Individual Extra work

Name: Pooja Prakashchand
Group ID: 12
SJSU ID: 010814258
Class ID: 53

Introduction

Existing features:

In our project, we designed a video streaming server which has the following characteristics:

- It allows the user play the video on the go, without waiting for the entire file to be downloaded.
- Capable of sending payload using RTP.
- Unicast the video to multiple users requesting for it.

Additional feature added:

An important feature of a server is to handle information of clients connected to it. I have added this feature to store some of the basic information of the client in a text file. This enables the server to keep a track of the number of clients and also information about the port, IP addresses and the time stamp. This feature could be used for analytics. Business uses of knowing client side details helps in the following way:

- Send marketing messages to clients based on its location
- Determine marketing campaigns for the particular for better revenue of the company.

Design

The below figure shows how the StaticVideoStreamingServer class interacts with the AnalysticsServer class.

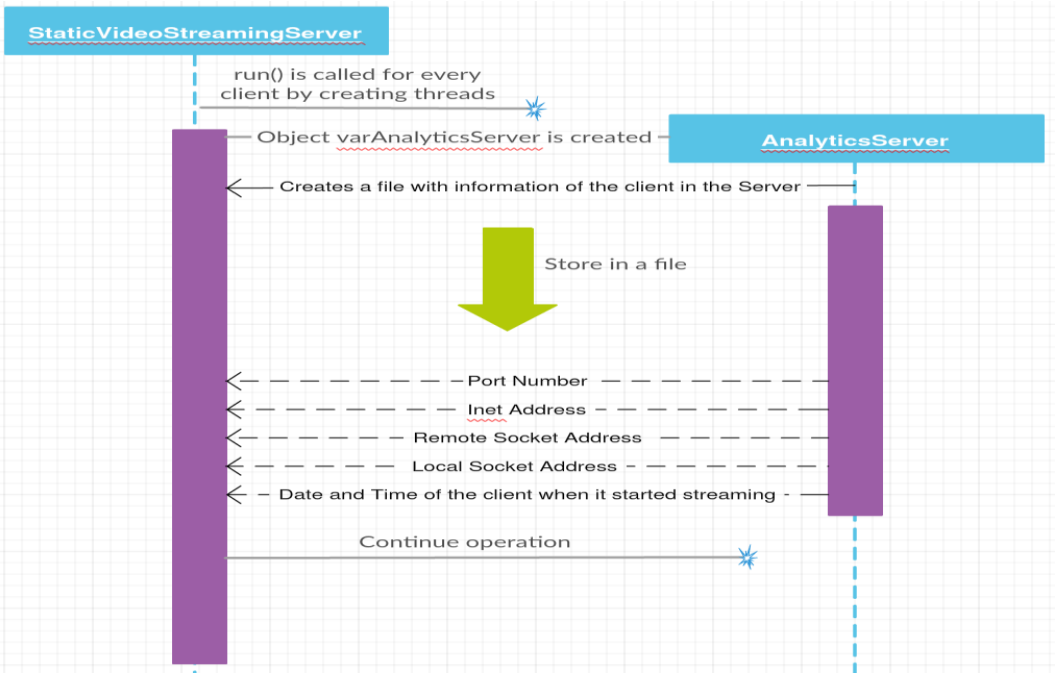

Figure 1: Interaction of StaticVideoStreamingServer class and AnalysticsServer class

The feature to store information about the clients is implemented in the AnalysticsServer class. This class is used to create a file that stores client side information. The following information is stored in the file – "Client details":

- Port Number
- Inet Address
- Remote socket address
- Local Socket address
- Date and time of the client when it started to stream the video

The result below shows the result of how the information about the clients is stored. The first two clients stream the video on 2016-05-12 at 2:26:05.805 and 02:26:16.772 respectively, whereas the next two clients stream the video at 3:45:55.668 and 3:46:07.973. The information about the port numbers and IP addresses is also stored.

Result

The After adding the code, a file is generated by the name – "ClientDetails". A sample snapshot of the file is as follows:

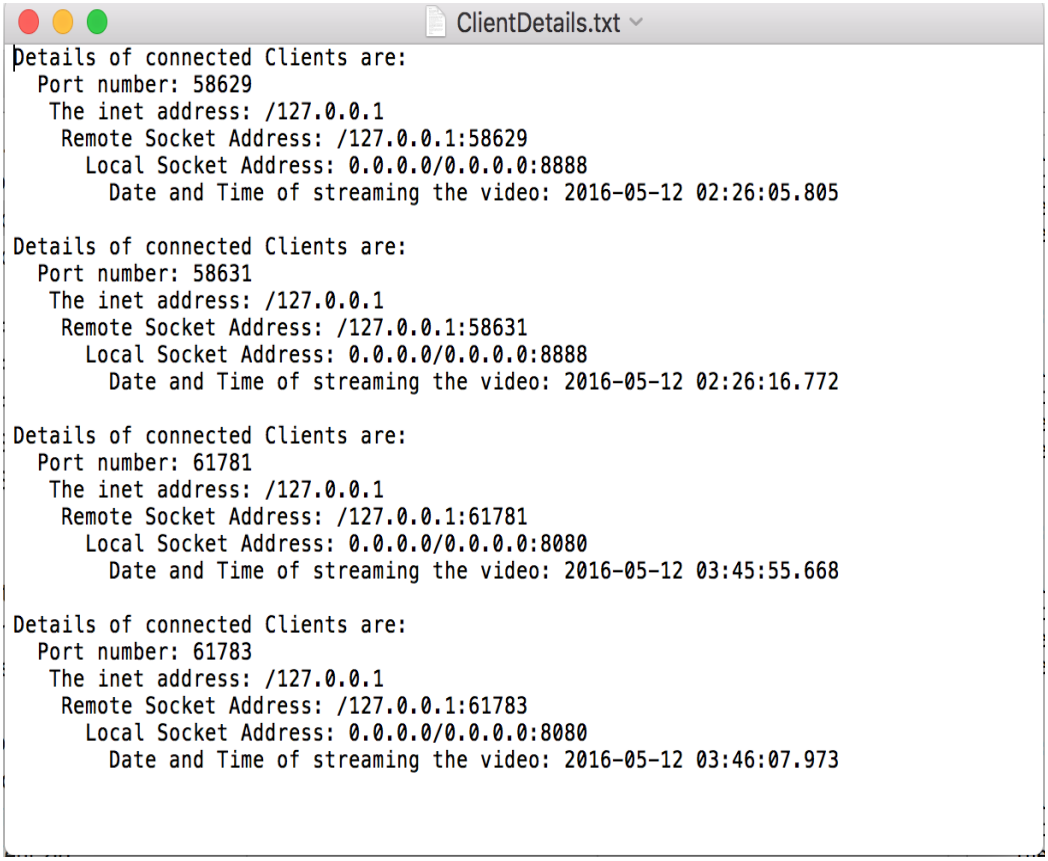

Figure 2: Format of how information about the clients are stored in a file

Code

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.Socket;
import java.util.Calendar;

public class AnalyticsServer {
     AnalyticsServer(Socket SocketOfClient){
     File file = new File("ClientDetails.txt");
   if(!file.exists()){
                    try {
                                        file.createNewFile();
                                } catch (IOException e) {

                                        e.printStackTrace();
                                }
                            }
       FileWriter fileWritter;
     try {
             fileWritter = new FileWriter(file.getName(),true);
             BufferedWriter bufferWritter = new BufferedWriter(fileWritter);
             bufferWritter.write("Details of connected Clients are:\n" + "  ");
             bufferWritter.write("Port number: "+SocketOfClient.getPort() + "\n " + "
");
             Calendar calendar = Calendar.getInstance();
             bufferWritter.write("The inet address:
"+SocketOfClient.getInetAddress() + "\n"+ "    " );
             bufferWritter.write("Remote Socket Address: "+
SocketOfClient.getRemoteSocketAddress() + "\n" + "      ");
             bufferWritter.write("Local Socket Address: "+
SocketOfClient.getLocalSocketAddress() + "\n"+ "       ");
         java.sql.Timestamp ClientTimeStamp = new
java.sql.Timestamp(calendar.getTime().getTime());
         bufferWritter.write("Date and Time of streaming the video: " +
ClientTimeStamp + "\n\n");
             bufferWritter.close();
     } catch (IOException e1) {

             e1.printStackTrace();
             }
          }
}
```