|  |
| --- |
| Data Communications 2013 |
| Testing Document |
| Comm Audio |

|  |
| --- |
| Jesse Wright, Luke Tao  April 1 2013 |

|  |  |  |
| --- | --- | --- |
| Test | Expected Result | Result |
| Connecting to Sever for File Transfer | Server will accept connection and create a socket to deal with client request. | Refer to figure 1 |
| Multicast Playing Library | See the name of the current song being sent to the multicast group. | Refer to figure 2 |
| Client Getting a list of songs on the server. | Client should print a list of songs that the server sends over. | Refer to figure 3 |
| File Transfer Actually complete. | We should be able to see the new audio file on the client and play it in any media player. | Refer to figure 4 |
| Server waits for a client request on start up | Server will start program and listen for client requests for P2P Microphone session. | Refer to figure 5 |
| Client sends a P2P request and microphone session starts | Client will enter the server IP address, select the P2P button on the menu and the server will start displaying the session timer. | Refer to figure 6 & 7 |
| Client and server sends/receives audio data during the session | Client and server will send/receive audio data from microphone input during the session. | Refer to figure 8 |
| Server ends the microphone session | On the server side, the server ends the microphone session (via 'q' on standard input) and both client and server will get an ending session message. | Refer to figure 9 & 10 |

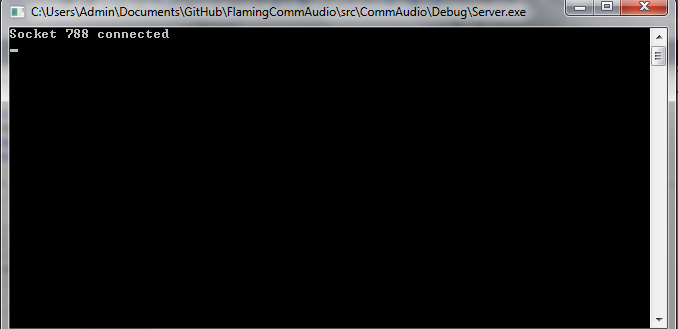


Figure 1.

After a client connects to the server we are notified that we someone has connected and present the socket number. Something similar to this happens when the client disconnects as well.

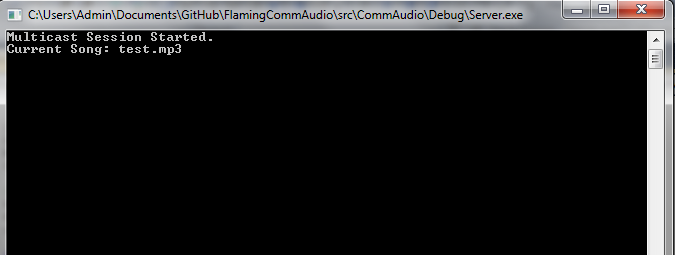


Figure 2.

When the multicast server starts we are prompted with a message saying the server has started. The current song will be displayed whenever the song changes.

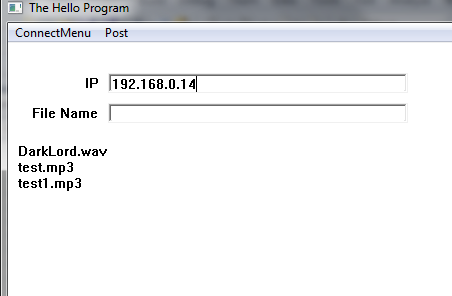


Figure 3.

Once the client connects to the server, the server responds with a list of files that the client can choose from to download.

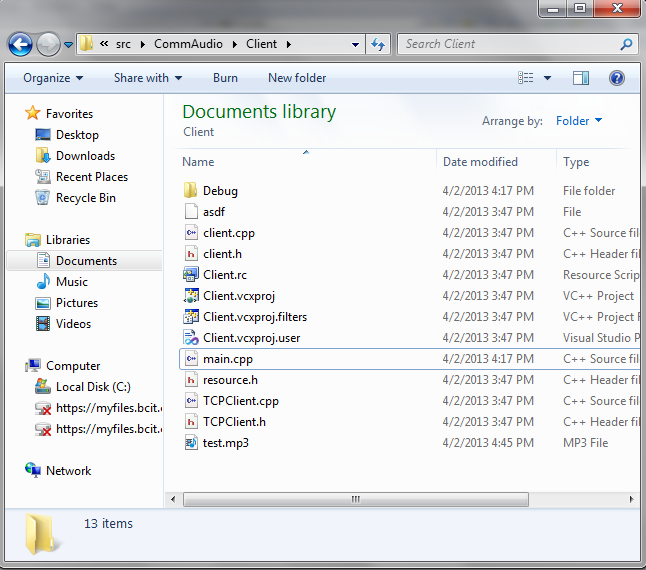


Figure 4.

Once the client has finished receiving data from the server we can check the current directory and see that the "Test.mp3" has been transferred.

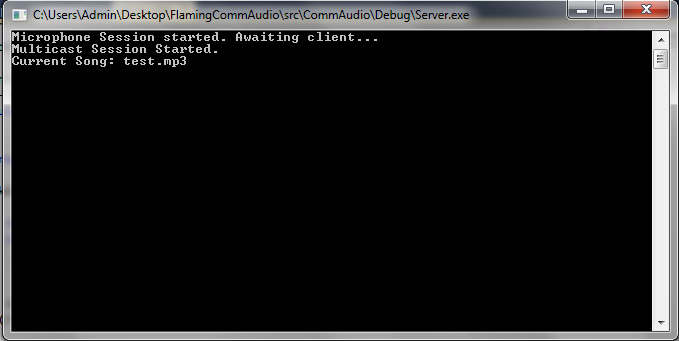


Figure 5.

Once the server program starts, the server will wait for a client request to start the UDP P2P Microphone session.

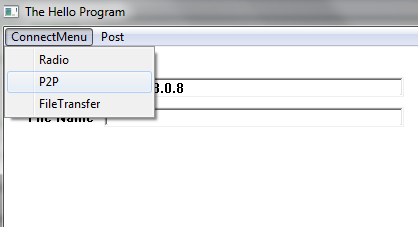


Figure 6 (Client).

Here the client will select the P2P Microphone session (via send a request).

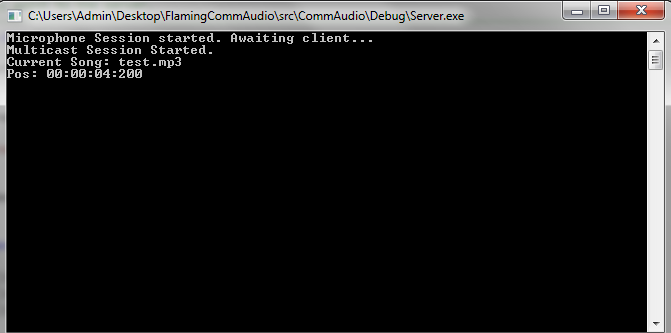


Figure 7 (Server).

Once the server gets the request, the session timer will appear and the microphone connection is established.

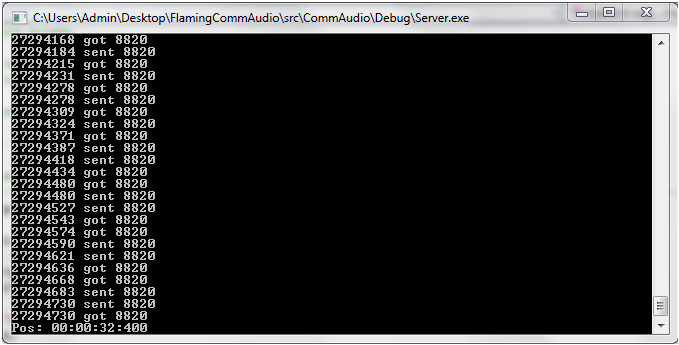


Figure 8 (Debug/older version of server).

Here, the server is sending/receiving microphone data to and from the client during the session. Note that the packet size is 8820 Bytes.

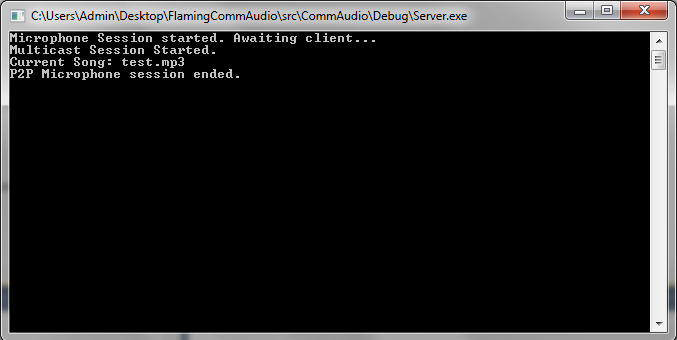


Figure 9 (Server).

If the server wants to end the session, the user can press 'q' from standard input and the session timer will be replaced by an ending message.

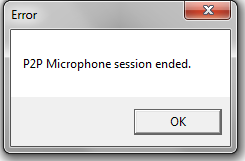


Figure 10 (Client).

Once the server disconnects, it will signal the client that the microphone session has now ended.