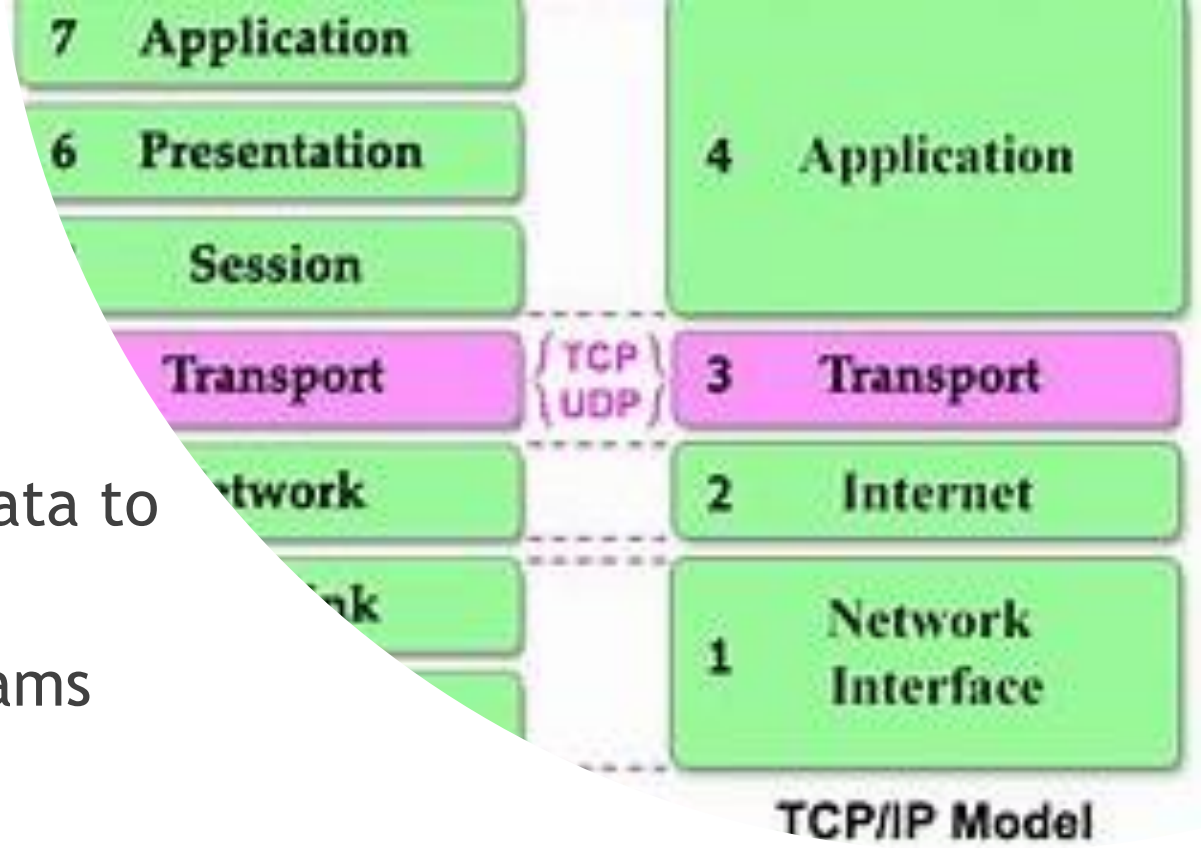
Transport Protocols

UDP : User Datagram Protocol

TCP : Transmission Control Protocol

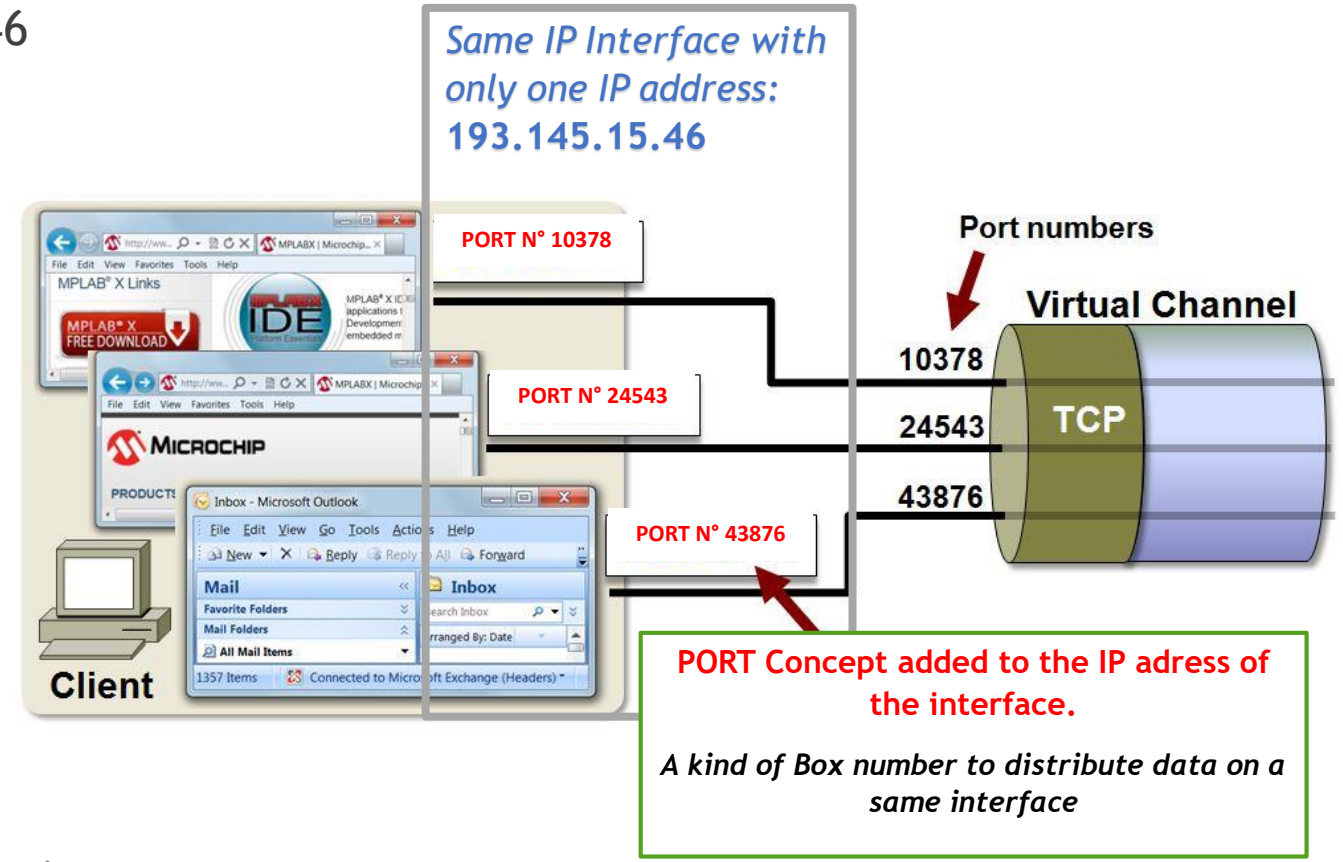
Why IP is not enough for data communications ?

- Internet Protocol only allows to send data to one network interface to another
- But fortunately several software programs may used a same IP interface (ex. web browser, mail browser, etc.).
- Two main transport protocol add then the concept of PORT
- For a same IP interface, PORT are kind of Boxes number to distribute data

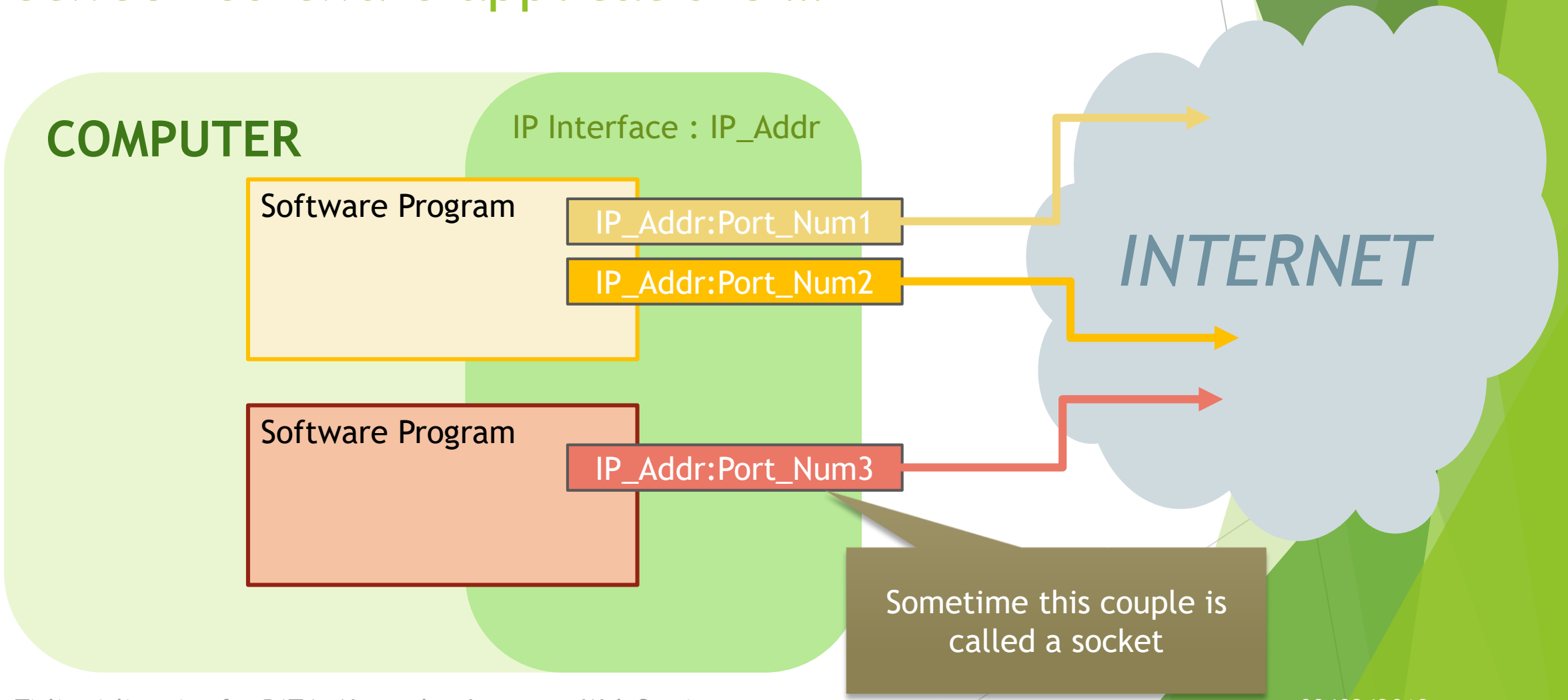


PORT is the key concept for Transport Protocols

- Here for example, the client has one IP interface with the IP Address : 193.145.15.46
- Three software application communicate with remote programs through the IP interface.
- They used then three PORTs, one for each
- A endpoint for a software application using IP is then, IP_ADDRESS:PORT_NUM
- Ex :
  - 193.145.15.46:10378
  - 193.145.15.46:24543
  - 193.145.15.46:43876

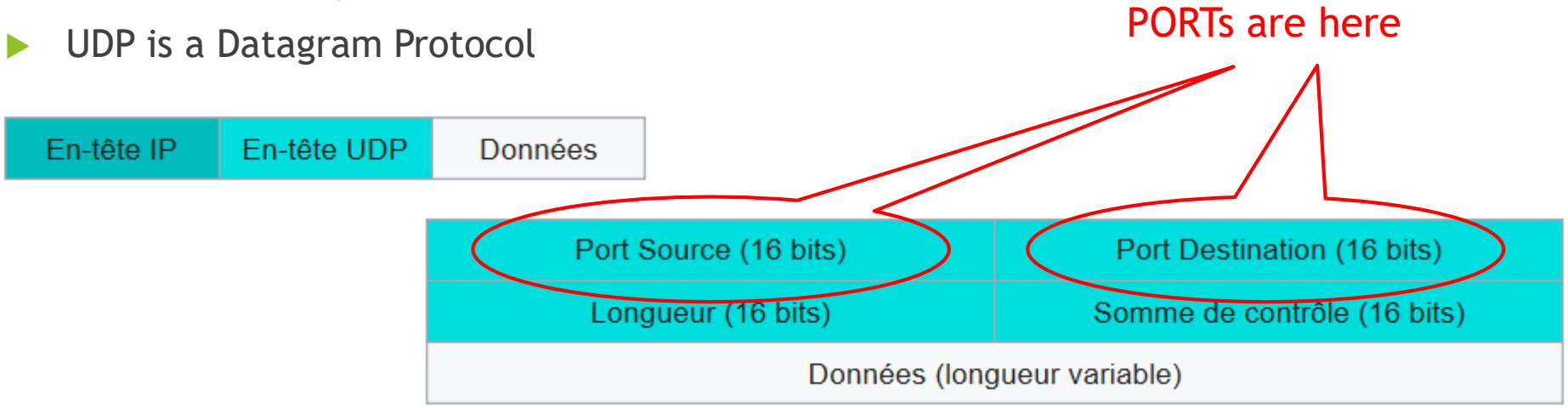


Simple Model for IP network Communication between software applications …



First Transport Protocol : UDP
The simplest !

- UDP: User Datagram Protocol
- UDP is a Datagram Protocol



- Limitation : One Packet / One Data, called Datagram
- Limitation : Transmitted Data are limited in size
- Limitation : Because of the changing route between routers and uncertainties to reach the destination, UDP doesn't garantee that the message is arrived

Low popularity of UDP and then TCP …

- UDP/IP is not very used
- Applications that use UDP/IP : TFTP etc.
- Others like, HTTP (Web), SMTP (mail), IMAP (mail) and 99 % of Internet applications use TCP/IP

- Why ?
- Because of UDP limitations …

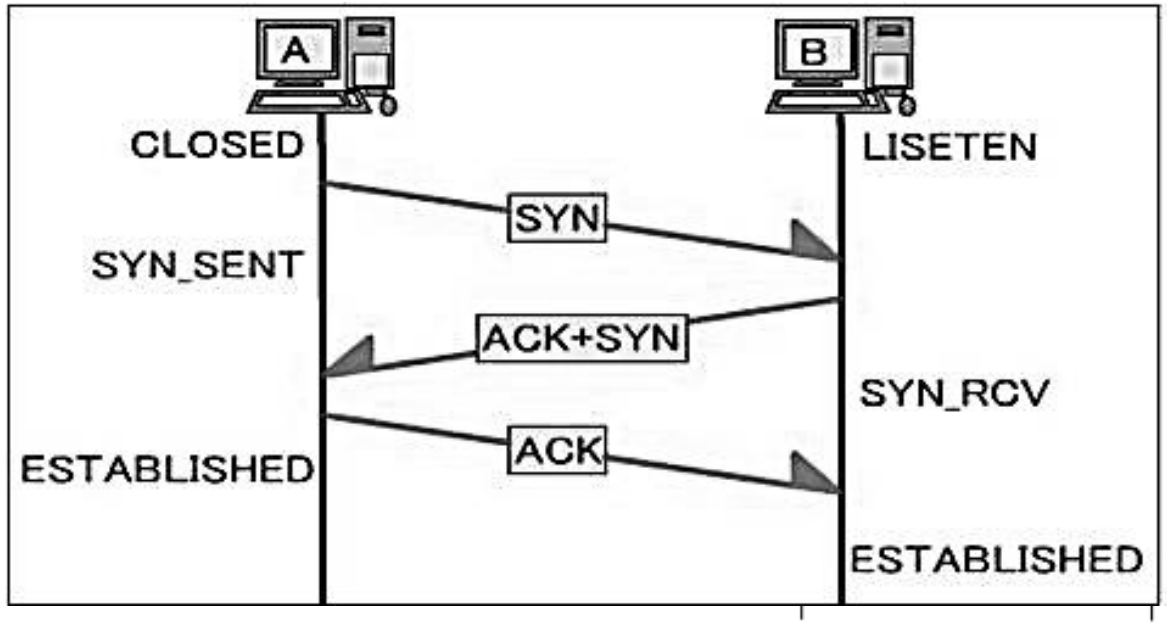
- TCP avoid them !

TCP : Transmission Control Protocol

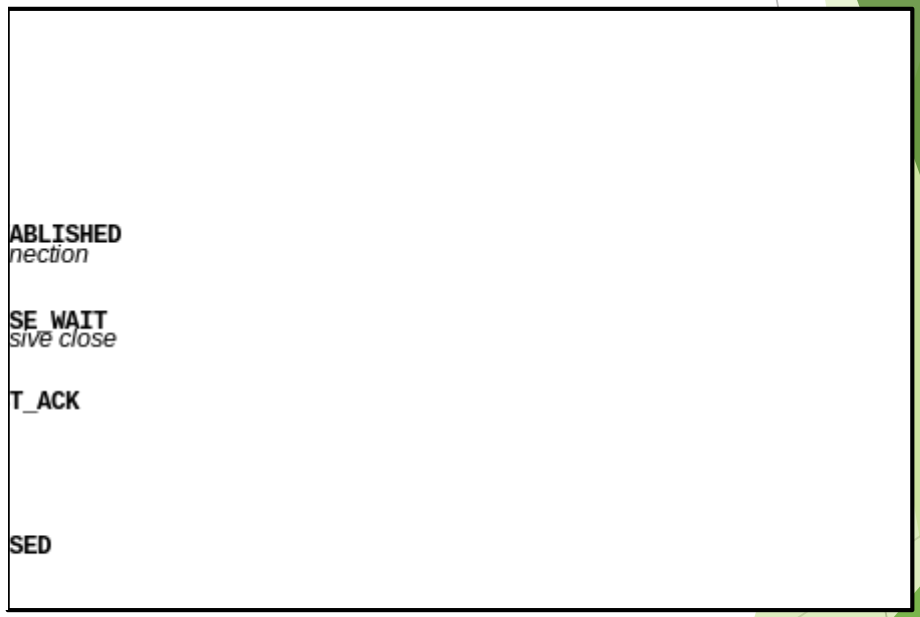
- TCP properties :
  - One Data, called Message / several Packets
  - So, Transmitted Data are NOT limited in size
  - The arrival of messages at destination is verified (we also say « guaranteed »)
- But for that, a « connection » between both software programs that communicate must be established before data communications and ended after.
- TCP/IP is a Connected Protocol

Connection / Disconnection for TCP

Connection (3-way Handshaking)



Disconnection



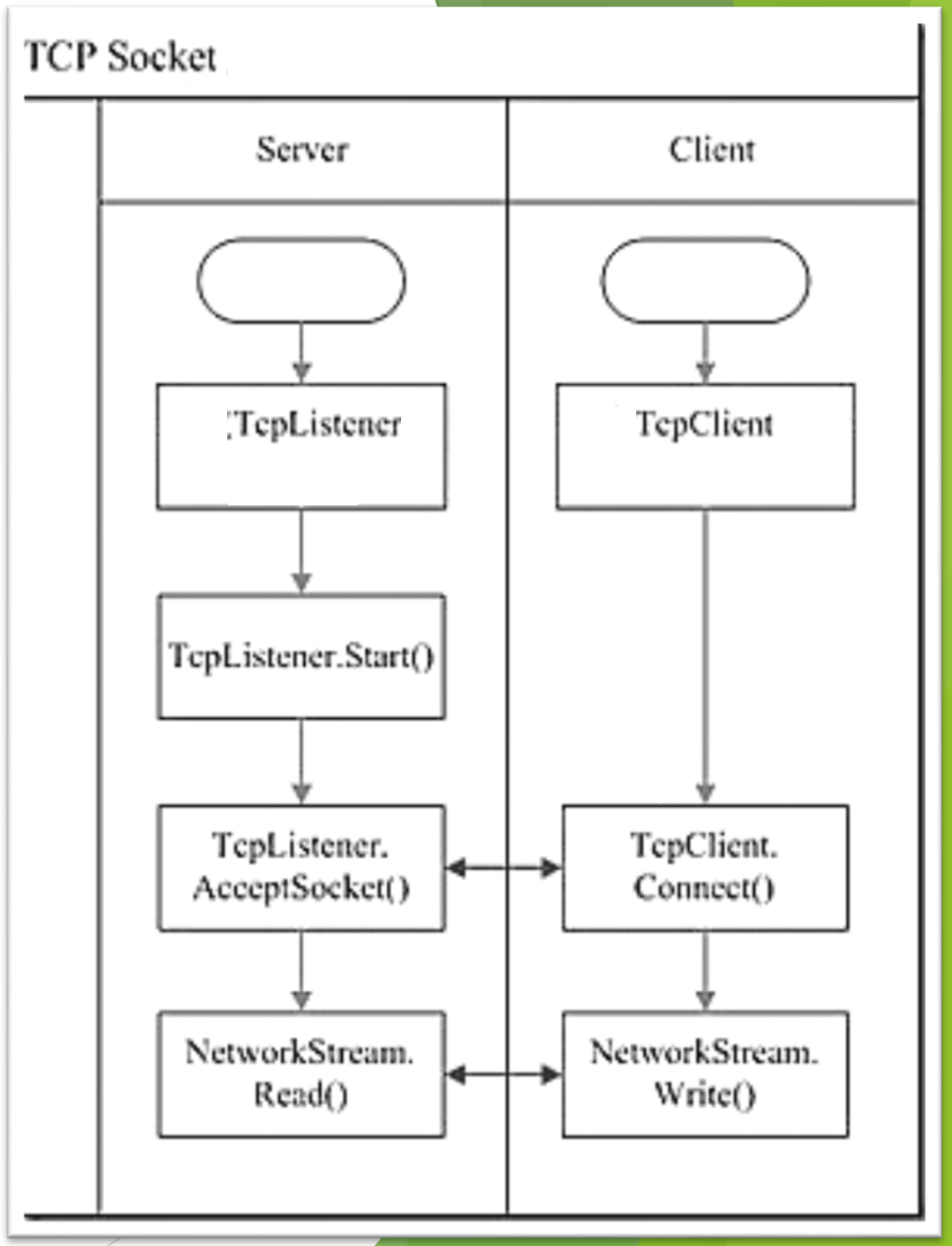
- One program is listening and the other is connecting
- The first one is a server and the last one a client

How to program TCP/IP communications

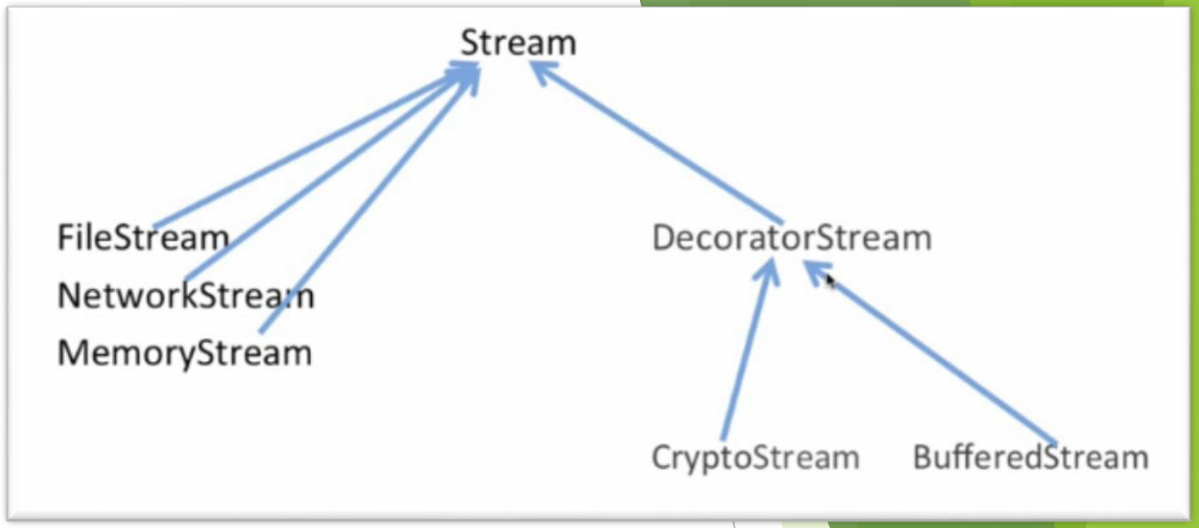
- C#, .Net
- System.Net.Sockets namespace

- Two classes
  - TcpListener (Class of a server)
  - TcpClient (Class of a client)

- After connection, data are communicated through a NetworkStream



What is a Stream ?



- Stream Classes provide generic methods for dealing with input/output
- NetworkStream class provides methods for sending and receiving data over stream sockets
- They are Methods similar to te other stream classes including Read and Write methods

The TcpClient Class

- The TcpClient class provides client-side connections which is used to communicate with the TCP server.
- The TcpClient class provides simple methods for connecting to another socket application over the network, and for sending and receiving data to it

Public Methods of TcpClient Class

Name	Description
Close()	Closes the TCP connection.
Connect()	Connects to a remote TCP host.
GetStream()	Returns the NetworkStream used for transferring data between the client and the remote host.

The TcpListener Class

- Typically, a server-side application starts by binding to the local endpoint and listening to incoming requests from clients :

  (IP_Address_Server : Port_Num_Server)

- As soon as a client is found knocking on the port, the application activates by accepting the request and then creating a channel that is then responsible for communicating with the client :

  (IP_Address_ServerforClient : Port_Num_ServerforClient)

- The application can continue to listen for more incoming client requests on the main thread.

The TcpListener Class

- The TcpListener sequence is then :
  - listening to the client's request,
  - accepting it,
  - and then creating a new instance of theTcpClient class that we can use to communicate with the client,
  - Just like the TcpClient, the TcpListener also provide a NetworkStream for communications after connection

The TcpListener Class

Public Methods

Name	Description
AcceptSocket()	Accepts a pending connection request and returns a Socket object to use to communicate with the client
AcceptTcpClient()	Accepts a pending connection request and returns a TcpClient object to use to communicate with the client
Pending()	Indicates whether there are any connection requests pending
Start()	Causes the TcpListener to start listening for connection requests
Stop()	Closes the listener

TCP Client

C# Code

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

```
// Create a TcpClient.
// Note, for this client to work you need to have a TcpServer
// connected to the same address as specified by the server, port
// combination
```

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

// Translate the passed message into ASCII and store it as a Byte array.

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

```
// Get a client stream for reading and writing.
//  Stream stream = client.GetStream();
```

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

// Send the message to the connected TcpServer.

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

Now we are going to receive the TcpServer.response.

// Buffer to store the response bytes.

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

// String to store the response ASCII representation.

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

// Read the first batch of the TcpServer response bytes.

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
using System;
using System.Net.Sockets;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Client("127.0.0.1", 13000, "Test");
        }


        static void Client(String server, Int32 port, String message)
        {
            TcpClient client = new TcpClient(server, port);

            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

            NetworkStream stream = client.GetStream();

            stream.Write(data, 0, data.Length);

            Console.WriteLine("Sent: {0}", message);

            data = new Byte[256];

            String responseData = String.Empty;

            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            Console.WriteLine("Received: {0}", responseData);

            stream.Close();
            client.Close();

            Console.WriteLine("\n Press Enter to continue...");
            Console.Read();
        }
    }
}
```

TCP Client C# Code

Jean-Yves Tigli - tigli@unice.fr – BAT4 - Networks – Internet – Web Services

TCP Server

C# Code

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Set the TcpListener on port

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// TcpListener server = new TcpListener(port);

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Start listening for client requests.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Buffer for reading data

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Enter the listening loop.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Perform a blocking call to accept requests.
// You could also user server.AcceptSocket() here.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Server C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Get a stream object for reading and writing

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Client C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Loop to receive all the data sent by the client.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

TCP Client C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Translate data bytes to a ASCII string.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

TCP Client C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Process the data sent by the client.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr – BAT4 – Networks – Internet – Web Services

TCP Client C# Code

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);
    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Send back a response.

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

```
static void Server(String server_addr, Int32 port)
{
    TcpListener server = null;
    IPAddress localAddr = IPAddress.Parse(server_addr);

    server = new TcpListener(localAddr, port);

    server.Start();

    Byte[] bytes = new Byte[256];
    String data = null;

    while (true)
    {
        Console.Write("Waiting for a connection... ");

        TcpClient client = server.AcceptTcpClient();
        Console.WriteLine("Connected!");

        data = null;

        NetworkStream stream = client.GetStream();

        int i;

        while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
        {
            data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);
            Console.WriteLine("Received: {0}", data);

            data = data.ToUpper();
            byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

            stream.Write(msg, 0, msg.Length);
            Console.WriteLine("Sent: {0}", data);
        }
        client.Close();

        Console.WriteLine("\nHit enter to continue...");
        Console.Read();
    }
}
```

// Shutdown and end connection

TCP Server C# Code

Jean-Yves Tigli - tigli@unice.fr - BAT4 - Networks - Internet - Web Services

08/03/2018

QUESTIONS ?

Next, See Lab on your first TCP Server …