

- Ví dụ về phương thức truyền lại dữ liệu
- SndRcvData()
- Ví dụ RetryUdpClient.cs
- Sử dụng phương thức trong chương trình

☐ **Xây dựng ứng dụng UDP hoàn chỉnh**

Ví dụ BestUdpClient.cs

149

Chương 4

Sử dụng các lớp trợ giúp

150

Mục lục chương

1. Giới thiệu
2. Lớp TcpClient
3. Lớp TcpListener
4. Lớp UdpClient

151

Giới thiệu

- Ở các chương trước chúng ta đã học các lớp Socket mức thấp để tạo ứng dụng TCP và UDP
- Tuy nhiên với người mới lập trình mạng thì cách này khó hiểu và dễ nhầm lẫn
- Chương này cung cấp các lớp trợ giúp nhằm đơn giản quá trình lập trình mạng

152

Lớp TcpClient

- Lớp TcpClient được đặt trong System.Net.Sockets namespace
- Được thiết kế để trợ giúp cho việc xây dựng ứng dụng TCP Client

153

Hàm tạo

- Có 3 hàm tạo của lớp TcpClient đó là:
- **TcpClient()**
- Hàm tạo này tạo ra một đối tượng TcpClient và gắn nó với địa chỉ cục bộ của hệ thống đồng thời gắn với một địa chỉ cổng ngẫu nhiên.
- Sau khi gọi hàm tạo này ta phải kết nối nó đến một máy tính ở xa sử dụng phương thức Connect()

154

Hàm tạo

- Ví dụ
 - `TcpClient newcon = new TcpClient();`
`newcon.Connect("www.ispnet.net", 8000);`

155

Hàm tạo

- Hàm tạo thứ 2 là:
- **`TcpClient(IPEndPoint localEP)`**
- Hàm tạo thứ 2 cho phép ta chỉ ra địa chỉ cục bộ của hệ thống cùng với số hiệu cổng
- Được sử dụng phổ biến khi máy tính có nhiều card mạng

156

Hàm tạo

- Ví dụ
 - `EndPoint iep = new EndPoint(IPAddress.Parse("192.168.1.6"), 8000);`
 - `TcpClient newcon = new TcpClient(iep);`
 - `newcon.Connect("www.isp.net", 8000);`

157

Hàm tạo

- Hàm tạo thứ 3 là:
- **`TcpClient(String host, int port)`**
- Hàm tạo này được sử dụng phổ biến nhất
- Nó cho phép chỉ ra địa chỉ cũng như số hiệu cổng của máy tính ở xa
- Vì vậy không cần gọi phương thức `Connect()` nữa

158

Hàm tạo

- Ví dụ:
 - `TcpClient newcon = new`
`TcpClient("www.isp.net", 8000);`

159

Các phương thức của lớp TcpClient

Method	Description
<code>Close()</code>	Closes the TCPconnection
<code>Connect()</code>	Attempts to establish a TCP connection with a remote device
<code>Equals()</code>	Determines if two TcpClient objects are equal
<code>GetHashCode()</code>	Gets a hash code suitable for use in hash functions
<code>GetStream()</code>	Gets a Stream object that can be used to send and receive data
<code>GetType()</code>	Gets the Type of the current instance
<code>ToString()</code>	Converts the current instance to a String object

160

Các thuộc tính của lớp TcpClient

Property	Description
LingerState	Gets or sets the socket linger time
NoDelay	Gets or sets the delay time used for sending or receiving TCP buffers that are not full
ReceiveBufferSize	Gets or sets the size of the TCP receive buffer
ReceiveTimeout	Gets or sets the receive timeout value of the socket
SendBufferSize	Gets or sets the size of the TCP send buffer
SendTimeout	Gets or sets the send timeout value of the socket

161

Ví dụ

- Ví dụ về TcpClientSample.cs

162

Lớp TcpListener

- Lớp TcpListener cũng được đặt trong System.Net.Sockets namespace
- Nó giúp đơn giản hóa việc tạo ra ứng dụng TCP Server

163

Hàm tạo

- TcpListener có 3 hàm tạo:
 - **TcpListener(int port)** This constructor binds to a specific local port number.
 - **TcpListener(IPEndPoint ie)** This constructor binds to a specific local EndPoint object.
TcpListener(IPAddress addr, int port)
 - **TcpListener(IPAddress addr, int port)** This constructor binds to a specific local IPAddress object and port number.

164

Hàm tạo

- Không giống như hàm tạo của lớp `TcpClient`, hàm tạo của lớp `TcpListener` luôn cần phải chỉ ra một số hiệu cổng cụ thể mà server sẽ lắng nghe ở đó
- Nếu trên máy chủ có nhiều card mạng thì ta cũng cần chỉ ra một địa chỉ cụ thể mà server sẽ làm việc trên đó

165

Các phương thức của lớp `TcpListener`

Method	Description
<code>AcceptSocket()</code>	Accepts an incoming connection on the port and assign it to a <code>Socket</code> object
<code>AcceptTcpClient()</code>	Accepts an incoming connection on the port and assigns it to a <code>TcpClient</code> object
<code>Equals()</code>	Determines if two <code>TcpListener</code> objects are equal
<code>GetHashCode()</code>	Gets a hash code suitable for use in hash functions
<code>GetType()</code>	Gets the type of the current instance
<code>Pending()</code>	Determines if there are pending connection requests
<code>Start()</code>	Starts listening for connection attempts
<code>Stop()</code>	Stops listening for connection attempts (closes the socket)
<code>ToString()</code>	Creates a string representation of the <code>TcpListener</code> object

166

Các phương thức của lớp TcpListener

- Để tạo ra một TcpListener và lắng nghe các kết nối đến từ Client ta làm như sau:

```
TcpListener server = new  
TcpListener(IPAddress.Parse("127.0.0.1"),  
9050);  
server.Start();  
TcpClient newclient =  
server.AcceptTcpClient();
```

167

Ví dụ

- Ví dụ TcpListenerSample.cs

Một ứng dụng Server đơn giản

168

Lớp UdpClient

- Lớp UdpClient cũng được đặt trong System.Net.Sockets namespace
- Lớp này được thiết kế nhằm trợ giúp cho lập trình viên trong việc xây dựng các ứng dụng UDP Client và UDP Server

169

Hàm tạo

- UdpClient có 4 hàm tạo
 - **UdpClient()** This format creates a new UdpClient instance not bound to any specific address or port.
 - **UdpClient(int port)** This constructor binds the new UdpClient object to a specific UDP port number.

170

Hàm tạo

- Hai hàm tạo còn lại là
 - UdpClient(IPEndPoint iep)** This constructor binds the new [UdpClient](#) object to a specific local IP address and port number.
 - UdpClient(string host, int port)** This format binds the new [UdpClient](#) object to any local IP address and port and associates it with a specific remote IP address and port.

171

Các phương thức của lớp UdpClient

Method	Description
Close()	Closes the underlying socket
Connect()	Allows you to specify a remote IP endpoint to send and receive data with
DropMulticastGroup()	Removes the socket from a UDP multicast group
Equals()	Determines if two UdpClient objects are equal
GetHashCode()	Gets a hash code for the UdpClient object
GetType()	Gets the Type of the current object
JoinMulticastGroup()	Adds the socket to a UDP multicast group
Receive()	Receives data from the socket
Send()	Sends data to a remote host from the socket
ToString()	Creates a string representation of the UdpClient object

172

Sử dụng lớp UdpClient trong chương trình

- Có một số khác biệt nhỏ giữa hàm nhận và gửi dữ liệu của lớp UdpClient so với hàm SendTo() và ReceiveFrom() của lớp Socket

173

Phương thức Receive()

- Lớp UdpClient sử dụng phương thức Receive() để chấp nhận các gói tin đến trên một địa chỉ và một số hiệu cổng cụ thể.
- Nguyên mẫu của phương thức này như sau:
 - byte[] Receive(ref IPEndPoint iep)

174

Phương thức Receive()

- Ví dụ về sử dụng phương thức Receive()
 - `EndPoint ipep = new EndPoint(IPAddress.Any, 9050);`
 - `UdpClient newsock = new UdpClient(ipep);`
 - `byte[] data = new byte[1024];`
 - `EndPoint ipep2 = new EndPoint(IPAddress.Any, 0); data = host.Receive(ref ipep2);`
 - `Console.WriteLine("The remote host is: {0}, port {1}", ipep2.Address, ipep2.Port);`
`Console.WriteLine(Encoding.ASCII.GetString(data));`

175

Phương thức Send()

- Có 3 formats của hàm Send()
 - **Send(byte[] data, int sz)** This format sends the byte array *data* of size *sz* to the default remote host. To use this format, you must specify a default remote UDP host using either [UdpClient](#) constructor, or the Connect() method:
 - `UdpClient host = new UdpClient("127.0.0.1", 9050);`

176

Phương thức Send()

- Hai formats khác là:
 - **Send(byte[] data, int sz, IPEndPoint iep)**
This format sends the byte array *data* of size *sz* to the remote host specified by *iep*.
 - **Send(byte[] data, int sz, string host, int port)** This format sends the byte array *data* of size *sz* to the host *host* at port *port*.

177

Ví dụ

- Ví dụ về UdpClient Server
- Ví dụ về UdpClient Client

178