CHƯƠNG 3
LẬP TRÌNH SOCKETS

NỘI DUNG

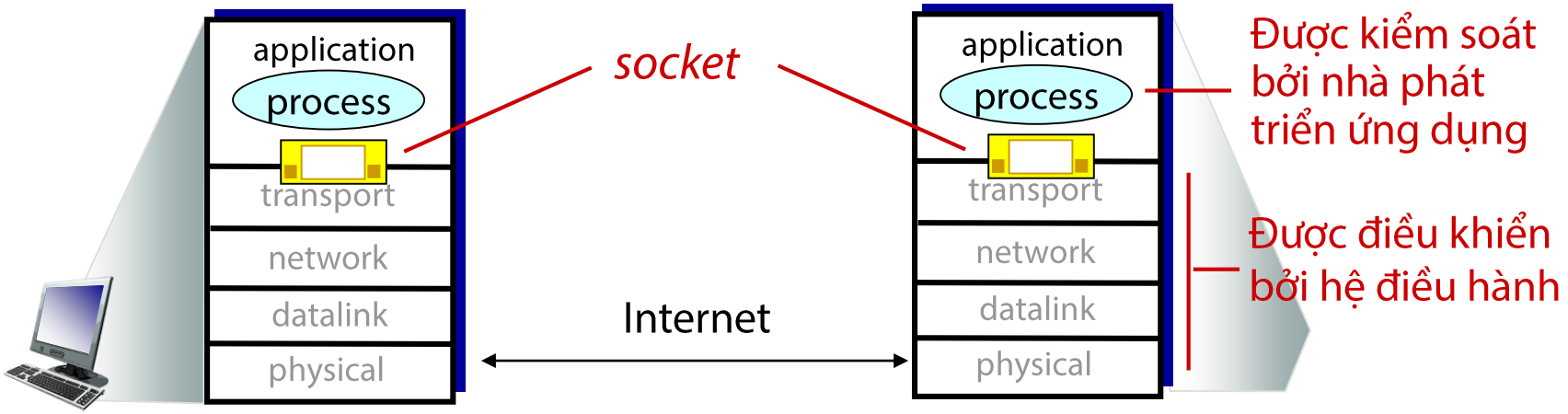
- Giới thiệu socket
- Khảo sát chức năng các class trong lập trình mạng:
  - TCPClient, TCPListener
  - UDPClient
  - IPAddress, IPHostEntry, IPEndpoint …
- Khai báo và sử dụng class UDP, TCP

SOCKET

- Lập trình socket là nền tảng của lập trình mạng
- Socket là một đối tượng đại diện cho một điểm truy cập mức thấp vào IP stack
- Socket có thể ở chế độ mở, đóng
- Socket có thể gửi, nhận dữ liệu
- Dữ liệu được gửi theo từng khối (packet), khoảng vài KB/lần

SOCKET (CNT)

- Tiến trình gửi/nhận thông điệp đến/ từ socket của nó
- Socket tương tự như "cửa ra vào"



SOCKET (CNT)

Socket(AddressFamily, SocketType, ProtocolType)

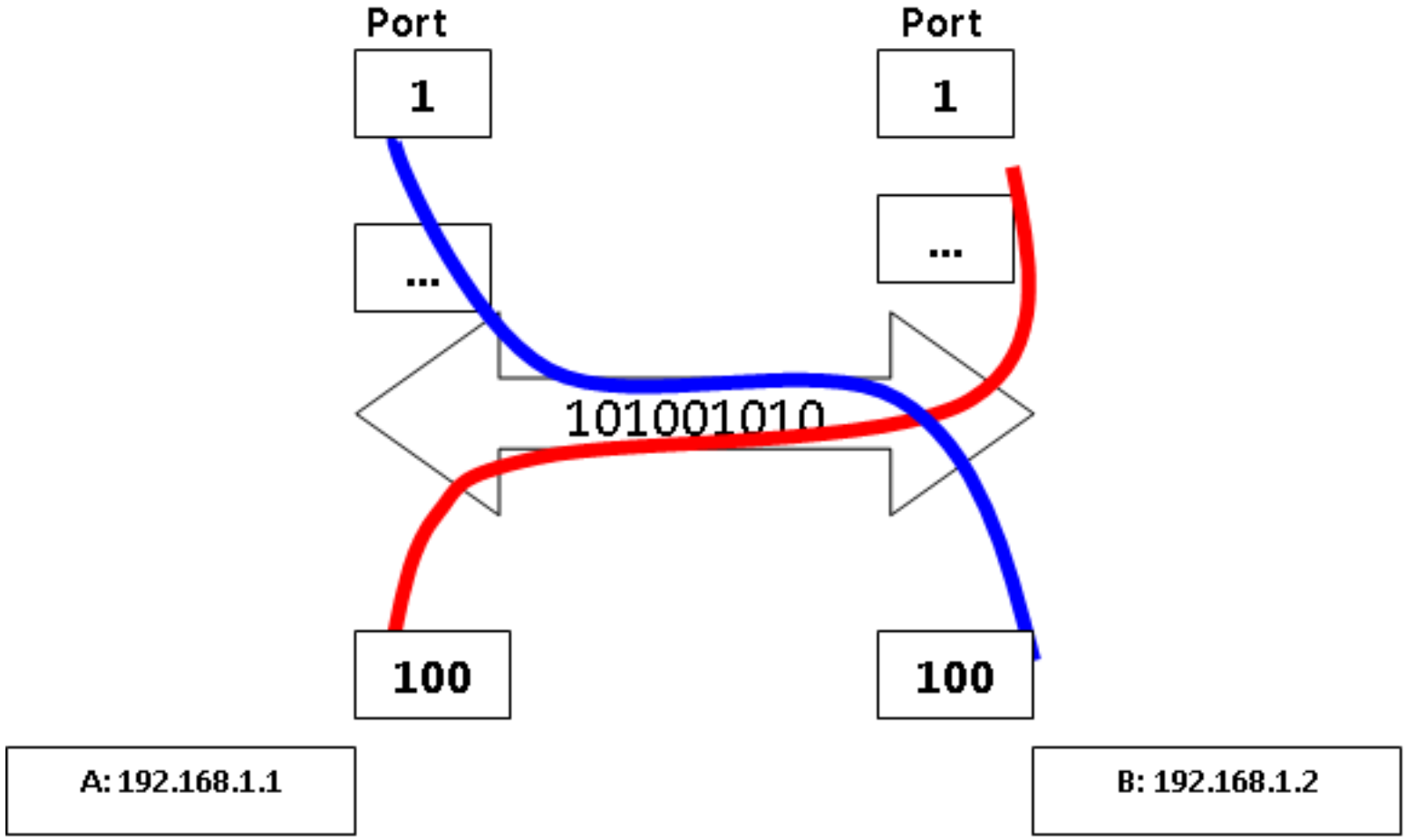
```
// Khởi tạo 1 Socket gửi nhận dữ liệu thông qua kết nối TCP
s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.TCP );
```

InterNetwork	Address for IP version 4.
InterNetworkV6	Address for IP version 6.
AppleTalk	AppleTalk address.
Atm	Native ATM services address.

Stream	Hỗ trợ tin cậy, 2 chiều, có hướng. Thường dung cho TCP
Dgram	Hỗ trợ datagram, kết nối không hướng. Dữ liệu có thể trùng, mất hoặc sai thứ tự. Thường dùng cho UDP

IP
TCP
UDP
ICMP
IPv4
IPv6

KHÁI NIỆM ĐỊA CHỈ VÀ CỔNG



ĐỊA CHỈ VÀ CỔNG

- Trong máy có rất nhiều ứng dụng trao đổi với các ứng dụng khác qua mạng
- Mỗi máy tính chỉ có duy nhất một đường truyền dữ liệu (để gửi và nhận)

→ Có thể xảy ra "nhầm lẫn" khi máy A gửi dữ liệu đến máy B, máy B không biết là dữ liệu đó gửi cho ứng dụng nào?

ĐỊA CHỈ VÀ CỔNG

- Mỗi ứng dụng trên máy B sẽ được gán một cổng [0..65,535]
- Khi ứng dụng máy A muốn gửi cho ứng dụng nào trên máy B thì chỉ việc điền thêm số port (trường RemotePort) vào gói tin cần gửi
- Trên máy B, các ứng dụng chỉ việc kiểm tra giá trị port trên mỗi gói tin xem có trùng với số hiệu port của mình (đã được gán – chính là giá trị LocalPort) hay không? Nếu bằng thì xử lý, ngược lại thì không làm gì.

GIAO THỨC VÀ PORT THƯỜNG GẶP

Port	Protocol
20	FTP data
21	FTP control
25	SMTP (email, outgoing)
53	DNS (Domain Name Service)
80	HTTP (Web)
110	POP3 (email, incoming)
143	IMAP (email, incoming)

MỘT SỐ QUY ĐỊNH

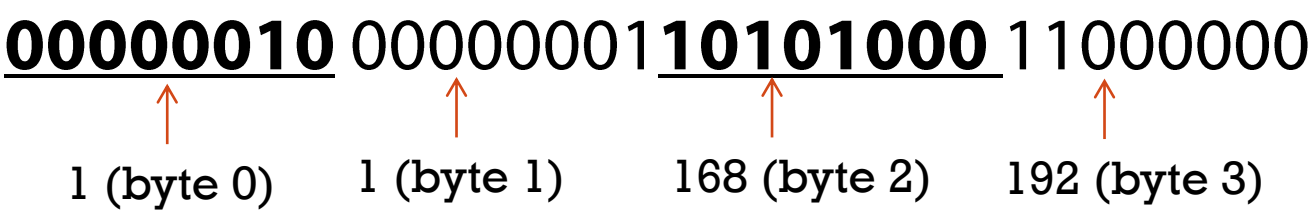
- Không bao giờ có 2 ứng dụng cùng dùng 1 port
- Port [0..1023] (Well-know): các ứng dụng quan trọng trên hệ điều hành
- Port [1024..49151] (Registered): cho lập trình viên (khuyến cáo tuân theo)
- Port [49152..65535] (Dynamic): dự trữ

LỚP IPADDRESS

- Trên Internet, mỗi trạm đều có một định danh duy nhất, thường gọi là địa chỉ (Address).
- Địa chỉ trên Internet là một tập hợp gồm 4 con số có giá trị từ 0-255 và cách nhau bởi dấu chấm.

LỚP IPADDRESS

- Địa chỉ có thể viết dưới các dạng:
  - Tên: ví dụ như May01, Server, …
  - Địa chỉ IP đặt trong một chuỗi: "192.168.1.1", "127.0.0.1“
  - Đặt trong một mảng 4 byte, mỗi byte chứa một số từ 0-255.
  - Là một số (long), có độ dài 4 byte. Ví dụ, với địa chỉ 192.168.1.2 ở trên thì giá trị đó sẽ là 33663168 (hệ thập phân khi xếp liền 4 byte ở trên lại với nhau)



LỚP IPADDRESS

- Để đổi một địa chỉ chuẩn ra dạng số chúng ta chỉ việc tính toán cho từng thành phần.
- Ví dụ:

  Đổi địa chỉ 192.168.1.2 ra số, ta tính như sau:

  2 * 256 ^ 3 + 1 * 256 ^ 2 + 168 * 256 ^ 1 + 192 * 256 ^ 0 = 33663168

LỚP IPADDRESS

Tên thuộc tính	Mô tả
Any	Cung cấp một địa chỉ IP (thường là 0.0.0.0) để chỉ ra rằng Server phải lắng nghe các hoạt động của Client trên tất cả các Card mạng (sử dụng khi xây dựng Server). Thuộc tính chỉ đọc.
Broadcast	Cung cấp một địa chỉ IP quảng bá (Broadcast, thường là 255.255.255.255), ở dạng số Long. Thuộc tính chỉ đọc.
Loopback	Trả về địa chỉ IP lặp (IP Loopback, ví dụ 127.0.0.1). Thuộc tính chỉ đọc.

https://docs.microsoft.com/en-us/dotnet/api/system.net.ipaddress?view=netframework-4.8

LỚP IPADDRESS

Tên phương thức	Mô tả
AddressFamily	Trả về họ địa chỉ của địa chỉ IP hiện hành. Nếu địa chỉ ở dạng IPv4 thì kết quả là Internetwork và InternetworkV6 nếu là địa chỉ IPv6.
Constructor	- IPAddress(Số_Long) → Tạo địa chỉ IP từ một số kiểu long - IPAddress(Mảng_Byte) → Tạo địa chỉ IP từ một mảng byte (4 byte)
GetAddressBytes	Chuyển địa chỉ thành mảng byte (4 byte)
HostToNetworkOrder	Đảo thứ tự byte của một số cho đúng với thứ tự byte trong địa chỉ IPAddress.

LỚP IPADDRESS

Tên phương thức	Mô tả
IsLoopback	Cho biết địa chỉ có phải là địa chỉ lặp hay không?
NetworkToHostOrder	Đảo thứ tự byte của một địa chỉ cho đúng với thứ tự byte thông thường.
Parse	Chuyển một địa chỉ IP ở dạng chuỗi thành một địa chỉ IP chuẩn (Một đối tượng IPAddress)
ToString	Trả về địa chỉ IP (một chuỗi) nhưng ở dạng ký pháp có dấu chấm. (Ví dụ "192.168.1.1").
TryParse (S: String)	Kiểm tra xem một địa chỉ IP (ở dạng chuỗi) có phải đúng là địa chỉ IP hợp lệ hay không? True = đúng

IPADDRESS: VÍ DỤ TẠO ĐỊA CHỈ

- **Cách 1:** Dùng hàm khởi tạo

```
Byte[] b = new Byte[4];
b[0] = 192;
b[1] = 168;
b[2] = 10;
b[3] = 10;
IPAddress Ip1 = new IPAddress(b);
```

IPADDRESS: VÍ DỤ TẠO ĐỊA CHỈ

- **Cách 2:** Dùng hàm khởi tạo

  IPAddress Ip2 = new IPAddress(16885952);

- **Cách 3:** Dùng hàm khởi tạo

  IPAddress Ip3 = IPAddress.Parse("172.16.1.1")

- **Cách 4:** Thông qua tính toán

  Long So = 192* 256^0+168* 256^1+1* 256^2 + 2*256^3;

  IPAddress Ip4 = new IPAddress(So);

IPADDRESS: VÍ DỤ KIỂM TRA ĐỊA CHỈ

```
private void KiemTra()
{
    IPAddress ip;
    String Ip4 = "127.0.0.1";
    String Ip5 = "999.0.0.1";
    MessageBox.Show(Ip4.ToString() + ": " +
IPAddress.TryParse(Ip4, out ip).ToString());
    MessageBox.Show(Ip5.ToString() + ": " +
IPAddress.TryParse(Ip5, out ip).ToString());
}
```

IPADDRESS: VÍ DỤ CHUYỂN ĐỊA CHỈ HIỆN HÀNH RA MẢNG

```
void ConvertToIPArray()
{
    IPAddress Ip3 = new IPAddress(16885952);
    Byte[] b = new Byte[4];
    b = Ip3.GetAddressBytes();
    MessageBox.Show("Address: " + b[0] + "."
+ b[1] + "." + b[2] + "." + b[3])
}
```

LỚP IPENDPOINT

- Lớp IPAddress mới chỉ cung cấp địa chỉ IP → còn thiếu số cổng (Port number) để trao đổi thông tin
- Lớp IPEndpoint tạo 1 endpoint:
  - IPAddress
  - Port number

LỚP IPENDPOINT

Tên thuộc tính	Mô tả
Address	Trả về hoặc thiết lập địa chỉ IP cho endpoint. (Trả về một đối tượng IPAddress)
AddressFamily	Lấy về loại giao thức mà Endpoint này đang sử dụng.
Port	Lấy về hoặc thiết lập số hiệu cổng của endpoint.

https://docs.microsoft.com/en-us/dotnet/api/system.net.ipendpoint?view=netframework-4.8

LỚP IPENDPOINT

Tên phương thức	Mô tả
IPEndPoint (Int64, Int32)	Tạo một đối tượng mới của lớp **IPEndPoint,** tham số truyền vào là địa chỉ IP (ở dạng số) và **cổng** sẽ dùng để giao tiếp.
IPEndPoint (IPAddress, Int32)	Tạo một đối tượng mới của lớp **IPEndPoint,** Tham số truyền vào là một địa chỉ **IPAddress** và số hiệu cổng dùng để giao tiếp. (Tham khảo cách tạo IPAddress ở phần trên)
Create	Tạo một endpoint từ một địa chỉ socket (socket address).
ToString	Trả về địa chỉ IP và số hiệu cổng theo khuôn dạng ĐịaChỉ: Cổng, ví dụ: 192.168.1.1:8080

LỚP IPENDPOINT: VÍ DỤ KHỞI TẠO

```
private void CreateEndpoint()
{
    // Tạo một địa chỉ IP
    IPAddress IPAdd = IPAddress.Parse("127.0.0.1");

    // Truyền vào cho hàm khởi tạo để tạo IPEndpoint
    IPEndPoint IPep = new IPEndPoint(IPAdd, 10000);
}
```

LỚP IPHOSTENTRY

- Chứa (Container) thông tin địa chỉ của các máy trạm trên Internet.

→ trước khi sử dụng cần phải “nạp" thông tin vào.

- Thường được dùng với lớp DNS

https://docs.microsoft.com/en-us/dotnet/api/system.net.iphostentry?view=netframework-4.8

LỚP IPHOSTENTRY

Tên thuộc tính	Mô tả
AddressList	Lấy về hoặc thiết lập danh sách các địa chỉ IP liên kết với một host.
Aliases	Lấy về hoặc thiết lập danh sách các bí danh (alias) liên kết với một host.
HostName	Lấy về hoặc thiết lập DNS name của host.

LỚP DNS

- DNS (Domain Name Service) giúp phân giải tên miền (Domain Resolution).

  ServerCNTT →192.168.3.8

- Cung cấp nhiều phương thức để xác định thông tin về máy cục bộ như tên, địa chỉ,…

https://docs.microsoft.com/en-us/dotnet/api/system.net.dns?view=netframework-4.8

LỚP DNS

Tên phương thức	Mô tả
GetHostEntry (String IP) GetHostEntry (IPAddress IP)	Trả về thông tin (IPHostEntry) của trạm có địa chỉ IP được truyền vào.
GetHostEntry (String hostname)	Trả về thông tin (IPHostEntry) DNS của một trạm.
HostName	Cho biết tên của máy vừa được phân giải. Nếu không phân giải được thì có giá trị là địa chỉ IP.

LỚP DNS

Tên phương thức	Mô tả
GetHostAddresses (String IP_Or_HostName)	Trả về tất cả các địa chỉ IP của một trạm. Kiểu IPAddress
GetHostEntry (String IP_Or_HostName) GetHostEntry (IPAddress IP)	Giải đáp tên hoặc địa chỉ IP truyền vào và trả về một đối tượng **IPHostEntry** tương ứng.
GetHostName	Lấy về tên của máy tính cục bộ (String).
GetHostEntry (String Hostname)	Chuyển tên của máy hoặc địa chỉ IP thành **IPHostEntry** tương ứng.

LỚP DNS

- **Lưu ý:**
  Đây là các **phương thức static**
- Cách dùng:
  Dns.Resolve, Dns.GetHostname, Dns.GetHostEntry

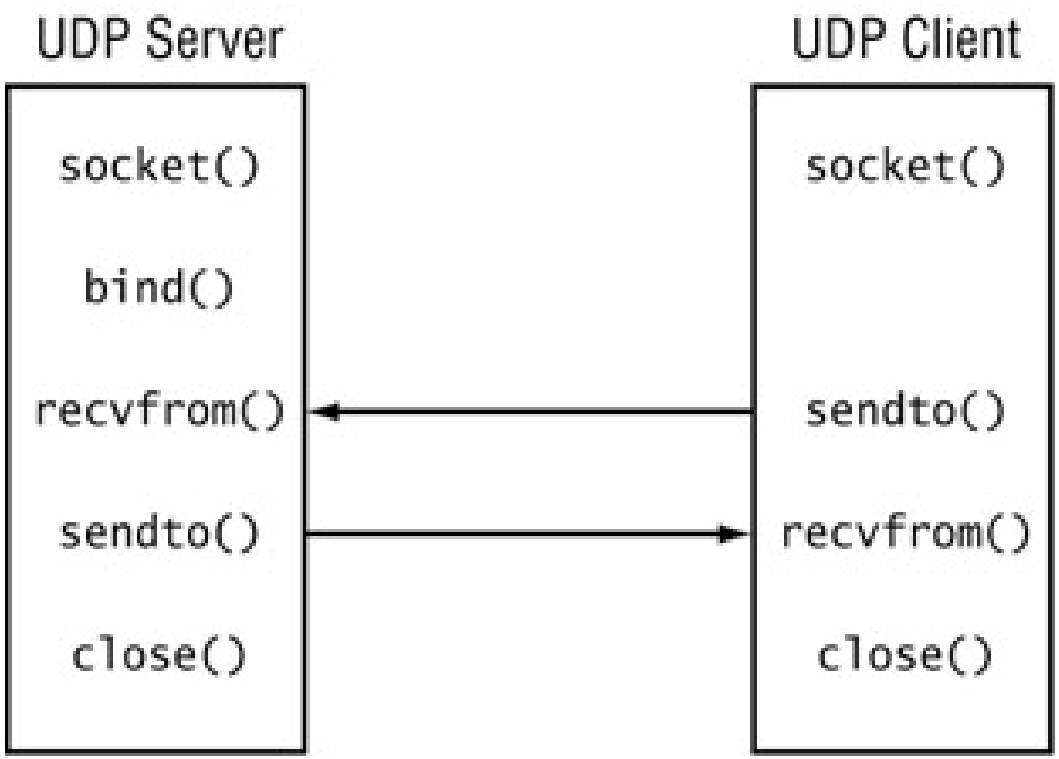
LỚP DNS: VÍ DỤ

```
private void ShowIPs()
{
    // Lấy tất cả địa chỉ IP của máy
    IPAddress[] add = Dns.GetHostAddresses("MY-PC");
    foreach (IPAddress ip in add)
    {
        MessageBox.Show(ip.ToString());
    }
    //Cách 2
    //for (int i = 0; i < add.Length; i++)
    //{
    //    MessageBox.Show(add[i].ToString());
    //}
}
```

LỚP UDPCLIENT

- UDP (User Datagram Protocol hay User Define Protocol)
- **Giao thức phi kết nối** (Connectionless)
- **Không tin cậy** bằng giao thức TCP nhưng tốc độ lại nhanh và dễ cài đặt.
- Có thể gửi các gói tin quảng bá (Broadcast) cho đồng thời nhiều máy

LỚP UDPCLIENT: TRÌNH TỰ KẾT NỐI



LỚP UDPCLIENT

Tên phương thức, thuộc tính	Mô tả
UdpClient ()	Tạo một đối tượng (thể hiện) mới của lớp UDPClient.
UdpClient (AddressFamily)	Tạo một đối tượng (thể hiện) mới của lớp UDPClient. Thuộc một dòng địa chỉ (AddressFamily) được chỉ định.
UdpClient (LocalPort: Int32)	Tạo một **UdpClient** và gắn (bind) một cổng cho nó.
Active	Lấy về hoặc thiết lập giá trị cho biết kết nối với máy ở xa đã được tạo ra chưa. Kiểu dữ liệu là bool.
Client	Lấy về hoặc thiết lập các socket.

https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.udpclient?view=netframework-4.8

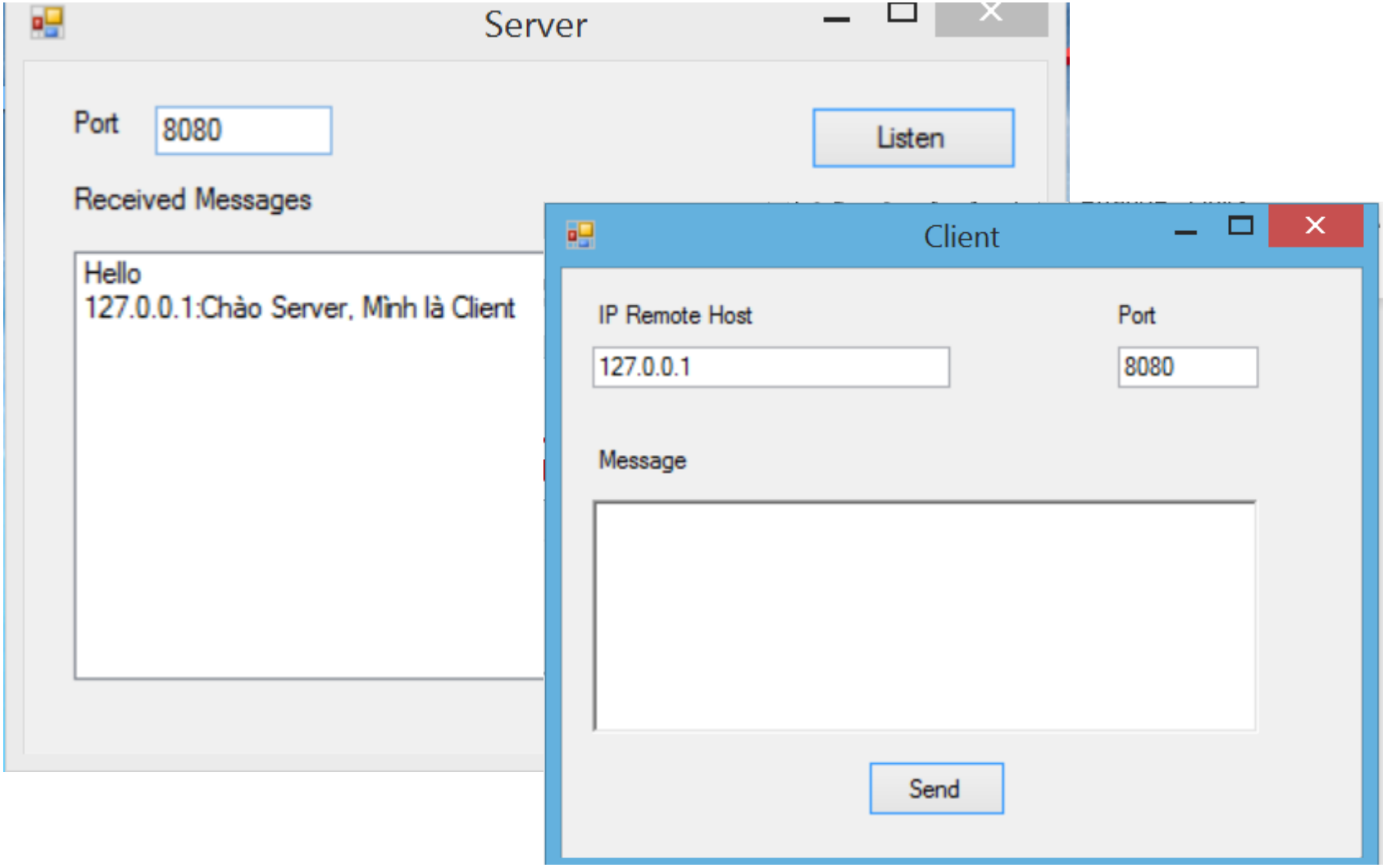
LỚP UDPCLIENT

Tên phương thức	Mô tả
UdpClient (IPEndPoint)	Tạo một **UdpClient** và gắn (bind) một IPEndpoint (gán địa chỉ IP và cổng) cho nó.
UdpClient (Int32, AddressFamily)	Tạo một **UdpClient** và gán số hiệu cổng, AddressFamily
UdpClient (Remotehost: String, Int32)	Tạo một **UdpClient** và thiết lập với một trạm từ xa mặc định.
UdpClient (IPEndPoint)	Tạo một **UdpClient** và gắn (bind) một IPEndpoint (gán địa chỉ IP và cổng) cho nó.

LỚP UDPCLIENT

Tên phương thức	Mô tả
BeginReceive	Nhận dữ liệu không đồng bộ từ máy ở xa.
BeginSend	Gửi không đồng bộ dữ liệu tới máy ở xa
Close	Đóng kết nối.
Connect	Thiết lập một Default remote host.
EndReceive	Kết thúc nhận dữ liệu không đồng bộ ở trên
EndSend	Kết thúc việc gửi dữ liệu không đồng bộ ở trên
Receive (EndPoint của máy ở xa) As Byte()	Nhận dữ liệu (đồng bộ) do máy ở xa gửi. (Đồng bộ có nghĩa là các lệnh ngay sau lệnh Receive chỉ được thực thi nếu Receive đã nhận được dữ liệu về, còn nếu nó chưa nhận được thì nó vẫn cứ chờ (blocking))
Send	Gửi dữ liệu (đồng bộ) cho máy ở xa.

LỚP UDPCLIENT: CHƯƠNG TRÌNH CHAT



LỚP UDPCLIENT: CLIENT

```
private void ClientSend_Click(object sender, EventArgs e)
{
    UdpClient udpClient = new UdpClient();
    //Lấy địa chỉ IP từ textbox và chuyển thành kiểu
IPAddress
    IPAddress ipadd = IPAddress.Parse(textBox1.Text);
    int port = Convert.ToInt32(textBox2.Text);
    IPEndPoint ipend = new IPEndPoint(ipadd, port);
    //Chuyển chuỗi dữ liệu nhập sang kiểu byte
    Byte[] sendBytes =
Encoding.UTF8.GetBytes(richTextBox1.Text);
    //Gởi dữ liệu đến IPEndPoint đã định nghĩa địa chỉ IP và
Port
    udpClient.Send(sendBytes, sendBytes.Length, ipend);
    //Xóa dữ liệu vừa gửi ở ô nhập
    richTextBox1.Text = "";
}
```

LỚP UDPCLIENT: SERVER

```
public void serverThread()
{
    int port = Convert.ToInt32(textBox1.Text);
    UdpClient udpClient = new UdpClient(port);
    while (true)
    {
        IPEndPoint IpEnd = new IPEndPoint(IPAddress.Any, 0);
        //Đón nhận và đẩy dữ liệu nhận được vào mảng Byte
        Byte[] recvBytes = udpClient.Receive(ref IpEnd);
        string Data = Encoding.UTF8.GetString(recvBytes);
        string mess = IpEnd.Address.ToString() + ":" +
IpEnd.Port.ToString() + ": " + Data.ToString();
        // Gọi hàm hiển thị thông điệp nhận được lên màn hình
        InfoMessage(mess);
    }
}
```

LỚP UDPCLIENT: TỔNG KẾT

- Gửi dữ liệu đi:
  - Bước 1: Chuyển chuỗi thành mảng byte
  - Bước 2: Gọi phương thức Send, trong đó truyền địa chỉ IP của máy ở xa vừa tạo ở 2 và thêm vào số cổng mà máy ở xa đang dùng để nhận dữ liệu.
- Khi nhận:

  Dùng phương thức Receive để nhận dữ liệu về. Khi đó, cần tạo một đối tượng IPEndPoint với địa chỉ và số cổng của máy ở xa muốn nhận dữ liệu.

LỚP TCPCLIENT

- TCP (Transport Control Protocol)
- Đảm bảo độ tin cậy trong các ứng dụng mạng
- Giao thức có kết nối.
- Trên Internet chủ yếu là dùng loại giao thức này, ví dụ như Telnet, HTTP, SMTP, POP3…
- Để lập trình theo giao thức TCP, .NET cung cấp hai lớp có tên là TCPClient và TCPListener.

LỚP TCPCLIENT

Tên phương thức	Mô tả
TcpClient ()	Tạo một đối tượng **TcpClient**.
TcpClient (IPEndPoint)	Tạo một **TcpClient** và gắn cho nó một EndPoint cục bộ. (Gán địa chỉ máy cục bộ và số hiệu cổng để sử dụng trao đổi thông tin về sau)
TcpClient (RemoteHost: String, RemotePort: Int32)	Tạo một đối tượng **TcpClient** và kết nối đến một máy có địa chỉ và số hiệu cổng được truyền vào. RemoteHost có thể là địa chỉ IP chuẩn hoặc tên máy.

https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.tcpclient?view=netframework-4.8

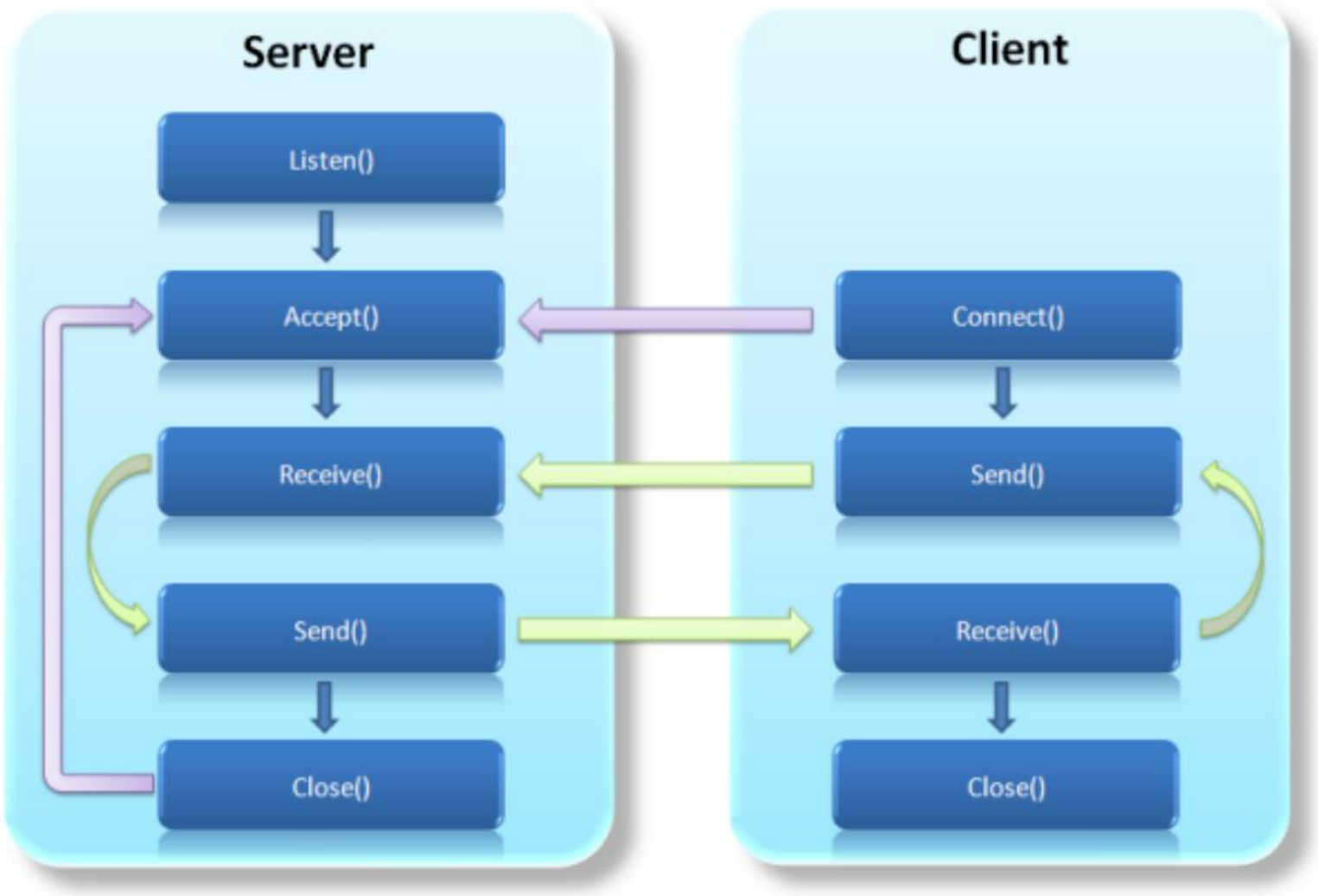
LỚP TCPCLIENT

Tên thuộc tính	Mô tả
Available	Cho biết số byte đã nhận về từ mạng và có sẵn để đọc.
Client	Trả về Socket ứng với TCPClient hiện hành.
Connected	Trạng thái cho biết đã kết nối được đến Server hay chưa

LỚP TCPCLIENT

Tên phương thức	Mô tả
Close	Giải phóng đối tượng **TcpClient** nhưng không đóng kết nối.
Connect (RemoteHost, Port)	Kết nối đến một máy TCP khác có tên và số hiệu cổng.
GetStream	Trả về NetworkStream để từ đó giúp ta gửi hay nhận dữ liệu. (Thường làm tham số khi tạo StreamReader và StreamWriter) . Khi đó gắn vào StreamReader và StreamWriter rồi ta có thể gửi và nhận dữ liệu thông qua các phương thức ReadIn, WriteIn tương ứng của các lớp này.

LỚP TCPCLIENT: TRÌNH TỰ KẾT NỐI



SO SÁNH TCP & UDP

TCP	UDP
Hoạt động tin cậy	Hoạt động không tin cậy
Phải thiết lập kết nối giữa client & server	Không cần thiết lập kết nối
Dữ liệu gửi không chứa địa chỉ và port của máy nhận	Phải xác định địa chỉ và port của máy nhận trong dữ liệu gửi

LỚP TCPLISTENER

Cho phép xây dựng các ứng dụng Server như SMTP Server, FTP Server, DNS Server, POP3 Server hay server tự định nghĩa,…

Ứng dụng server khác với ứng dụng Client?

LỚP TCPLISTENER

Tên phương thức	Mô tả
TcpListener (Port: Int32)	Tạo một **TcpListener** và lắng nghe tại cổng chỉ định.
TcpListener (IPEndPoint)	Tạo một **TcpListener** với giá trị Endpoint truyền vào.
TcpListener (IPAddress, Port: Int32)	Tạo một **TcpListener** và lắng nghe các kết nối đến tại địa chỉ IP và cổng chỉ định.
Active	Trả về một giá trị cho biết TcpListener đang lắng nghe kết nối từ client. Kiểu bool
Server	Trả về socket

LỚP TCPLISTENER

Tên phương thức	Mô tả
AcceptSocket	Chấp nhận một yêu cầu kết nối đang chờ.
AcceptTcpClient	Chấp nhận một yêu cầu kết nối đang chờ. (Ứng dụng sẽ dừng tại lệnh này cho đến khi nào có một kết nối đến)
Pending	Cho biết liệu có kết nối nào đang chờ đợi không? (True = có).
Start	Bắt đầu lắng nghe các yêu cầu kết nối.
Stop	Dừng việc nghe.

DEBUGGING NETWORK CODE

- Phương pháp quan trọng để theo dõi và giải quyết các lỗi phát sinh khi viết chương trình
- Nên dùng cấu trúc try/catch

```
try {
        serverSocket.Bind(ipepServer);
        serverSocket.Listen(-1);
}
catch(SocketException e) {
         MessageBox.Show(e.Message);
}
```

DEBUGGING NETWORK CODE

- Để xác định các trục trặc trong ứng dụng multithreaded, cơ chế theo vết (tracing) đóng vai trò cực kỳ quan trọng
- Nên dùng System.Diagnostics.Trace
- Hoặc các câu lệnh dạng Console.WriteLine