

CMPE 476 TERM PROJECT

INTRODUCTION

In this project we were asked to design and implement an application layer protocol to download mp3 files from server. In this application, clients can browse the list of the mp3 files and can download them in their computers. Registered user may upload files to the server also.

DESIGN and PROGRAM

I designed the server side program as one class that contains multiple functions. I used c language to implement this application and compile it with gcc on ubuntu 14.04 machine. Application works on TCP as requested. I am going to describe the program briefly.

The server is designed to accept more than one connections between clients simultaneously. Also client programs are designed to do multiple tasks per connection. So user does not need to start the program again and again. Clients need to start the program with two arguments: “gokberk and 5001” the first argument is binded to the local host IP address and the second one is the port number.

Server listens to connections at the port:5001 after the connection is established between client and server, at the client side user enters the number which is associated with the services of this application. Client sends this number to the server and corresponding service runs. For inquiring the largest MID and listing all the mp3 files, server reads the master file and response the client according to algorithm. I used 256 bytes of TPDU's.

Downloading mp3 files is a little bit complex than the other services. Service needs another number(MID) to start the download. If the client gives a number that is not available, the mp3 file with the largest ID will be downloaded to the client. Server checks the master file and sends the name and the size(how many bytes) of the requested file. Client reads this packet, opens the file than sends an acknowledgment to the server, than server sends the file. I used 8192 bytes to buffer the downloaded stream, to make it faster, I did not calculate the optimal TPDU size because TCP has its own flow control on it. After the download complete client may continue in the program.

ASSUMPTIONS

User should not interrupt the download service; also in the server side, the mp3 files which are listed in the master.txt must be located at the same directory of the server program. Also the master.txt must be in the “ID NAME PATH” format and the uploaded mp3 files information must be added to the master.txt file by the required format.

CONCLUSION

To conclude, downloading mp3 files, listing them and getting the largest value of MIDs is tested and working properly. Registration of to the system is implemented in server side, however it isn't tested.