Mansoura University
Faculty of Computers and Information Sciences
Department of Computer Science
First Semester- 2020-2021

[CS412P] Distributed Systems

Grade : FORTH GRADE

By : Zeinab Awad

WHAT ARE SOCKETS ?

- Sockets are a strategy that is used for communication in client server-based systems.
- Socket is defined as an endpoint for communication
- A pair of processes or systems communicating over a network employ a pair of sockets –one for each process.
- A socket is identified by an IP address and a port number.

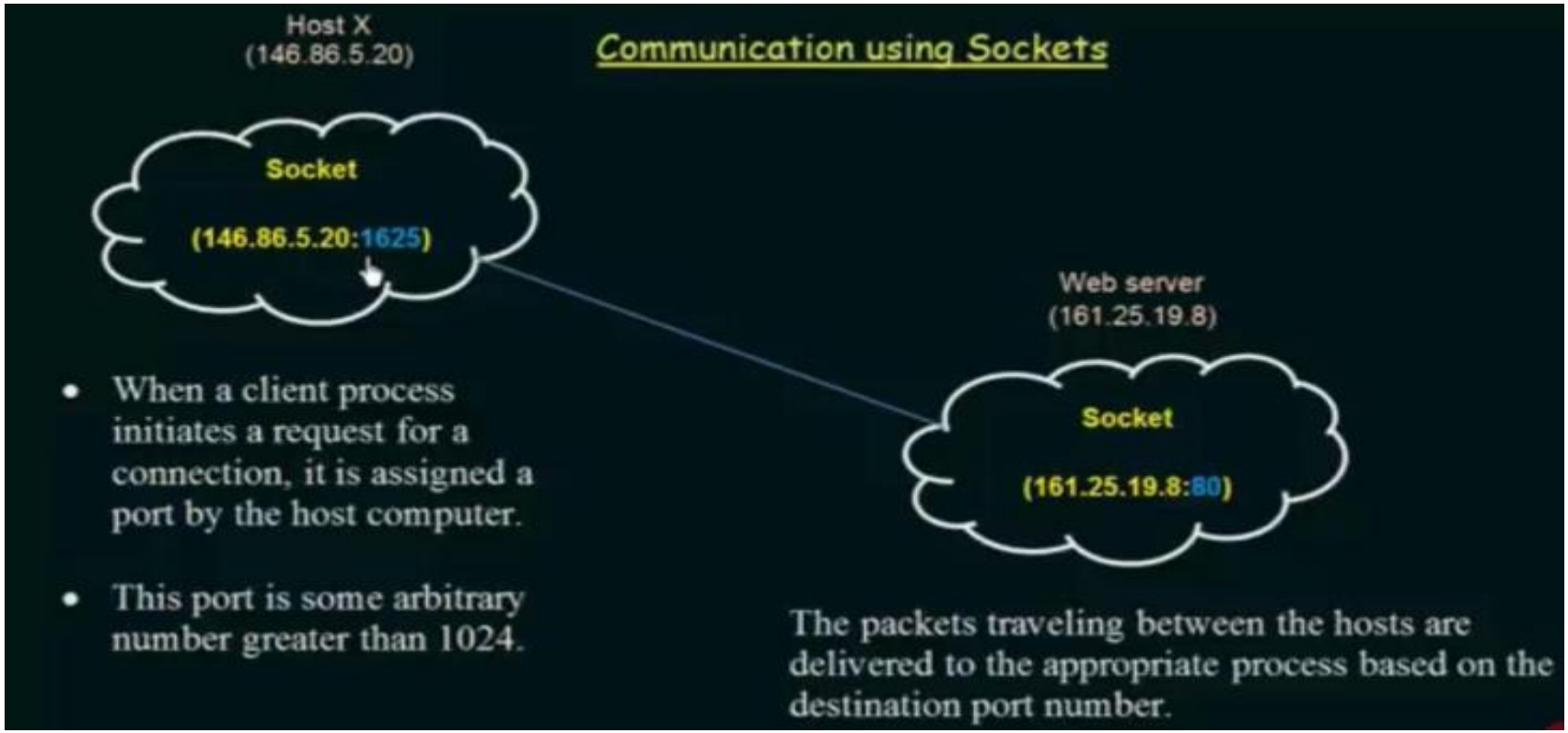
WHAT ARE SOCKETS ?

- The server waits for incoming client requests by listening to a specified port .
- Once the request is received the server accepts a connection from the client Socket to complete the connection
- Once the connection is established the client can ask for any information(request) from the server and the server present the information to the client (Reply).

WHAT ARE SOCKETS?

- Servers implementing specific services (such as telnet for remote log in, ftp for file transfer, and http)
- They listen to well-known ports(a telnet listen to port no 23,an ftp server to port 21,and a web or http server listen to port 80).
- Port numbers less than 1024 are kept for Standard server services ,so always define your server port to be greater than 1024.

Communication Using Sockets



Difference between UDP and TCP

- UDP : user datagram protocol
- TCP : transmission control protocol
- Both of them work on the transport layer which allow multiple application to share the same network connection simultaneously.

UDP VS. TCP

TCP	UDP
Reliable(all segments received-no error occurs-all segments are out together in the same order.	Unreliable (data segments may be lost and errors may occur).
Connection-based(ensure that connection is established before data transmission.by three way-handshake followed by 4 step procedure to make sure that every bit is delivered before close the connection. acknowledgment delivery	Connectionless (no handshake to establish the connection no procedures is followed before close the connection).
Stream oriented	Message oriented
Congestion control	No congestion control
Slow and less efficient(due to communication over head).	Light weight,faster and more efficient

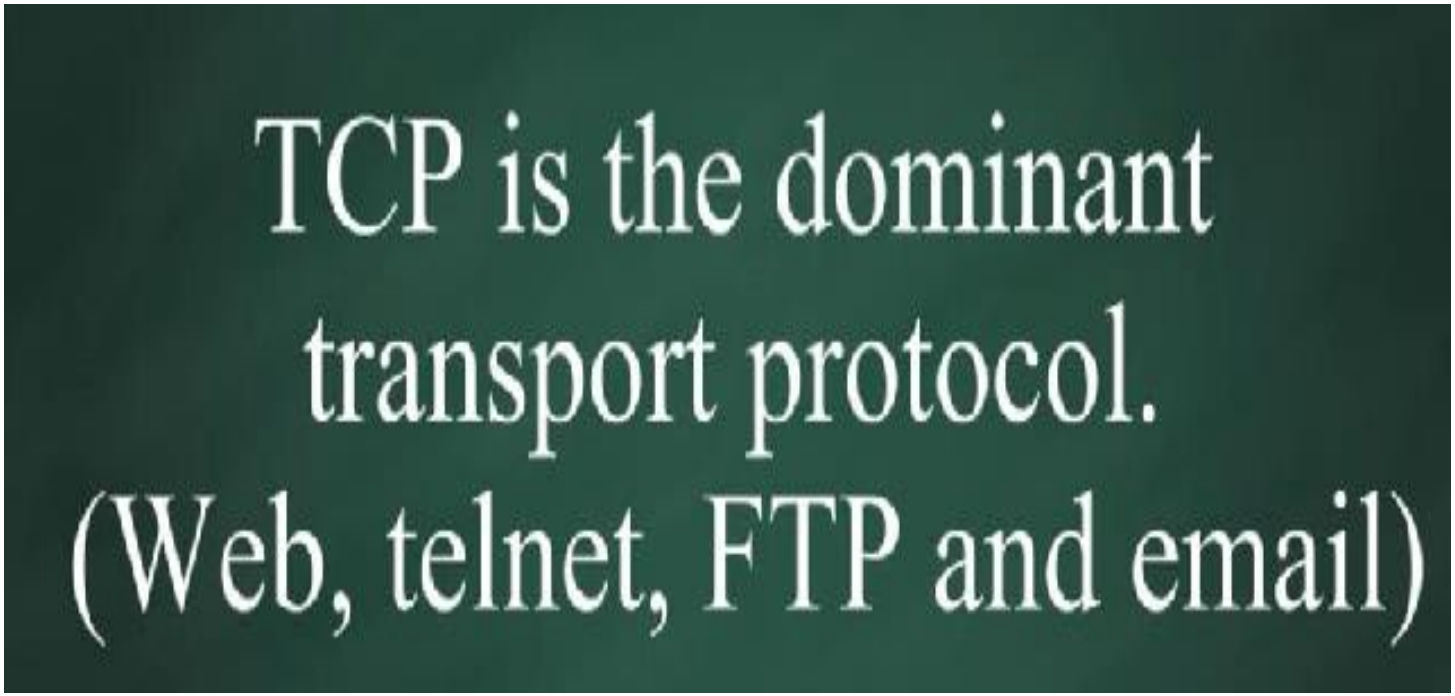
TCP VS. UDP

TCP	UDP
Larger packet sizes Bigger Header:20 bytes	Small packet sizes Header:8 bytes
Delivery acknowledgements, retransmission if not, in order delivery	No error recovery (corrupted segments discarded) No compensation for lost packets Packets can arrive out of order
Error *detection (check sum) is mandat*ory for ipv4 and 6 packets.	Error detection is mandatory for Ipv6 packets only.

TCP VS. UDP


UDP Usage
1) live streaming audio or video;
2)DNS queries, DHCP or VoIP.
3) A few applications use UDP

TCP VS. UDP


TCP is the dominant
transport protocol.
(Web, telnet, FTP and email)

SOCKET PROGRAMMING

-Server:

=> Create a socket with the socket()
=> Bind the socket to an address using the bind().
=> Listen for connections with the listen().
=> Accept a connection with the accept().
=> Send and receive data, use the read() and write() system calls.

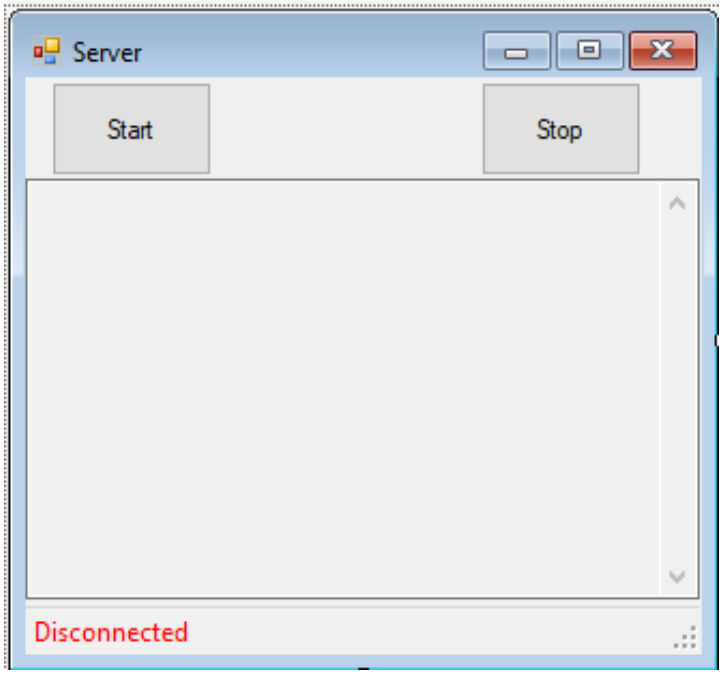
-Client:

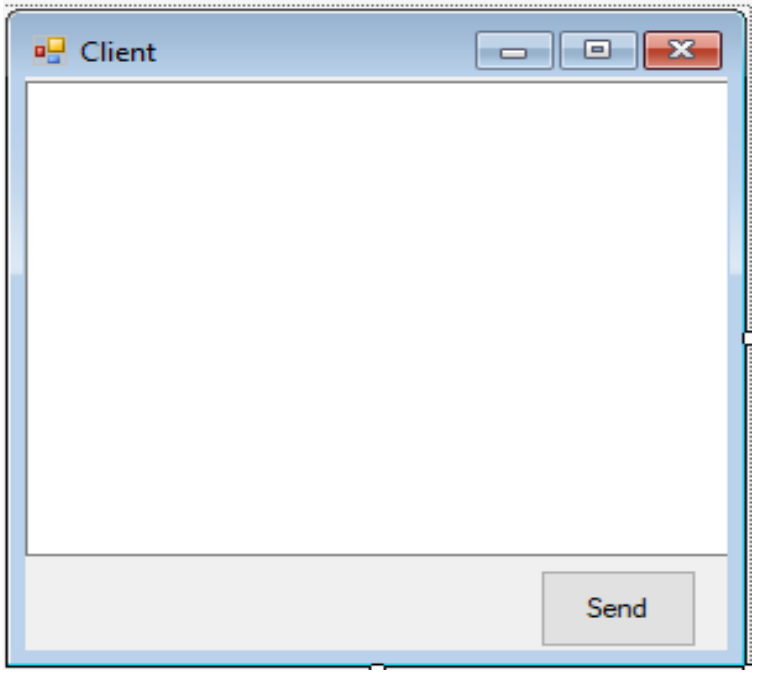
=> Create a socket with the socket()
=> Connect the socket to the address of the server using the connect() system call.
=> Send and receive data, use the read() and write() system calls.

-Major steps.
-Server coding.
-Client coding.
-Running.

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

- Step 1:use the visual studio .NET to create a new windows form project.
- Design the following forms for the client and server respectively.





EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

- Step 2 : Add the following code to the client form:

```
namespace Client
{
    public partial class Form1 : Form
    {
        Socket clientSocket;
        IPEndPoint serveraddress;

        public Form1()
 {
   InitializeComponent();
   clientSocket = new Socket(AddressFamily.InterNetworkSocketType.Dgram,
ProtocolType.Udp);
   serveraddress = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 6767)
}
```

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

```
private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Trim() == null)
            {
                MessageBox.Show("Empty");
                textBox1.Focus();
                return;
            }
            byte[] buffer = Encoding.ASCII.GetBytes(textBox1.Text.Trim());
            clientSocket.SendTo(buffer,serveraddress);
            textBox1.Text = "";
        }
        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

.Add the following code to the server form:

```
namespace Ds_sec1
{
    public partial class Form1 : Form
    {
        Socket serverSocket;
        IPEndPoint serveraddress;
        Thread thread;
        void startServer()
        {
serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
serveraddress=new IPEndPoint(IPAddress.Parse("127.0.0.1"), 6767);
serverSocket.Bind(serveraddress);
```

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

```
thread = new Thread(new ThreadStart(Listening));
            thread.Start();

        }
        bool isRunning = false;
        void Listening()
        {
            while (isRunning)
            {
      byte[] buffer = new byte[4096];
      EndPoint clientaddress = new IPEndPoint(IPAddress.Any,0);
      int bytescount = serverSocket.ReceiveFrom(buffer, ref clientaddress);
      string msg = Encoding.ASCII.GetString(buffer,0,bytescount);
```

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

```
logTxt.Invoke(new Action(delegate {
logTxt.AppendText(clientaddress.ToString() + ": " + msg + "\n");
                    }));
            }
        }
        void stopServer()
        {
            isRunning = false;
            thread.Abort();
            serverSocket.Close();
        }
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            button1.Enabled = false;
            button2.Enabled = true;
            statusLbl.Text = "Connected";
            statusLbl.ForeColor = Color.Green;
            isRunning = true;
            startServer();
        }
```

EXAMPLE: SOCKET PROGRAMMING IN UDP MODE

```
 private void button2_Click(object sender, EventArgs e)
        {
            button1.Enabled = true;
            button2.Enabled = false;
            statusLbl.Text = "Disconnected";
            statusLbl.ForeColor = Color.Red;
        }

    }
}
```

INTRODUCTION TO SOCKET PROGRAMMING

Thanks