Report

on

Assignment 1: Client-Server System
Connected Systems and Devices
(DA614A)

Submitted

by

Annwesh Mukherjee
Lakshmidas Gurukkalkandy
Manaswini Kolluru
Prakriti Dhang

Department of Computer Science,
Faculty of Technology and Society
Malmö University
Malmö, Sweden

I. Intoduction

This assignment is to build a client-server system, which includes one server application running on the axis camera and one client application running on your desktop computer. The main functionality of this system is that the server should capture and send images that should be displayed by the client.

The client–server [1] [2] architectural model is a distributed system application which describes the communication between an application. Clients and servers communicates over a computer network on separate hardware. Both client and server may reside in the same system, and can communicate using the localhost address (127.0.0.1). Client and server can be in different system where same port number has to be used for communication. A server host can run one or more server programs which share their resources with clients. A client component interacts with individual servers. Client sends a requests to the server component, server accepts the request from the client and shares message that a client requested for.

In this system, the user will enter IP address, port number, select the image resolution, and enter number of frames. After getting connected with the server, image resolution, and number of frames will be sent to the server. After, receiving the image formats from client, server will capture the image and send back to the client. Client will display the image with specified resolutions and frames. Figure 1 and 2 shows the UML diagrams of our model.

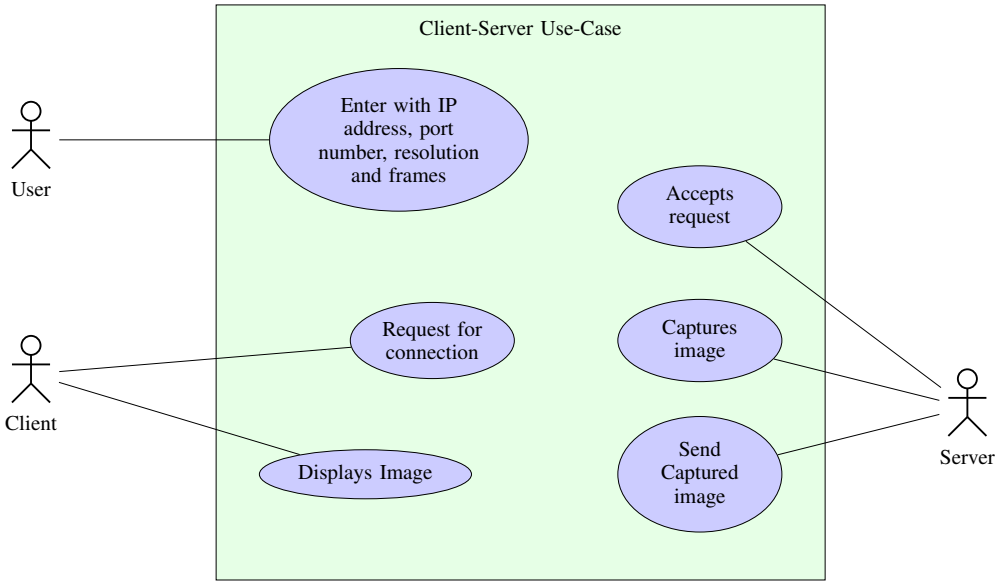

Fig. 1: Use case diagram of Client-Server System

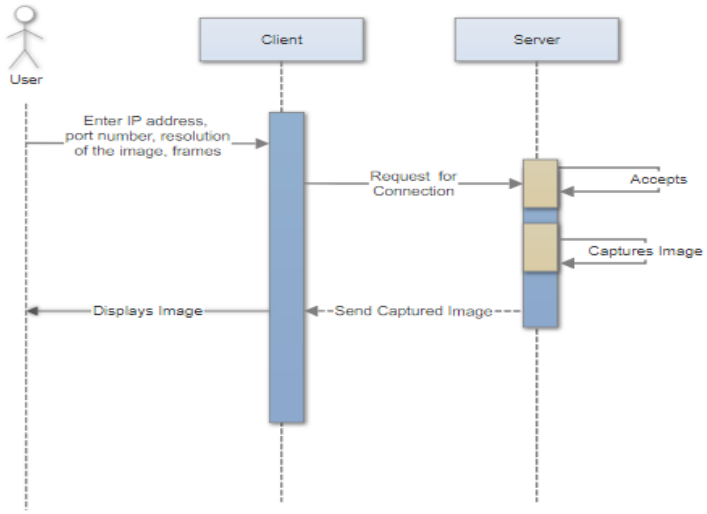

Fig. 2: Sequence diagram of Client-Server System

A. Requirements for the client-server application

1) Server:

- It manages connection with multiple clients.
- Capture images from an image stream.
- Captured image is sent back to the client.

2) Client:

- Establishes a connection with the server.
- Allows the user to specify the IP address and port of the server.
- Allows the user to specify the resolution and frequency of the captured image.
- Display the image.

B. Client and Server Model

1) Server Component: We developed our server component with C language. We used socket programming to connect with client component. We used fork() to handle multiple clients at the same time.

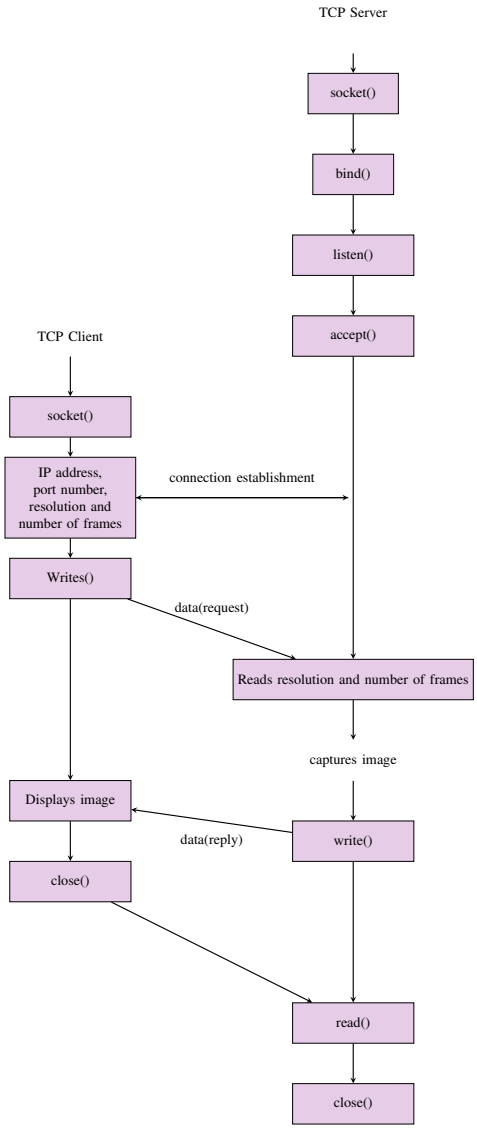

Fig. 3: Client-Server System [3]

2) Client Component: We used java platform to develop our client component. It consists of one frame in which user will enter IP address, port number, resolution of a image and a frames. There is a button for start streaming. When click on the button it displays a seperate window for image to display. We made alert message also when user enters wrong value. We set a default value.

C. Protocols Used

We used Transmission Control Protocol(TCP/IP). TCP provides connection oriented, full duplex stream delivery service using IP to transport messages between two process. TCP establishes a virtual path between the source and destination processes. Two procedures, connection establishment to start reliably and connection termination to terminate are used. For Connection establishement three way handshake is used.

D. Techniques Used

1) Server Component:

- Socket: Creates a socket of a given domain, type, protocol
- Bind: This function helps to bind to a address and a port.
- Listen: This function listens when a client tries to connect.
- Accept:This function is used to accept the client component that means establshing a connection.
- Read/Write: Send/Receive Stream based equivalents of read and write.
- Close:Close kernel data stuctures.

2) Client Component:

- Socket Creates the socket.
- Connect: This helps to get coonected with the server socket by giving IP address and port number.
- Write: Write to connection.
- Read: Read from the connection.
- Close: Close the connection.

II. Interesting and Challenging

A. Interesting

- Interesting part was that the server capturing the image and sending back to the client.
- Live streaming of an image.
- From this, assignment we learned about Socket programming and communication between client and server using different platform.
- Debuging with SYS.LOG in c programming.
- We made our user interface interactive as if user inserts incorrect input an error message will display.

B. Challenges

- Difficulty in displaying image. So to ensure that the server component is capturing image, we first created a file to store the image in that file. Later we used buffered image.

III. Outputs

User enters all the required fileds and on clicking the button ”Start Streaming” it displays in a separate window.

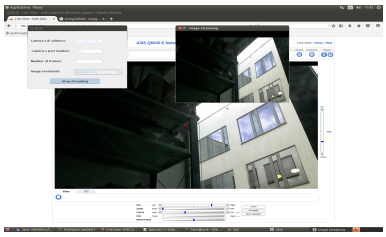

Image1: From 192.168.20.252

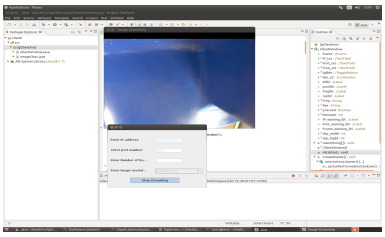

Image2: From 192.168.20.247

References

[1] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, ”Distributed System: Concepts and Design, Ed. 5th, Pearson Education Limited
[2] https://en.wikipedia.org/wiki/Client
[3] https://www.tutorialspoint.com/unix_sockets/client_server_model.htm