Nicky Randles

Network Distributed Systems

Assessment 1

Project aims

The main aim of this project is to create a program which uses the unreliable udp/ip protocol in a reliable way. We need to create a program that consists of two files, the sender and the receiver. The sender will establish a connection with the receiver and then the sender will send the file to the receiver via DatagramPackets. A socket must be initiated on both the sender and receiver files. The receiver will have to send a packet to the sender first, to initialise the connection and give the sender the correct ip address and port number to send the file. If the receiver is busy we should be able to block the sender until the receiver is not busy. The sender should also be able to put the packets in the correct sequence. Every 50 megabits the sender and receiver files should show two statistics. They should show the time the last 50 megabits took to transfer the packets to the receiver from the sender, it should also show the amount of data transferred so far. When the process has completely finished it should show the total amount of bits copied to the receiver, the total time and the average megabits per second.

Initial Design

Server.java: It creates a DatagramSocket object and assigns a port number to it. The port number in this case is 6060. It then greats a buffer of bytes. This is used originally to store the received data temporally. Then it creates a DatagramPacket object. It has two arguments, one is the buffer and the other is the buffer length. Since it is receiving data from the client we use the receive method to get the data. Once we have the data, we need to store the ip address and port number from the packet we are we receiving. We use an

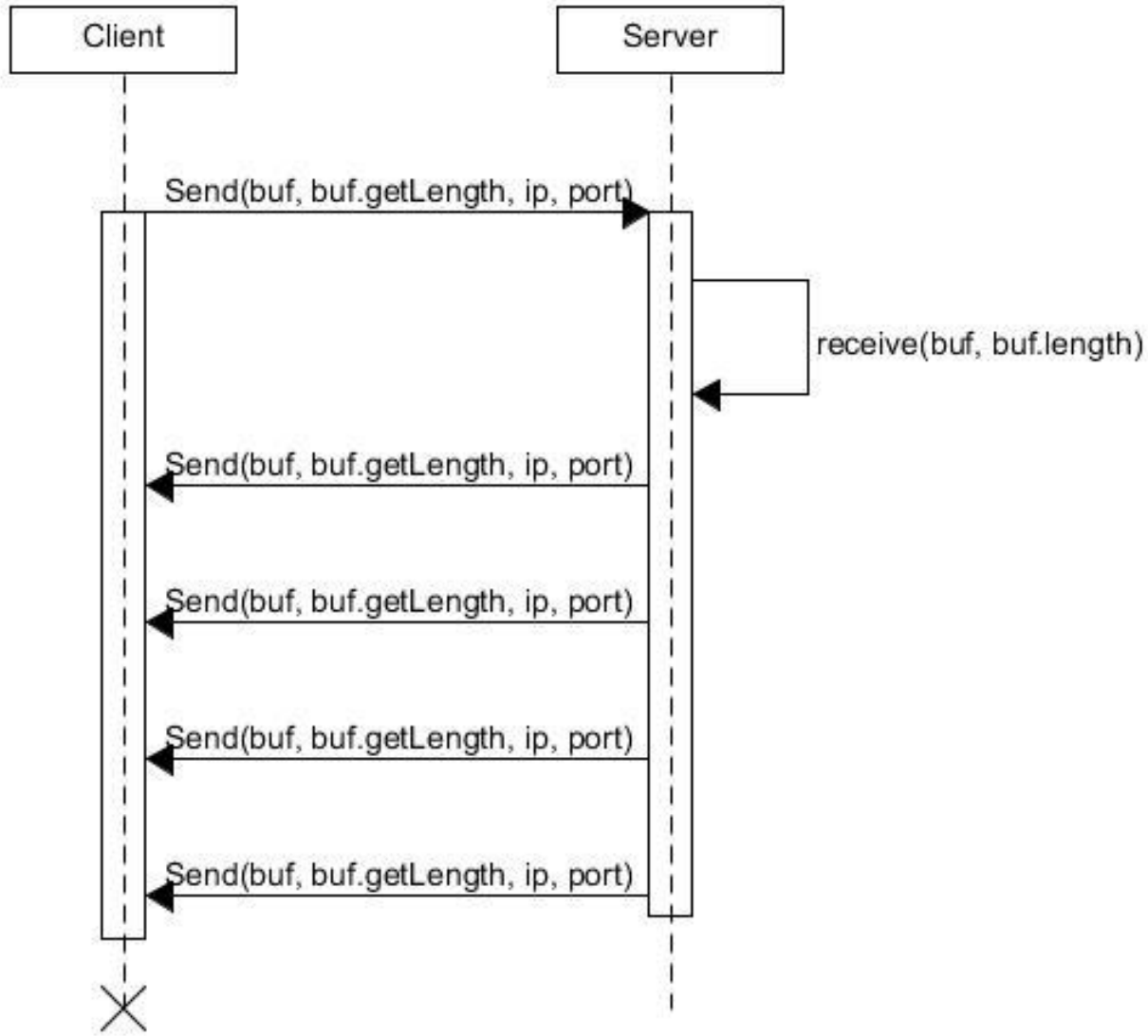
InetAddress object to store the ip address and an int to store the port. Now that we have this information we can send data to the client. We create a new packet, this time we are adding four arguments because we require more information when we are sending data. Before we send it we need to get the data we are going to send. Since we are copying a file, we must use the BufferedReader to input the file first. I created a GUI so that the user could select a file path with the FileChooser feature. Once the file path is selected, each line of the file is added to an ArrayList object. We then convert each array string into an array of bytes. We then send each array of byte over to the client in its own packet. We use the packet we initiated above. We need to use the send method this time instead of receive. When the ArrayList becomes empty the while loop sending each packet will break.

Client.java: First thing we do is create a DatagramSocket object. We then create a buffer of type byte to store data temporally. We then create a DatagramPacket object. This object is going to be used to establish the connection with the Server class. We need to assign it the ip address and port number of the client. We use the InetAddress object to get the ip address by name. Since we are working locally on one machine we will use the localhost. The DatagramPacket will need four arguments in total, the buffer, buffer.length, ip and port number. Since the whole purpose of this packet is to establish a connection and give the server the ip addres and port number of the client, we do not necessarily need to send a message. Once the server receives are packet and the connection is established we will start to receive packets from the server. We will hand this constant flow of packets using a while loop. We will display a GUI for the user giving them the ability to select a new file name for the file and the option to choose a location for the file. We use the BufferedWriter

object to write the data to the file. Once we are finished receiving packets we will break the while loop and close the connection.

Diagrams

Sequence diagram



Class diagram