

Lập trình mạng

Chương 2: Lập trình Socket hướng kết nối

TS. TRẦN NGÔ NHƯ KHÁNH

Định nghĩa

- Socket - File mô tả dùng để tham chiếu đến các kết nối mạng
- Socket định nghĩa những đặc trưng sau:
 - Một kết nối mạng hay một đường ống dẫn để truyền tải dữ liệu
 - Một kiểu truyền thông như stream hay datagram
 - Một giao thức như TCP hay UDP
- Socket phải được gắn vào một địa chỉ mạng và một port trên hệ thống cục bộ hay ở xa

Lớp Socket trong .NET framework

➤ Phương thức tạo lập:

`Socket (AddressFamily, SocketType, ProtocolType)`

- AddressFamily: họ địa chỉ được dùng.
- SocketType: kiểu Socket.
- ProtocolType: kiểu Giao thức

➤ Ví dụ:

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

Một số lớp hỗ trợ

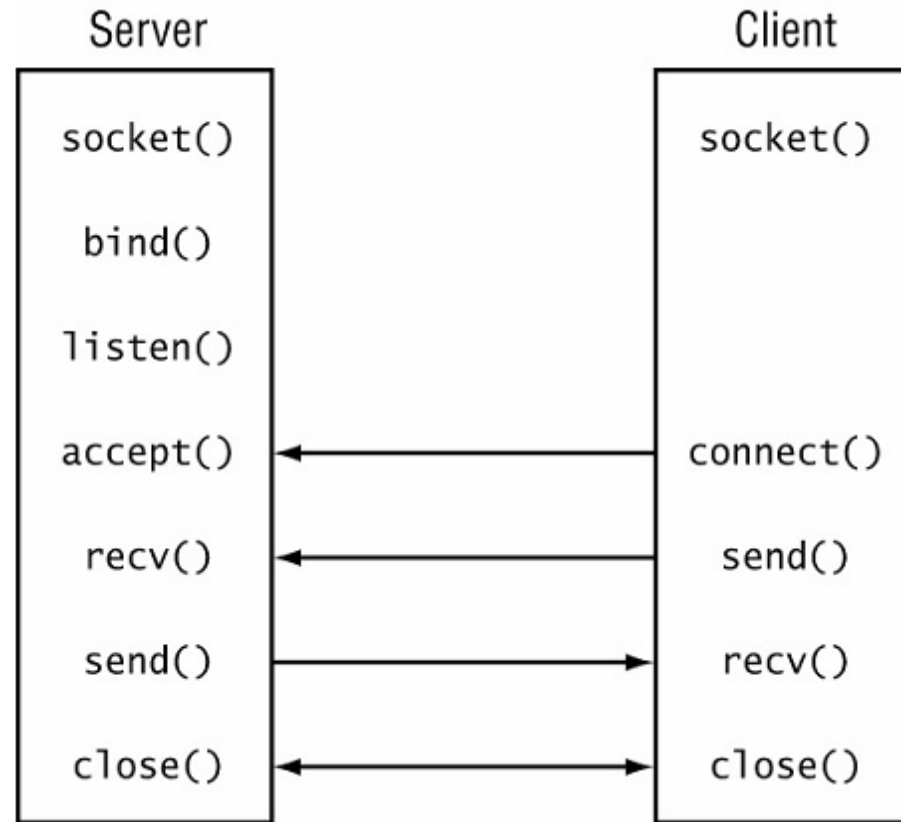
➤ **IPAddress:**

- Đối tượng dùng để mô tả một địa chỉ IP
- Sử dụng trong nhiều phương thức của Socket

➤ **EndPoint:**

- Đối tượng mô tả sự kết hợp của một địa chỉ IP và port.
- Dùng để gắn kết các Socket với các địa chỉ cục bộ hoặc các địa chỉ ở xa

Lập trình Socket hướng kết nối



Ví dụ: TCP Server và Client đơn giản

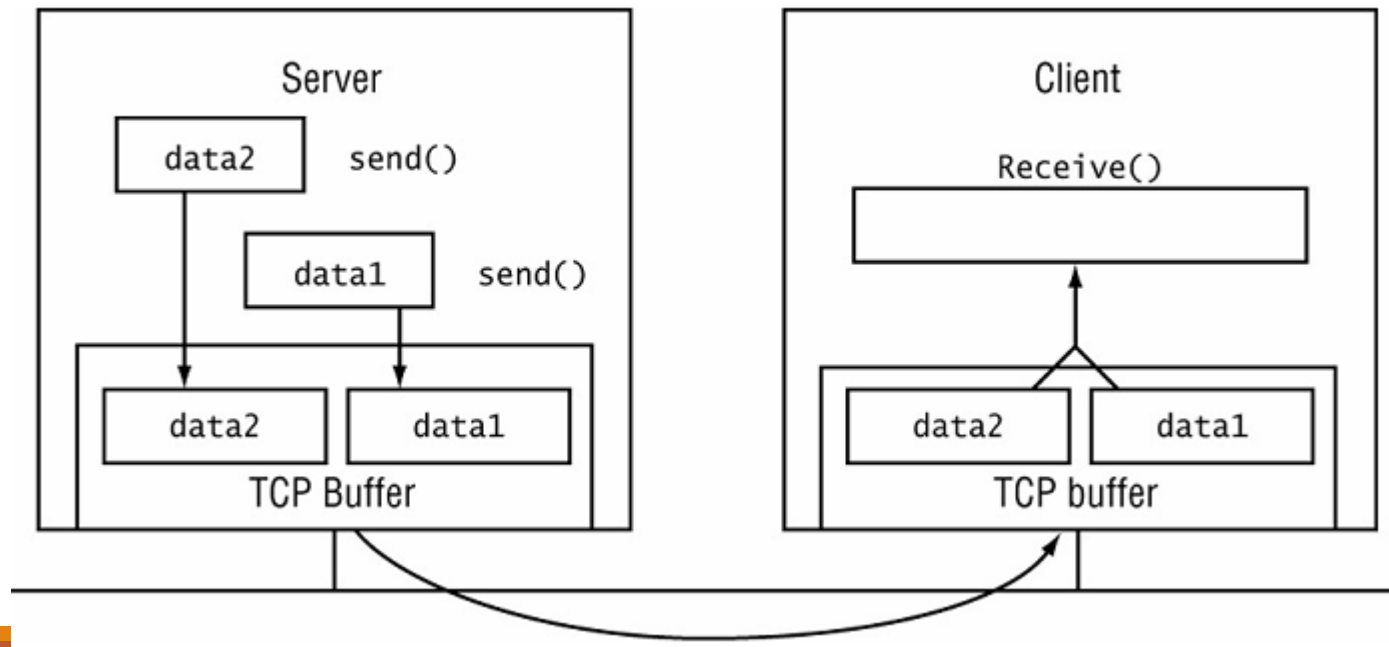
- Phương thức Send(): gửi dữ liệu, trả về số byte dữ liệu đã gửi
 - Buff : mảng các byte cần gửi
 - Offset: vị trí đầu tiên trong mảng cần gửi
 - Size: số byte cần gửi
 - SocketFlags: chỉ ra cách gửi dữ liệu trên Socket
- Phương thức Receive(): nhận dữ liệu từ socket, trả về số byte dữ liệu nhận được
 - Buff : mảng chứa dữ liệu nhận được
 - Offset: vị trí đầu tiên trong mảng nhận dữ liệu
 - Size: số byte cần nhận
 - SocketFlags: chỉ ra cách nhận dữ liệu trên Socket

Vấn đề với bộ đệm dữ liệu

- Số lượng dữ liệu được đọc bởi phương thức `Receive()` được điều khiển bởi hai yếu tố:
 - Số byte dữ liệu cần đọc được chỉ ra trong phương thức `Receive()`
 - Kích thước bộ đệm được chỉ ra trong tham số của phương thức `Receive()`

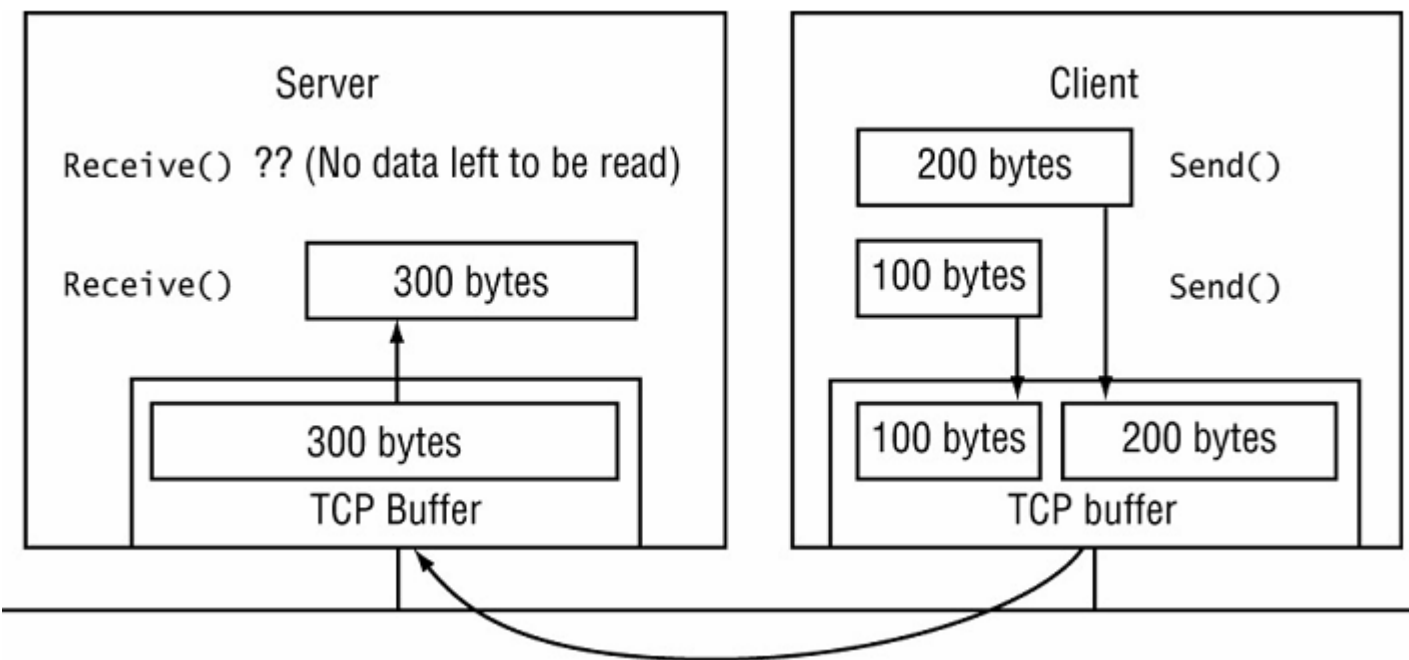
Xử lý với các bộ đệm có kích thước nhỏ

- TCP dùng bộ đệm để gửi và nhận dữ liệu.
- Dữ liệu chỉ được xóa khỏi bộ đệm khi nhận hồi báo thành công
- *Ví dụ: thay đổi kích thước buffer ở ví dụ trên thành 10 byte*



Vấn đề với các thông điệp TCP

- TCP không quan tâm đến biên dữ liệu
- Không đảm bảo mỗi phương thức Send() sẽ được đọc bởi một phương thức Receive()



Ví dụ: chương trình BadTCP

Giải quyết vấn đề thông điệp TCP

- Sử dụng thông điệp với kích thước cố định.
- Gửi kèm kích thước thông điệp với mỗi thông điệp.
- Sử dụng hệ thống đánh dấu để phân biệt thông điệp.

1) Truyền thông điệp có kích thước cố định

```
private static int SendData(Socket s, byte[] data)
{
    int total = 0;
    int size = data.Length;
    int dataLeft;
    int send;
    dataLeft = size;
    while (total < size)
    {
        send = s.Send(data, total, dataLeft, SocketFlags.None);
        total += send;
        dataLeft -= send;
    }
    return total;
}
```

1) Truyền thông điệp có kích thước cố định

```
private static byte[] ReceiveData(Socket s, int size)
{
    int total = 0;
    int receive;
    int dataLeft = size;
    byte[] data = new byte[size];
    while (total < size)
    {
        receive = s.Receive(data, total, dataLeft, SocketFlags.None);
        if (receive == 0)
        {
            data = Encoding.ASCII.GetBytes("Exit");
        }
        total += receive;
        dataLeft -= receive;
    }
    return data;
}
```

2) Gửi kèm kích thước thông điệp cùng với thông điệp

```
private static int SendVarData(Socket s, byte[] buff)
{
    int total = 0;
    int size = buff.Length;
    int dataleft = size;
    int sent;
    byte[] datasize = new byte[4];
    datasize = BitConverter.GetBytes(size);
    sent = s.Send(datasize);
    while (total < size)
    {
        sent = s.Send(buff, total, dataleft,
SocketFlags.None);
        total += sent;
        dataleft -= sent;
    }
    return total;
}
```

2) Gửi kèm kích thước thông điệp cùng với thông điệp

```
private static byte[] ReceiveVarData(Socket s)
{
    int total = 0;
    int recv;
    byte[] datasize = new byte[4];
    recv = s.Receive(datasize, 0, 4, 0);
    int size = BitConverter.ToInt32(datasize, 0);
    int dataleft = size;
    byte[] data = new byte[size];
    while (total < size)
    {
        recv = s.Receive(data, total, dataleft, 0);
        if (recv == 0)
        {
            data = Encoding.ASCII.GetBytes("exit ");
            break;
        }
        total += recv;
        dataleft -= recv;
    }
    return data;
}
```

Lớp `NetworkStream`

- Phương thức `Read()`: đọc các khối dữ liệu từ `NetworkStream`
`int Read(byte[] buffer, int offset, int size)`
 - Buffer: mảng các byte được đọc vào
 - Offset: vị trí bắt đầu để đọc vào trong bộ đệm
 - Size: số byte tối đa được đọc được
- Phương thức `Write()`: gửi các khối dữ liệu
`void Write(byte[] buffer, int offset, int size)`
 - Buffer: mảng các byte để ghi
 - Offset: vị trí bắt đầu để ghi trong bộ đệm
 - Size: số byte tối đa được ghi bắt đầu tại vị trí offset

Lớp StreamReader và StreamWriter

- Điều khiển việc đọc và ghi các thông điệp text từ mạng
- Có thể được khởi tạo từ một đối tượng NetworkStream

```
public StreamReader(Stream stream);  
public StreamWriter(Stream stream);
```

- Phương thức hay sử dụng: ReadLine() và WriteLine()