

Implementing Reliability in UDP Application Layer

Computer Networking Project

By Sanjay Dookhoo, Sachin Rajkumar and Shanea Lewis

Group Name: Client/Server Hackers

Table of Contents

[Introduction 3](#_Toc468128113)

[About udp reliability 4](#_Toc468128114)

[Implementing udp reliability 5](#_Toc468128115)

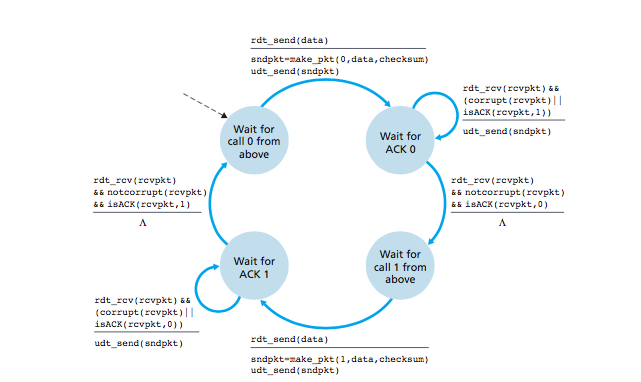
[Conclusion 6](#_Toc468128116)

[References 7](#_Toc468128117)

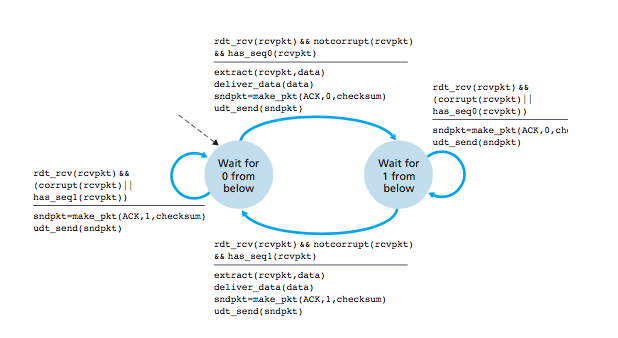
# Introduction

# About udp reliability

**Diagram Showing: rdt2.2 Sender Implementation**



**Diagram Showing: rdt2.2 Receiver Implementation**



# Implementing udp reliability

The project we worked on was concerned with creating an API. The functions necessary for server and client communication are encapsulated in two classes with its associated functions. The server and client files that are included was used for testing.

User Manual for the demo (Single Computer):

1. Download file
2. Extract zip file (Project3150.zip) to folder Project3150
3. Enter the folder Project3150
4. Shift + Right Click an empty space then click “Open command window here”, repeat two more times
5. First command window:
   1. Type “javac Rdt.java” and hit enter key
   2. Type “java Rdt” and hit the enter key
6. Second command window:
   1. Type “javac Server.java” and hit enter key
   2. Type “java Server” and hit enter key
7. Third command window:
   1. Type “javac Client.java” and hit enter key
   2. Type “java Client” and hit enter key
8. You will observe that Strings are transferred over the network from the server to the client. Should any errors be encountered (as seen in RDT 2.2) it will be shown on the related command window

User Manual for the demo (Two Computers):

It depends on the computer that acts as a client or server. For a client, perform all steps as shown above, **but for step 4 (Open only 2 windows) and exclude step 6.**

For a server, perform all steps as shown above, **but for step 4 (Open only 2 windows) and exclude step 7.**

## Testing udp implementation

Testing UDP requires at least one computer or two computers. The unreliability of a network determines the unreliability of data transfer over UDP. Testing UDP utilizing only one computer will not be as effective of a UDP test because the reliability of the network will be much greater compared to two computers at remote locations. Because of this, for testing purposes, a simulation of a network will be possible by altering the chances of a packet being corrupted or transferred in the wrong order. In the two examples to follow, it will be shown how UDP can be set up on a single computer (with a simulated unreliable network) or on two separate remote computers.

### single computer

On line 18 it can be seen that “localhost” was used. Localhost is used to refer to the computer that is currently being used. The address that was supposed to be placed on line 18 refers to the computer the client connects to, to receive data. Since a single computer will be used, the server will be the current computer also. The port in line 16 can be any port that is available. From line 21 to 24, an initial data transfer occurs for the server to establish which client it is connected to and its port being used.

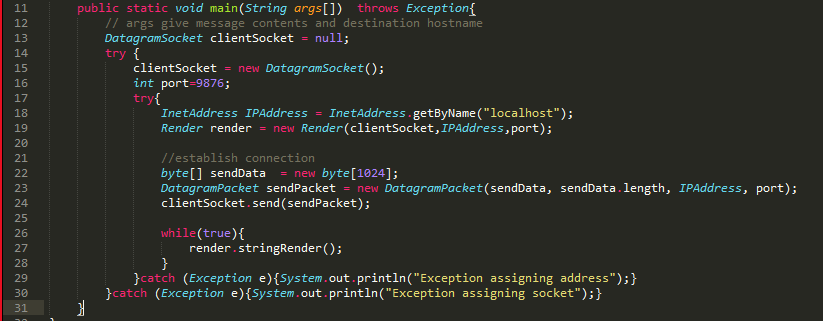


Figure 1: Client

Line 36 stores the IPAddress of the computer that it is connected to. Line 37 stores the port of the computer that it is also connected to.

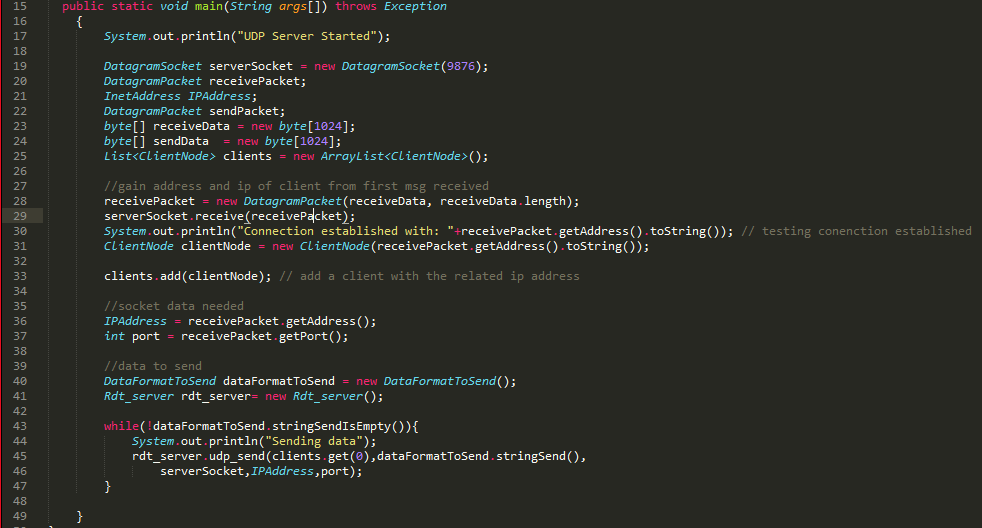


Figure 2: Server

### single computer (simulated network)

On a single computer the corruption of packets can be simulated by line 137-138. The concept that is utilized here is, once the packet to be sent is constructed, a RNG(Random Number Generator) determines which packet of the set of packets made by this package to alter. This altering is akin to the data corruption that can and will occur across a UDP network that does not implement a version of RDT.

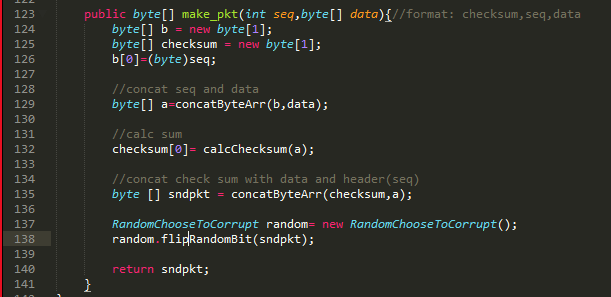


Figure 3: Packet altering

Naturally, if an unreliable network is to be simulated, the “degree” of unreliability should be alterable. On line 150, the int in the function call rand.nextInt(15) can be changed. The degree of unreliability can be viewed as a 1 in 15 chance of the packet being manually corrupted.

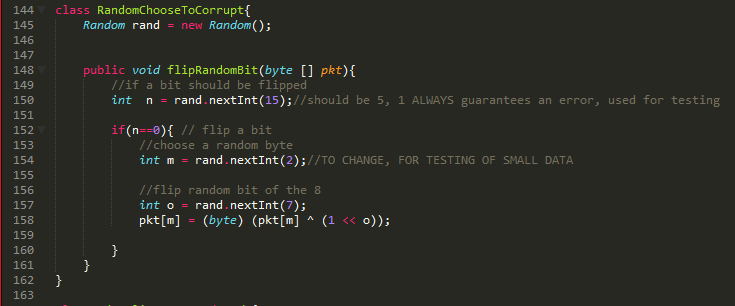


Figure 4: Altering degree of unreliability to be simulated

### Two remote computers

For two remote computers, the server side does not change from what is seen in Figure 2. On the other hand, in the client side, the only change from what was seen with “Single Computer”, is the entering of the ip address of the server to be connected to, into line 18 as shown below.

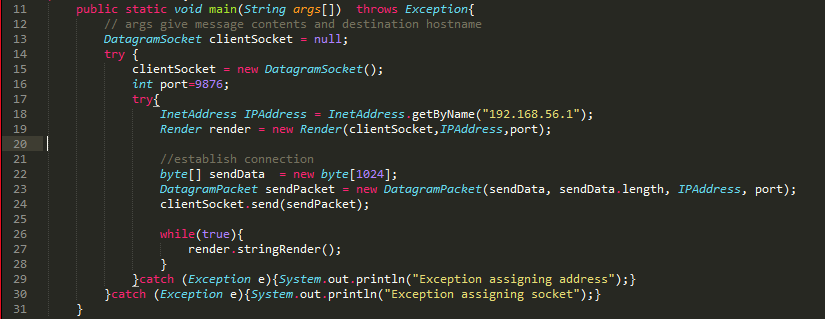


Figure 5: Client side for remote computers

# Conclusion

The team was able to introduce reliability (specifically RDT 2.2) into the application layer of UDP. From this project, we were able to view why UDP may be preferred over TCP despite lacking in established protocols for reliable data transfer. We were able to conclude that in the application layer, the degree of reliability can be coded for the exact needs of what need to be transferred, how fast and the distance etc.

# Bibliography

Ross, K. W. (2013). *Computer Networking: A Top-Down Approach (6th Edition).*