

Lecture (05)

Network Programming (II)

Dr. Ahmed ElShafee

1

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

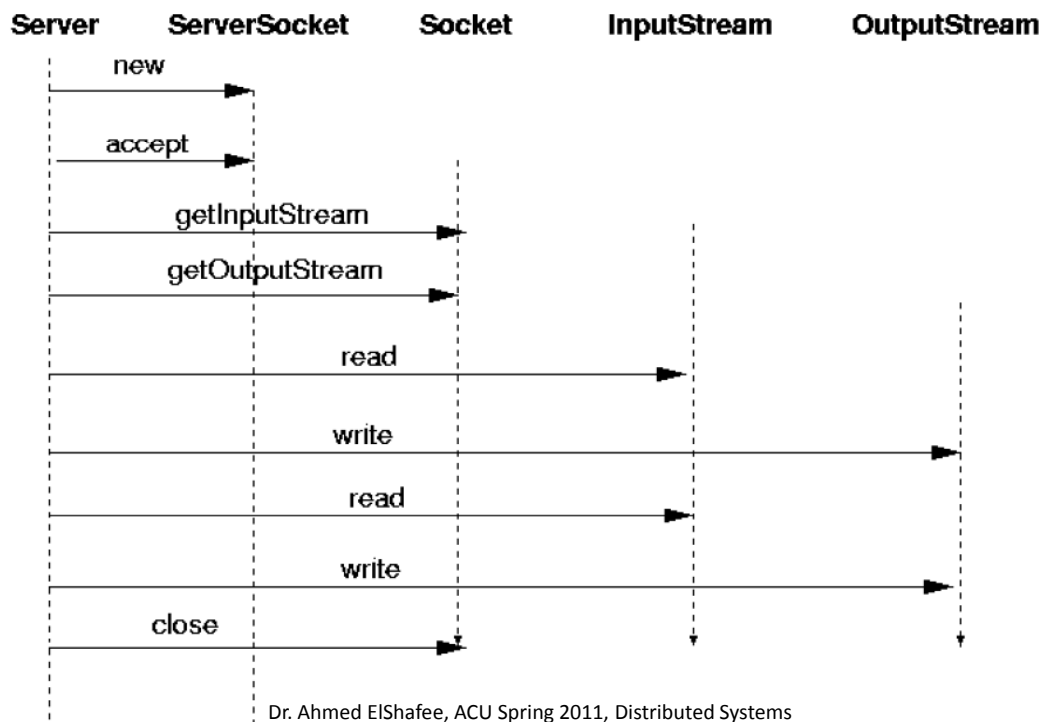
TCP/IP client/server



2

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

TCP Server



3

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

Echoserver.java (1)

```
ServerSocket s = null;
try {s = new ServerSocket(MYECHOPORT);}

while (true)
    {
        Socket incoming = null;
        try {incoming = s.accept();}
        try {incoming.setSoTimeout(10000); //10 seconds}
        try {handleSocket(incoming);}
        try {incoming.close();}
    }
}
```

4

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

```
public static void handleSocket(Socket incoming) throws IOException
{
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        incoming.getInputStream()));
    PrintStream out = new PrintStream(incoming.getOutputStream());
    out.println(">>Server sent: Hello. Enter BYE to exit,..");
    System.out.println(">> Host is now connected,..");
    boolean done = false;
    while ( ! done)
    {
        String str = reader.readLine();
        out.println(str);
        System.out.println(">> Host sent: "+str);
        if (str.equals("BYE"))
            done = true;
    }
}
```

5

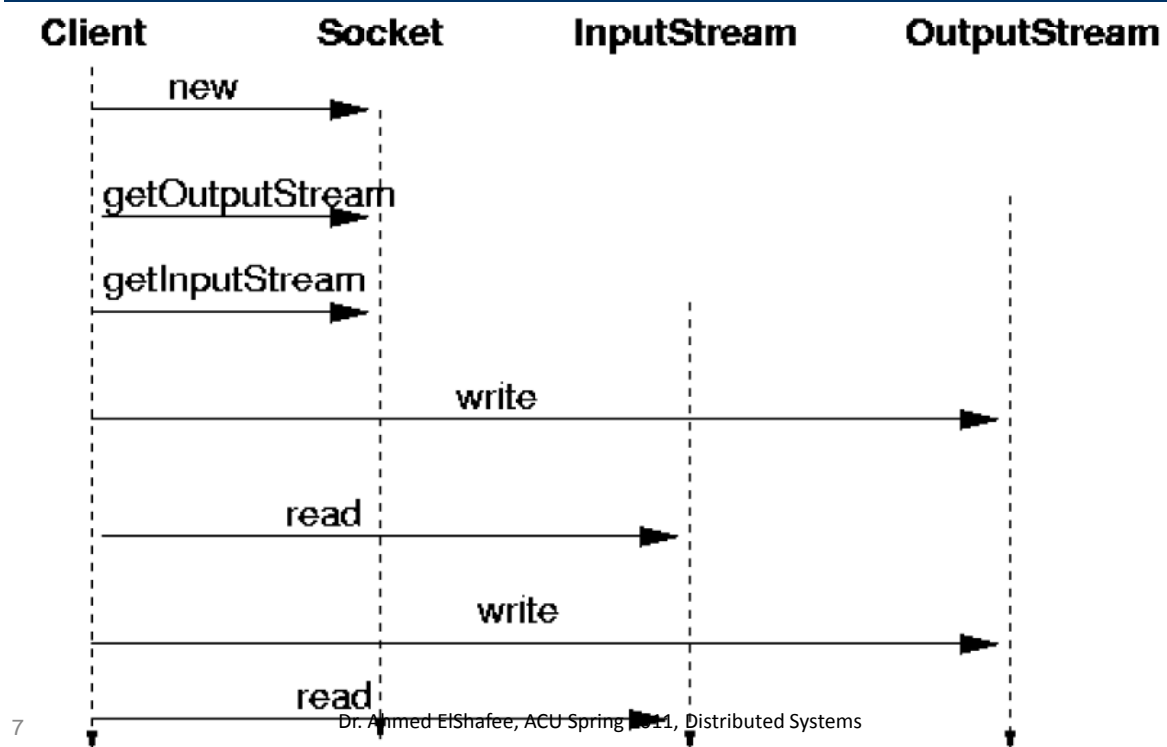
Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

```
System.out.println(">> Host is now dis-connected,..");
incoming.close();
}
}
```

6

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

TCP client



Echoclient.java (1)

```
Socket echoSocket = null;
PrintWriter out = null;
BufferedReader in = null;
BufferedReader stdin = null;
try
{
    echoSocket = new Socket(args[0], 8189);
    out = new PrintWriter(echoSocket.getOutputStream(), true);
    in = new BufferedReader(new
InputStreamReader(echoSocket.getInputStream()));
    System.out.println (in.readLine());
}
```

try

```
{  
    stdIn = new BufferedReader(new InputStreamReader(System.in));  
    String userInput;  
    while (true)  
        {  
            System.out.print(">");  
            userInput = stdIn.readLine();  
  
            out.println(userInput);  
            System.out.println(">>>Server sent: " + in.readLine());  
            if (userInput.equals("BYE")) break;  
        }  
}
```

9

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

UDP/IP client/server



10

Dr. Ahmed ElShafee, ACU Spring 2011, Distributed Systems

Udpserver.java (1)

```
DatagramSocket serverSocket = new DatagramSocket(9876);  
byte[] receiveData = new byte[1024];  
byte[] sendData = new byte[1024];  
while(true)  
    {  
        DatagramPacket receivePacket =  
        new DatagramPacket(receiveData , receiveData.length);  
        serverSocket.receive (receivePacket);  
        String sentence = new String(receivePacket.getData());  
        sentence=sentence.trim();  
        InetAddress IPAddress = receivePacket.getAddress();  
        int port = receivePacket.getPort();
```

Udpserver.java (2)

```
System.out.println(">>" + IPAddress.toString() + " sent: " + sentence );  
        sentence="Echo:" + sentence;  
        sendData = sentence.getBytes();  
        DatagramPacket sendPacket=new  
        DatagramPacket(sendData,sendData.length,IPAddress,port);  
        serverSocket.send (sendPacket);  
    }
```

Udpclient.java (1)

```
    BufferedReader inFromUser = new BufferedReader(new  
InputStreamReader(System.in));  
    DatagramSocket clientSocket = new DatagramSocket();  
    InetAddress IPAddress = InetAddress.getByName (args[0]);  
    byte[] sendData = new byte[1024];  
    byte[] receiveData = new byte[1024];  
    while(true)  
        {  
            System.out.print(">>");  
            String sentence = inFromUser.readLine();  
            sendData = sentence.getBytes();  
            DatagramPacket sendPacket= new DatagramPacket(sendData,  
sendData.length , IPAddress, 9876);  
            clientSocket.send(sendPacket);
```

Udpclient.java (2)

```
DatagramPacket receivePacket =  
    new DatagramPacket(receiveData, receiveData.length);  
    clientSocket.receive(receivePacket);  
    String modifiedSentence = new String (receivePacket. getData());  
    modifiedSentence=modifiedSentence.trim();  
    System.out.println(">>Sever sent: " + modifiedSentence );  
    if(sentence.equals("BYE"))break;  
  
    }  
    clientSocket.close();
```



Thanks,
See you next Week, isA