

🔗 1. Socket là gì?

Socket là một điểm cuối (endpoint) để gửi và nhận dữ liệu qua mạng. Nó có thể sử dụng các giao thức như **UDP** (không kết nối) hoặc **TCP** (có kết nối).

📦 2. Khai báo Socket

⚡ UDP

UDP dùng `UdpClient` đơn giản hơn:

```
csharp
CopyEdit
UdpClient udpClient = new UdpClient(); // hoặc UdpClient(localPort)
```

🔒 TCP

TCP dùng `Socket` hoặc `TcpClient`:

```
csharp
CopyEdit
Socket tcpSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
```

Hoặc:

```
csharp
CopyEdit
TcpClient tcpClient = new TcpClient();
```

🔗 3. Kết nối và thiết lập

⚡ UDP

UDP không cần `Connect()` bắt buộc:

- Gửi: `udpClient.Send(data, data.Length, ipRemote)`
- Nhận: `udpClient.Receive(ref remoteEP)`

🔒 TCP

- Client:

```
csharp
CopyEdit
tcpClient.Connect(ip, port);
NetworkStream stream = tcpClient.GetStream();
```

- **Server:**

```
csharp
CopyEdit
TcpListener listener = new TcpListener(IPAddress.Any, port);
listener.Start();
TcpClient client = listener.AcceptTcpClient();
NetworkStream stream = client.GetStream();
```

🔄 4. Cơ chế hoạt động:

⚡ UDP – Gửi và nhận không kết nối (Connectionless):

- Mỗi `Send()` là một gói độc lập (datagram).
- Nhận dữ liệu từ bất kỳ ai gửi đến port đang lắng nghe.
- Phù hợp truyền nhanh như video, game, nhưng không đảm bảo thứ tự, không kiểm tra lỗi.

🔄 Luồng:

```
csharp
CopyEdit
udpClient.Send(...) // gửi
udpClient.Receive(...) // nhận
```

🔒 TCP – Giao tiếp có kết nối (Connection-oriented):

- Dữ liệu được đảm bảo **đúng thứ tự**, không bị mất gói.
- Một bên làm `Server Listen`, bên kia `Connect`.

🔄 Luồng:

```
rust
CopyEdit
[Client]
Connect --> Send --> Receive --> Close

[Server]
Start --> Accept --> Receive --> Send --> Close
```

❑ 5. Các bước chuẩn để sử dụng Socket (TCP/UDP)

Bước	UDP	TCP
1. Tạo socket	<code>UdpClient</code>	<code>TcpClient</code> hoặc <code>Socket</code>
2. Ràng buộc port (nếu nhận)	<code>new UdpClient(port)</code>	<code>TcpListener.Start()</code>
3. Kết nối	không cần <code>Connect()</code>	<code>Connect(ip, port)</code>
4. Gửi dữ liệu	<code>Send(data, len, endpoint)</code>	<code>stream.Write(...)</code>
5. Nhận dữ liệu	<code>Receive(ref endpoint)</code>	<code>stream.Read(...)</code>
6. Đóng kết nối	<code>udpClient.Close()</code>	<code>tcpClient.Close()</code>

❑ 6. Tóm tắt điểm khác nhau giữa TCP và UDP

Tiêu chí	TCP	UDP
Kiểu giao thức	Có kết nối (connection-oriented)	Không kết nối (connectionless)
Đảm bảo dữ liệu	✓ Có	✗ Không
Tốc độ	❑ Chậm hơn	✂ Nhanh hơn
Ứng dụng phổ biến	Web, FTP, Email	Game, VoIP, video stream

📁 7. Kết luận

- **UDP phù hợp khi tốc độ quan trọng hơn độ tin cậy** (ví dụ: gửi file nhanh trong mạng LAN hoặc ứng dụng realtime).
- **TCP phù hợp khi cần chắc chắn dữ liệu đến nơi** (như chat, gửi file qua mạng Internet).