EEP-703 Computer Network Lab

Assignment4-Client-Server Socket Programming using UDP

Harshit Kumar Gupta
2013EET2369

Computer Technology
Department Of Electrical Engineering
IIT DELHI

March 12, 2014

Contents

Chapter 1

PROBLEM STATEMENT

Basics of Network Programming: An introduction to the basics of writing client-server stubs in JAVA in UNIX environment.

1. Problem Statement (a) Using Implement this: a client sends an AWK script to the server. The AWK script on execution generates a message. Server will (first) execute the script and (then) send the output message back to client. To avoid spamming, server will accept the AWK script only after authentication with a secret-code. This secret-code is preknown to all the true clients.

2. Problem Statement (b) Extend the above scenario to serve multiple clients simultaneously. The server should be able to keep track of up to 5 simultaneous users. However, server should accept the AWK script as a file only from the first client who makes a request. Server should make the AWK file into an executable, execute it locally and send the generated message back to the client.

3. Problem Statement (c) Extend the above scenario so that the server broadcasts the output message to all the available clients. Also provide the functionality to broadcast a custom message entered by ADMIN1. All the clients (except the 1st one) should stay only in listen state.

Chapter 2

ABSTRACT

The Intention of the JAVA Code is to make use of Socket Programming to create a Server and a Socket Client on a UDP approach where we are supposed to send AWK script file from the Client side and send it to the Server. The Server then in return send backs the entire record and displays it on the Client terminal.

Chapter 3

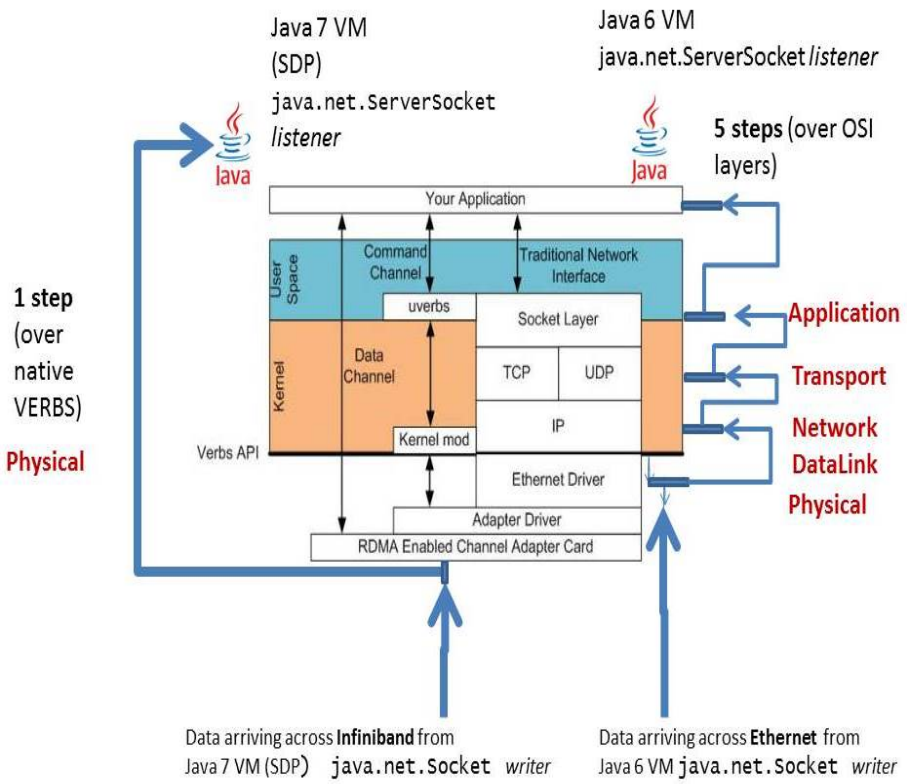
INTRODUCTION

In all computer networks, one of the computers acts as a server (for applications, data, services) to client computers. In this assignment, we will learn how to develop the JAVAcode which is used to program this functionality on the client and on the server.

JAVA is one of the most widely used programming languages of all time, and JAVA compilers are available for the majority of available computer architectures and operating systems. , UDP is a simple transport-layer protocol. The application writes a message to a UDP socket, which is then encapsulated in a UDP datagram, which is further encapsulated in an IP datagram, which is sent to the destination.

There is no guarantee that a UDP will reach the destination, that the order of the datagrams will be preserved across the network or that datagrams arrive only once. The problem of UDP is its lack of reliability: if a datagram reaches its final destination but the checksum detects an error, or if the datagram is dropped in the network, it is not automatically retransmitted.

Each UDP datagram is characterized by a length. The length of a datagram is passed to the receiving application along with the data. No connection is established between the client and the server and, for this reason, we say that UDP provides a connection-less service.


Java 7 VM
(SDP)
java.net.ServerSocket
listener
Java 6 VM
java.net.ServerSocket listener
5 steps (over OSI layers)
1 step
(over
native
VERBS)
Physical
Java
Java
Your Application
User Space
Command Channel
Traditional Network Interface
uverbs
Socket Layer
Kernel
Data Channel
TCP
UDP
IP
Kernel mod
Verbs API
Ethernet Driver
Adapter Driver
RDMA Enabled Channel Adapter Card
Application
Transport
Network
DataLink
Physical
Data arriving across Infiniband from Java 7 VM (SDP) java.net.Socket writer
Data arriving across Ethernet from Java 6 VM java.net.Socket writer

Chapter 4

SPECIFICATIONS AND ASSUMPTIONS

Specifications

1. Client must provide authentication to send the awkscript to the server.
2. buffer size used is 1024 as it is sufficient.
3. The input file awkscript.awk is stored in the same path.
4. Socket Programming has been implemented in C language.

Assumptions

1. There are as such no multiple entries for the same client.
2. Error handling has been done accordingly for non existent records or as such.

Chapter 5

LOGIC USED/METHODOLOGY

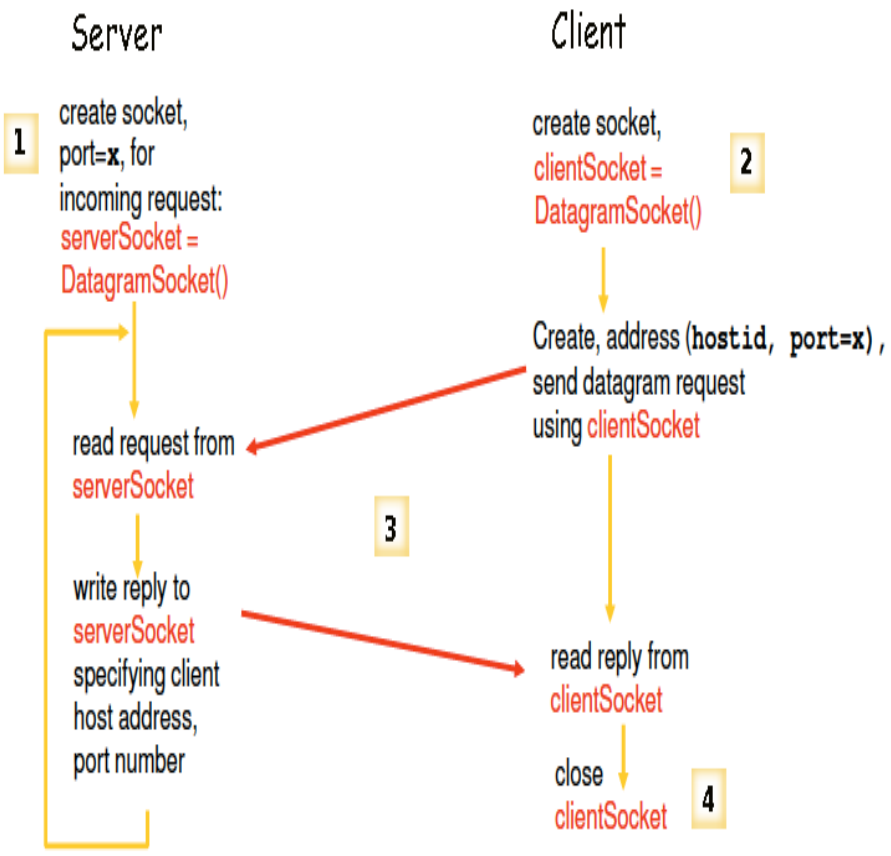
The methodology that is used for developing this project work is defined below:

1. First of all a Server is To be Created and the port no is defined.
2. The Server Code also connects to the awkscript File.
3. A buffer is defined so as to save the Input file.
4. Now the Socket function is called and structure Intialised.
5. Now the input Search string say name is sent to the server.
6. Server executes awkscript and returns the corresponding Record value.
7. If the User enters some Invalid value , the error message is displayed.

Chapter 6

FLOWCHART

Flowchart



Chapter 7

Execution Directives

In the terminal of LINUX system the following commands are executed in order to create a network topology and its analysis.

1. Simply Compile the file writing java *.java
2. Start server using java UDPServer
3. Now Simply run client using java UDPClient
4. close ——close the comnnection
5. execawk awkfile inputfile ——execute awkfile and gives output
6. execawk awkfile inputfile -b ——execute awkfile and broadcasts output
7. wall message ————broadcast message to all users
8. wall -f filename ————- broadcast content of file

Chapter 8

RESULTS AND CONCLUSIONS

We find that for the given code seems to work perfectly fine and server accepts and executes the awkscript perfectly well.