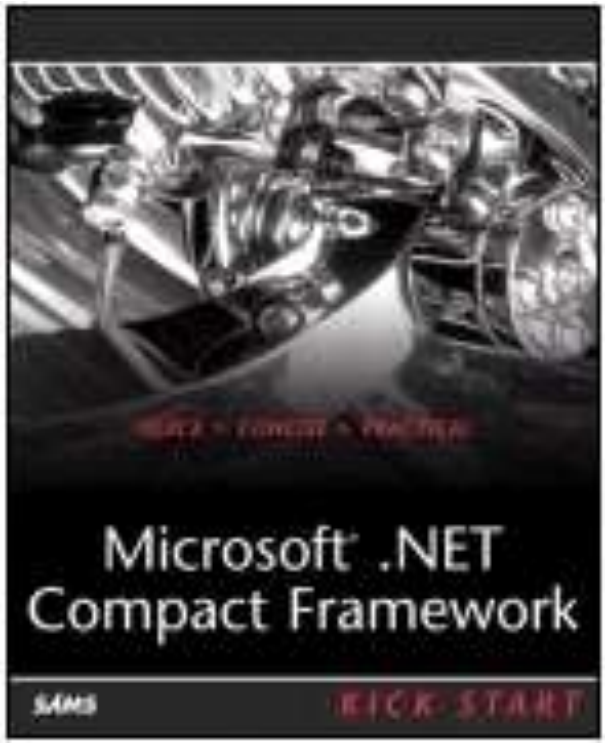
Đại học Khoa học Tự nhiên, ĐHQG-HCM
Khoa Công Nghệ Thông Tin

# Bài 5: Lập trình mạng trên .Net Compact Framework

TS. Trần Minh Triết

# Tham khảo

- ***Microsoft® .NET Compact Framework Kick Start*,**
  **Erik Rubin, Ronnie Yates(2003),** Sams Publishing
  - Chapter 5. Network Connectivity with the .NET Compact Framework



# Nội dung

- Sử dụng Socket
- Sử dụng kết nối UDP
- Sử dụng HttpWebRequest
- Sử dụng kết nối hồng ngoại

# Một số nhận xét ban đầu

- Các port có giá trị dưới 1024 thường được HĐH dành riêng cho các dịch vụ chuẩn
- Nên chọn port có giá trị từ 2000 đến 60000

- Chọn lựa giữa TCP/IP và UDP
  - TCP: đảm bảo “error-free delivery”, chậm hơn UDP
  - UDP: không đảm bảo “error-free delivery”, nhanh hơn TCP

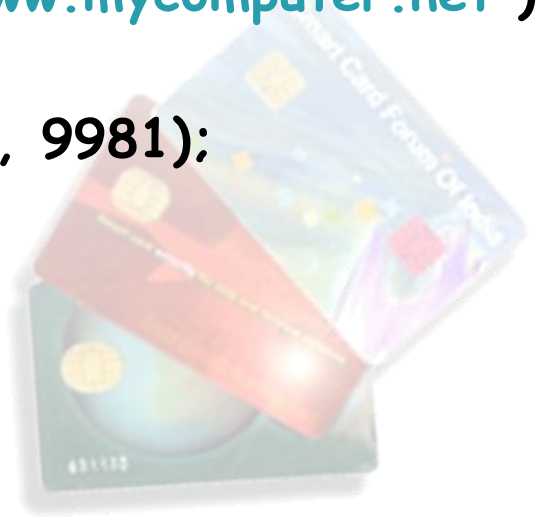
# Tạo kết nối tại **Client**

- Sử dụng địa chỉ IP

```
EndPoint l_EndPoint = new IPEndPoint
    (IPAddress.Parse( "172.68.25.34"), Convert.ToInt16(9981));
```

- Sử dụng tên của server (dùng DNS)

```
IPHostEntry l_IPHostEntry =Dns.Resolve("www.mycomputer.net");
EndPoint l_EndPoint = new
    IPEndpoint(l_IPHostEntry.AddressList[0], 9981);
```



# Tạo kết nối tại **Client**

```
try
{
    Socket l_Socket = new
    Socket(Socket(AddressFamily.InterNetwork,
   SocketType.Stream, ProtocolType.Tcp);
    l_Socket.Connect(l_EndPoint);
    if (l_Socket.Connected)
    {
       // l_Socket is now ready to send and receive data
    }
}
catch (SocketException e)
{ /* do something about it */ }
```

# Nhận yêu cầu kết nối tại **Host**

```
m_listenSocket = new Socket(AddressFamily.InterNetwork,
        SocketType.Stream, ProtocolType.Tcp);

m_listenSocket.Bind(new IPEndPoint(IPAddress.Any, 8758));
m_listenSocket.Listen((int)SocketOptionName.MaxConnections);

m_connectedSocket = m_listenSocket.Accept();

if (m_connectedSocket != null)
{
    if (m_connectedSocket.Connected)
    {
       // Someone has connected to us.
    }
 }
```

# Truyền tin bằng **Socket**

- Truyền dữ liệu: **Socket.Send**
- **Send (Byte[] buffer)**
- **Send (Byte[] buffer, SocketFlags socketFlags)**
- **Send (Byte[] buffer, Int32 size, SocketFlags socketFlags)**
- **Send (Byte[] buffer, Int32 offset,**
  **Int32 size, SocketFlags socketFlags)**

- Ví dụ:

**l_Socket.Send (l_buffer, 16, SocketFlags.None);**

Truyền tối đa 16 byte dữ liệu trong **l_buffer**

**l_Socket.Send (l_buffer, 2, 6, SocketFlags.None);**

Truyền tối đa 6 byte dữ liệu trong **l_buffer** từ byte thứ 3

# Truyền tin bằng **Socket**

```
l_Socket.Send(Encoding.ASCII.GetBytes("Send me") );
l_Socket.Send(Encoding.Unicode.GetBytes("Send me") );
l_Socket.Send(Encoding.ASCII.GetBytes(Convert.ToString(2003)) );
l_Socket.Send(Encoding.ASCII.GetBytes(Convert.ToString(2.71)) );
```

# Nhận tin bằng **Socket**

- Nhận dữ liệu: **Socket.Receive**
- **Receive (Byte[] buffer)**
- **Receive (Byte[] buffer, SocketFlags socketFlags)**
- **Receive (Byte[] buffer, Int32 size, SocketFlags socketFlags)**
- **Receive (Byte[] buffer, Int32 offset,**
  **Int32 size, SocketFlags socketFlags)**

- Ví dụ:

**l_Socket.Receive (l_buffer, 16, SocketFlags.None);**

Nhận tối đa 16 byte dữ liệu vào **l_buffer**

**l_Socket.Receive (l_buffer, 2, 6, SocketFlags.None);**

Nhận tối đa 6 byte dữ liệu ghi vào **l_buffer** từ byte thứ 3

# Nhận tin bằng **Socket**

- `string l_ASCII = Encoding.ASCII.GetString(l_Buffer);`
- `string l_ASCII = Encoding.ASCII.GetString(l_Buffer);`
- `int l_Integer = Convert.ToInt32(Encoding.ASCII.GetString(l_Buffer));`
- `Double l_Double = Convert.ToDouble(Encoding.ASCII.GetString(l_Double));`

# Sử dụng kết nối **UDP**

- **UDP** và **TCP**
  - **UDP**: connectionless protocol
  - **TCP**: connection-oriented protocol

- Sử dụng class **UdpClient**
  - `void Connect(String hostname, Int32 port);`
  - `void Send(Byte[] dgram, Int32 bytes);`
  - `void Receive(ref IPEndPoint remoteEP);`

- Ví dụ:

```
l_UdpClient.Connect(IPAddress.Parse("172.68.25.34"), 9981);
Send(aBuffer, aBuffer.Length, "www.mycomputer.net", 9981);
```

# Sử dụng **UdpClient**

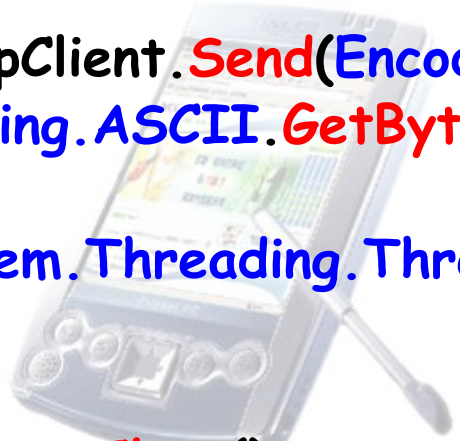
```
IPEndPoint      senderIP = new
        IPEndPoint(IPAddress.Parse("192.168.0.200"),
        Convert.ToInt32(8758));

UdpClient       l_UdpClient = new UdpClient();
l_UdpClient.Connect(senderIP);

for (int i = 0; i < 20; i++)
{
    l_UdpClient.Send(Encoding.ASCII.GetBytes("Hello_UDP_1"),
   Encoding.ASCII.GetBytes("Hello_UDP_1").Length);

    System.Threading.Thread.Sleep(1000);
}

l_UdpClient.Close();
```



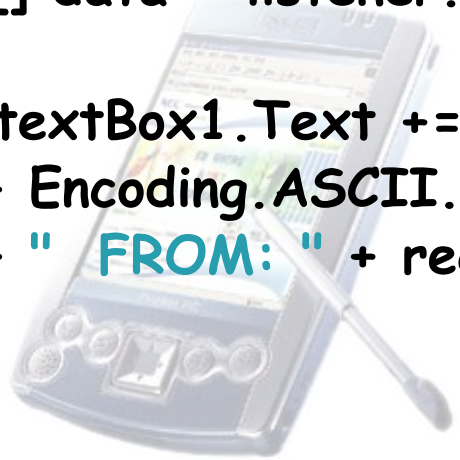
# Sử dụng **UdpClient**

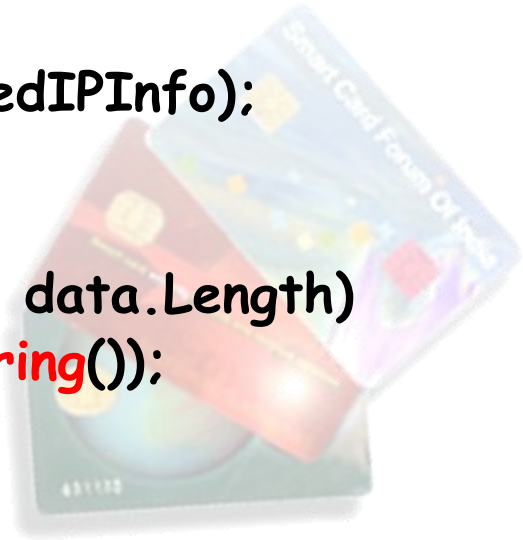
```
IPEndPoint  listenerIP = new IPEndPoint(IPAddress.Any, 8758);
UdpClient   listener = new UdpClient(listenerIP);

for (int i = 0; i < Convert.ToInt16(this.txtMaxPackets.Text); i++)
{
    // Now receive the three datagrams from the listener
    IPEndPoint receivedIPInfo = new IPEndPoint(IPAddress.Any, 0);

    byte[] data = listener.Receive(ref receivedIPInfo);

    this.textBox1.Text += ("GOT: "
        + Encoding.ASCII.GetString(data, 0, data.Length)
        + "  FROM: " + receivedIPInfo.ToString());
}
```





# Sử dụng **UdpClient**

```
UdpClient udpClient = new UdpClient(5000,
    AddressFamily.InterNetwork);

udpClient.BeginReceive(new AsyncCallback(ReceiveCallback), udpClient)
```

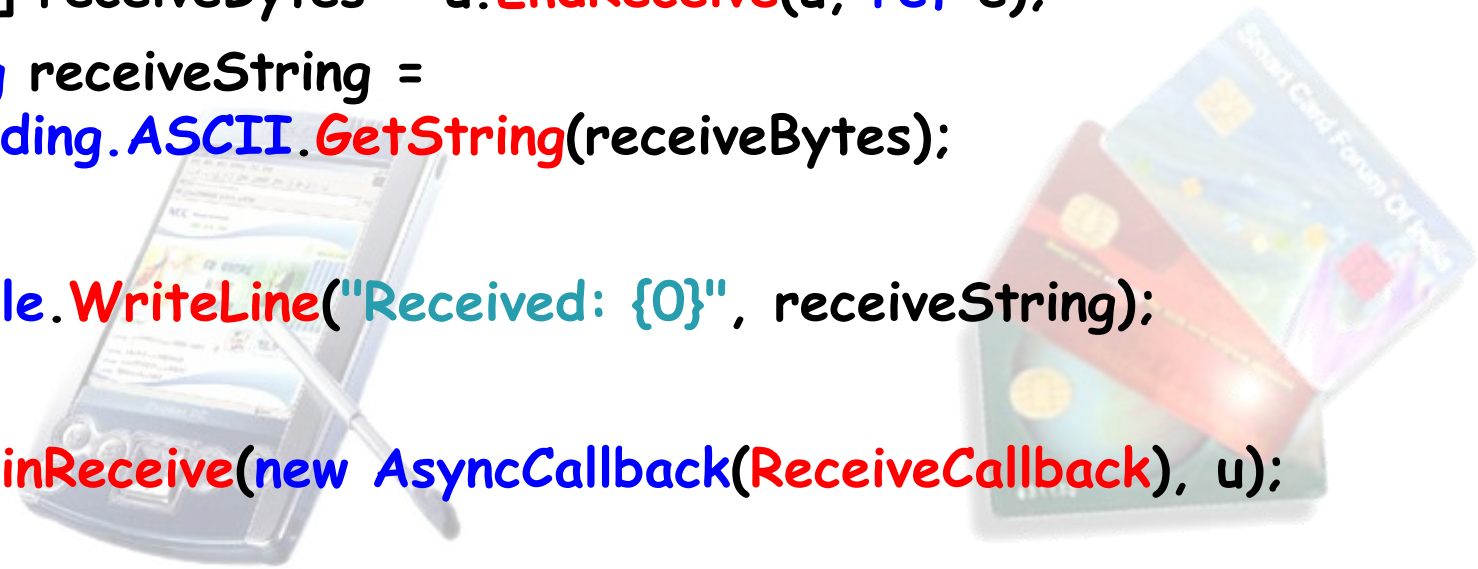
# Sử dụng **UdpClient**

```
public static void ReceiveCallback(IAsyncResult ar)
{
   UdpClient u = (UdpClient)ar.AsyncState;
   IPEndPoint e = new IPEndPoint(IPAddress.Broadcast, 0);

   Byte[] receiveBytes = u.EndReceive(u, ref e);
   string receiveString =
    Encoding.ASCII.GetString(receiveBytes);

   Console.WriteLine("Received: {0}", receiveString);

   u.BeginReceive(new AsyncCallback(ReceiveCallback), u);
}
```

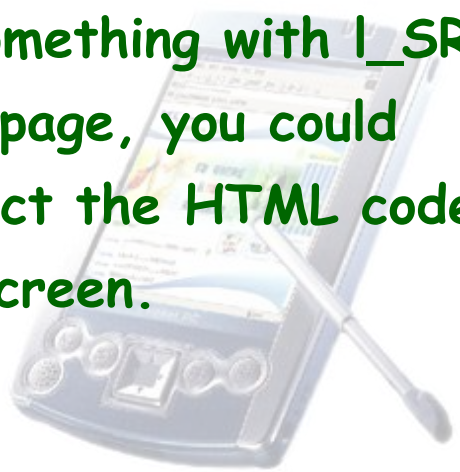


# Sử dụng **UdpClient**

```
IPAddress GroupAddress = IPAddress.Broadcast;
int GroupPort = 5000;
UdpClient sender = new UdpClient();
IPEndPoint groupEP = new IPEndPoint(GroupAddress, GroupPort);
byte[] bytes = Encoding.ASCII.GetBytes("Broadcast Message");
sender.Send(bytes, bytes.Length, groupEP);
sender.Close();
```

# Sử dụng **HttpWebRequest**

```
Uri      l_Uri = new Uri("http://www.myserver.com");
HttpWebRequest       l_WebReq =
    (HttpWebRequest)WebRequest.Create(l_Uri);
HttpWebResponse      l_WebResponse =
        (HttpWebResponse)l_WebReq.GetResponse();
Stream l_responseStream = l_WebResponse.GetResponseStream();
StreamReader l_SReader = new StreamReader(l_responseStream);
// Do something with l_SReader. For example, if you downloaded a
// Web page, you could
// extract the HTML code that came in the response and paint it on
// the screen.
```



# Kết nối bằng **IrDA** tại **Client**

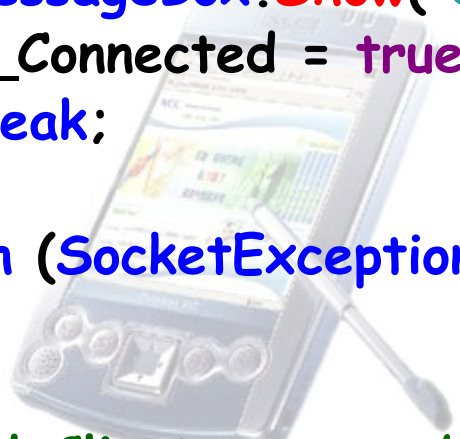
```
m_IrDAClient = new IrDAClient();
bool    l_foundAnyDevice = false;
int     MAX_DEVICES = 5;
// Find out who's out there to connect with...
IrDADeviceInfo[]     l_DevsAvailable =
        m_IrDAClient.DiscoverDevices(MAX_DEVICES);
// Show a MessageBox telling user every device we see out there
foreach (IrDADeviceInfo    l_devInfo    in     l_DevsAvailable)
{   l_foundAnyDevice = true;
    MessageBox.Show(l_devInfo.DeviceName,
                    "Discovered IrDA device");
    // Now try to connect to the devices, hoping it offers a service
    // named "IRDA_CHAT_SERVER"
```

# Kết nối bằng **IrDA** tại **Client**

```
try
   {
       // Assume that first device is offering a service that we
       // want
       IrDAEndPoint chatEndPoint = new IrDAEndPoint(
      l_DevsAvailable[0].DeviceID, "IRDA_CHAT_SERVER");
       m_IrDAClient.Connect(chatEndPoint);

       MessageBox.Show("Connected to chat server!", "Ready to chat"
       m_Connected = true;
       break;
    }
    catch (SocketException exc)   {    }
}

// m_IrdaClient can now be read from or written to.
```



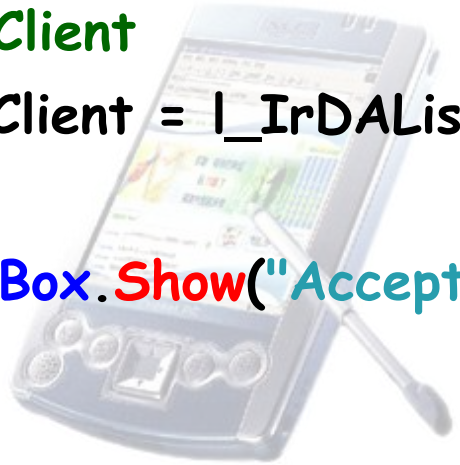
# Tạo kết nối **IrDa** tại **Server**

```
IrDAListener  l_IrDAListener = new
    IrDAListener("IRDA_CHAT_SERVER");

// Listen for anyone who wants to connect
l_IrDAListener.Start();

// And now pull the first queued connection request out as an
// IrDAClient
m_IrDAClient = l_IrDAListener.AcceptIrDAClient();

MessageBox.Show("Accepted a connection", "Ready to chat");
```



# Đọc dữ liệu từ **IrDAClient**

```
l_StreamReader = new
    StreamReader(this.m_IrDAClient.GetStream(),
        System.Text.Encoding.ASCII);

// Read a line of text and paint it into a GUI
this.lbInText.Items.Add(l_StreamReader.ReadLine());
l_StreamReader.Close();
```

# Ghi dữ liệu với **IrDAClient**

```
// Grab a reference to the stream in the m_IrDAClient and send the
// text to it.
StreamWriter          l_StreamWriter = new
    StreamWriter (this.m_IrDAClient.GetStream(),
        System.Text.Encoding.ASCII);
l_StreamWriter.WriteLine(this.txtSendText.Text);
l_StreamWriter.Close();
```