TCP and UDP Socket Programming

Tom Kelliher, CS 325
Feb. 22, 2008

1 Administrivia

Announcements

Assignment II due Monday.

Assignment

Read 3.1–3.3.

From Last Time

Wireshark labs.

Outline

1. Introduction.

2. TCP socket example.

3. UDP socket example.
2 Introduction

Basics:

1. Hostname and port are used to specify transport endpoints.

2. Socket — the communication object.

3. TCP properties: reliable, connection-oriented, byte-stream, connection established before application-level protocols exchange information, two-way communication.

4. UDP properties: unreliable, packet-switched, packet data, no connection overhead, application-level protocols exchange information immediately, two-way communication.

A socket connection is a 4-tuple — (HostA, PortA, HostB, PortB) — uniquely defining the connection.
3 TCP Socket Example

TCP client/server communication flow:

- **Server (Running on hostid)**
  - Create socket port=x, for incoming request: `welcomeSocket = ServerSocket()`
  - Wait for incoming connection request: `connectionSocket = welcomeSocket.accept()`
  - Read request from `connectionSocket`
  - Write reply to `connectionSocket`
  - Close `connectionSocket`

- **Client**
  - Create socket connected to `hostid, port=x`
  - Send request using `clientSocket`
  - Read reply from `clientSocket`
  - Close `clientSocket`
TCPClient.java communication model:

TCPServer.java communication model (UDP figure):
TCP server code:

```java
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);

        while(true)
        {
            Socket connectionSocket = welcomeSocket.accept();

            BufferedReader inFromClient =
                new BufferedReader(
                    new InputStreamReader(
                        connectionSocket.getInputStream()));

            DataOutputStream outToClient =
                new DataOutputStream(
                    connectionSocket.getOutputStream());

            clientSentence = inFromClient.readLine();

            capitalizedSentence = clientSentence.toUpperCase() + '\n';

            outToClient.writeBytes(capitalizedSentence);
        }
    }
}
TCP client code:

```java
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("hostname", 6789);

        DataOutputStream outToServer =
            new DataOutputStream(
                clientSocket.getOutputStream());

        BufferedReader inFromServer =
            new BufferedReader(
                new InputStreamReader(
                    clientSocket.getInputStream()));

        sentence = inFromUser.readLine();
        outToServer.writeBytes(sentence + '\n');

        modifiedSentence = inFromServer.readLine();

        System.out.println("FROM SERVER: " + modifiedSentence);

        clientSocket.close();
    }
}
```
4 UDP Socket Example

UDP client/server communication flow:
UDPClient.java communication model:

UDPServer.java communication model:
UDP server code:

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

class UDPServer
{
    public static void main(String args[]) throws Exception
    {
        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());
            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();

            String capitalizedSentence = sentence.toUpperCase();
            sendData = capitalizedSentence.getBytes();

            DatagramPacket sendPacket =
                new DatagramPacket(sendData, sendData.length,
                    IPAddress, port);

            serverSocket.send(sendPacket);
        }
    }
}
UDP client code:

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

class UDPClient
{
    public static void main(String args[]) throws Exception
    {
        BufferedReader inFromUser =
            new BufferedReader(
                new InputStreamReader(System.in));

        DatagramSocket clientSocket = new DatagramSocket();

        InetAddress IPAddress = InetAddress.getByName("hostname");

        byte[] sendData = new byte[1024];

        byte[] receiveData = new byte[1024];

        String sentence = inFromUser.readLine();

        sendData = sentence.getBytes();

        DatagramPacket sendPacket =
            new DatagramPacket(sendData, sendData.length,
                                IPAddress, 9876);

        clientSocket.send(sendPacket);

        DatagramPacket receivePacket =
            new DatagramPacket(receiveData, receiveData.length);

        clientSocket.receive(receivePacket);

        String modifiedSentence = new String(receivePacket.getData());

        System.out.println("FROM SERVER :" + modifiedSentence);

        clientSocket.close();
    }
}
```