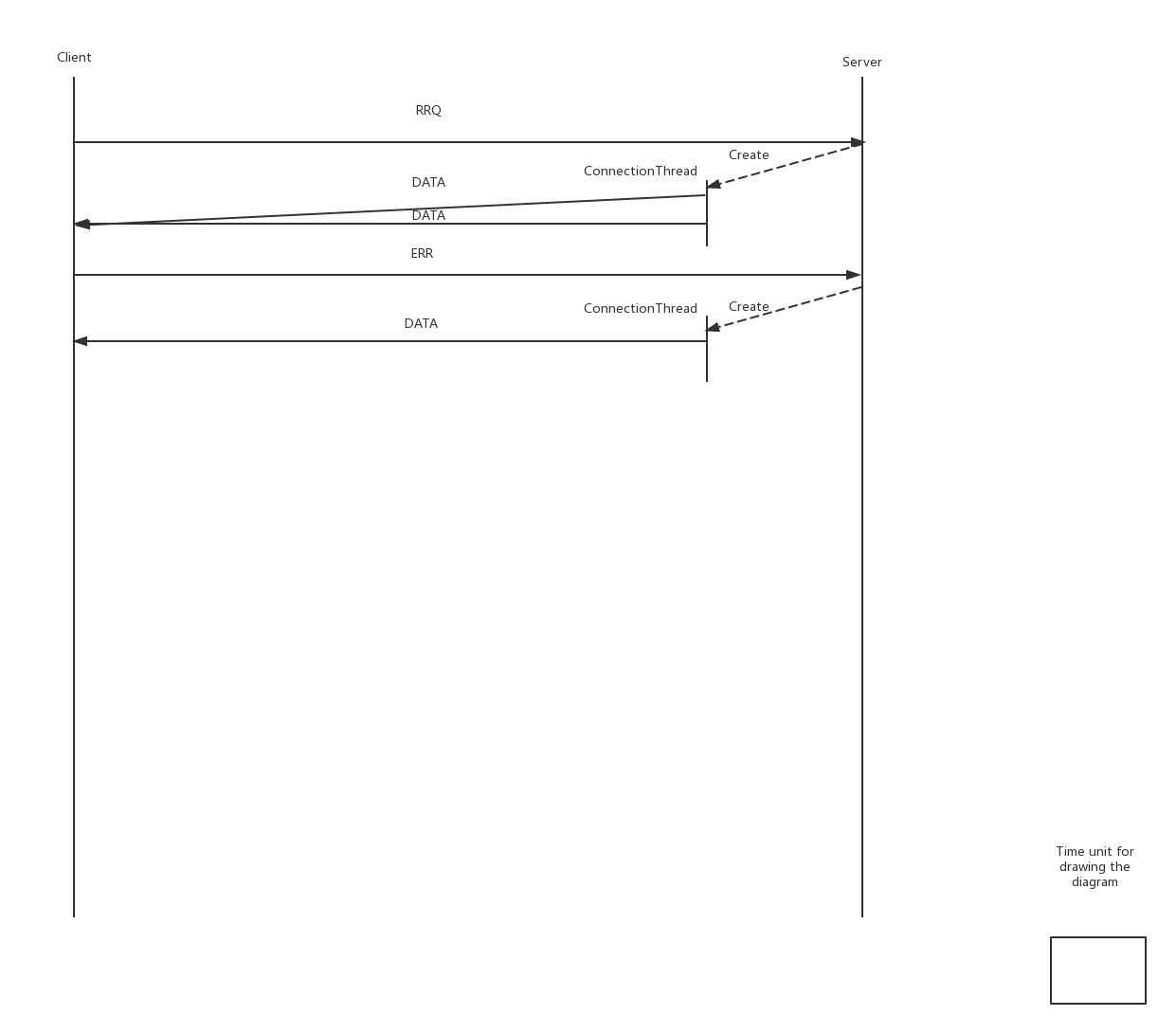
Client-Request-Lost

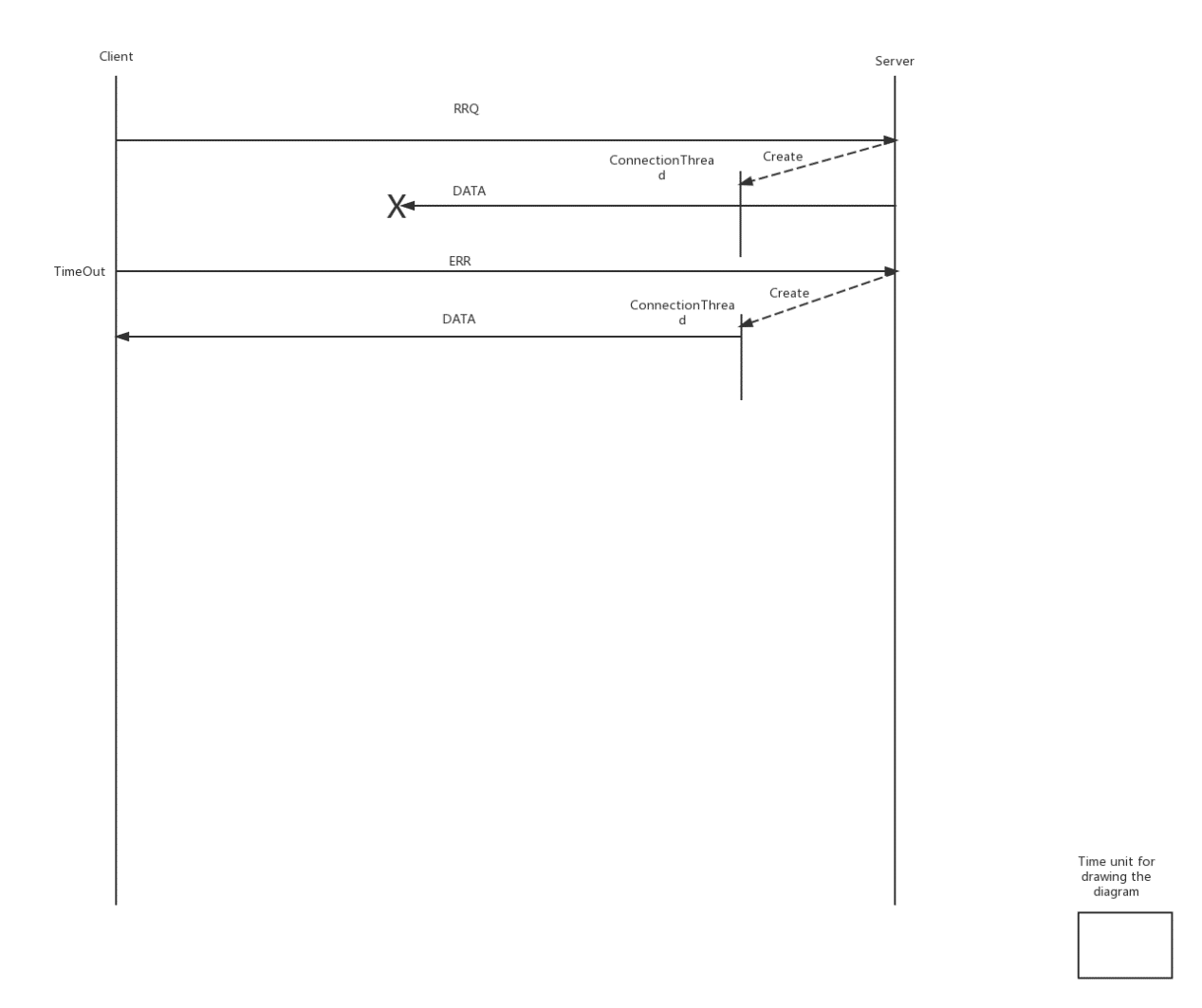
Server-ACK-Delay

Server-ACK-Duplicate

Server-ACK-Lost

Server-Data-Delay

Server-Data-Duplicate

Server-Data-Lost

# 7.Responsibility