TCP

```
        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPA
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.le
        clientSocket.receive(receivePacket);
        String modifiedSentence = new String(receivePacket.getData());
        System.out.println("FROM UDP SERVER:" + modifiedSentence);
        clientSocket.close();
   }
}

//-----------------------------------TCPServer.java
//home/tovantran/Ctest/TCPServer.java --> 2015-03-14 by ./.tv  owner: tovantran

//https://systembash.com/a-simple-java-tcp-server-and-tcp-client/
//package mythread.tcpserver;

import java.io.*;
import java.net.*;

class TCPServer
{
   public static void main(String argv[]) throws Exception
      {
         String clientSentence;
         String capitalizedSentence;
         ServerSocket welcomeSocket = new ServerSocket(6789);          SOCKET
                                    // ServerSocket only port
         while(true)                                                    *
         {
            Socket connectionSocket = welcomeSocket.accept(); // block call
            BufferedReader inFromClient =
               new BufferedReader(new InputStreamReader(connectionSocket.getInputSt
            DataOutputStream outToClient = new DataOutputStream(connectionSocket.ge
            clientSentence = inFromClient.readLine();                 //READ
            System.out.println("Received: " + clientSentence);
            capitalizedSentence = clientSentence.toUpperCase() + '\n';
            outToClient.writeBytes(capitalizedSentence);               // WRITE
         }
      }
}
//-----------------------------------TCPClient.java
//home/tovantran/Ctest/TCPClient.java --> 2015-03-14 by ./.tv  owner: tovantran

//https://systembash.com/a-simple-java-tcp-server-and-tcp-client/
//package mythread.tcpclient;

import java.io.*;
import java.net.*;

class TCPClient
{
 public static void main(String argv[]) throws Exception
 {
  String sentence;
  String modifiedSentence;
  BufferedReader inFromUser = new BufferedReader( new InputStreamReader(System.in))
  Socket clientSocket = new Socket("localhost", 6789); //Socket(client) host and po
  DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream(
  BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSock
  sentence = inFromUser.readLine();          //READ
  outToServer.writeBytes(sentence + '\n');   // WRITE
  modifiedSentence = inFromServer.readLine();
  System.out.println("FROM TCP SERVER: " + modifiedSentence);
  clientSocket.close();
 }
}
```

send

Rx

Socket API → binding

source IP
src Port
Destination IP
Destination port

— SOCK-STREAM
— SOCK-DATAGRAM

int socket(int domain, int type, int protocol)

— INET
— UNIX

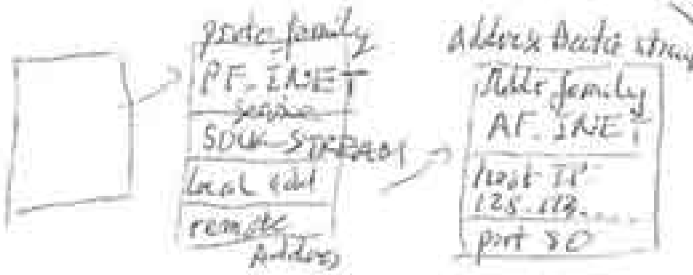
int bind (int socket, struct sockaddr * address, int len)
It binds the socket to the specified address.

Socket Descriptors

IN OUT) OBJ

— Descriptor each process
Socket descriptors are shared by threads

3 data structures — socket descriptor table
— Socket data structure
— Address data structure



SOCRET
SOCKET STREAM
OUT
IN) CREATE OBJ