



**EAST ASIA UNIVERSITY
OF TECHNOLOGY**

LẬP TRÌNH MẠNG **(Network Programming)**

Network Stream

Nguyễn Anh Thơ
natho5578@gmail.com

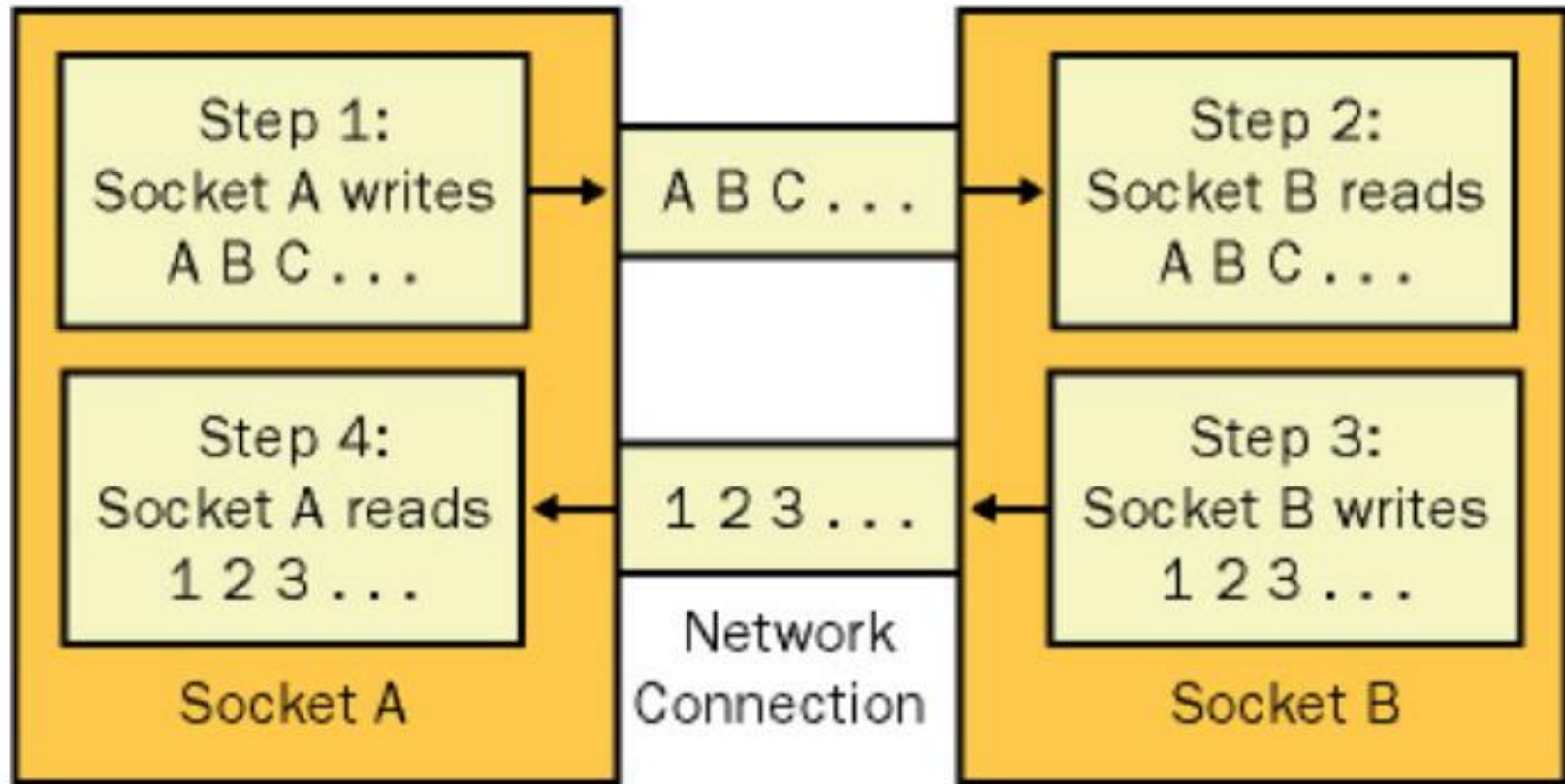
Nội dung

Tuần	Nội dung	
01	Network Stream là gì?	
02	Hoạt động Network Stream	
03	Quy trình tạo network stream	
04	Ví dụ	
05	Bài tập	

Network Stream làm gì?

- Giao tiếp giữa mạng và máy tính dựa trên lớp Sockets của System.Net
- Backing Store có khả năng kết nối với nhau
- Một ứng dụng có thể tạo ra một hoặc nhiều Socket để kết nối với nhau
- Hoạt động với các giao thức hướng kết nối như TCP/IP
- Không làm việc với mạng hoạt động dựa trên giao thức không kết nối (User Datagram Protocol - UDP)
- Stream-oriented sockets tạo ra các cặp kết nối ảo để truyền và nhận dữ liệu theo thứ tự thông qua giao thức TCP/IP

Hoạt động Network Stream



Bước 1. Tạo kết nối giữa các Socket (TCP)

Bước 2. Truyền dữ liệu (Nhận(đọc) – Gửi (Ghi))

Creating a Network Stream

Bước 1. Tạo kết nối Socket

Socket mySocket;

Bước 2. Khai báo và khởi tạo Network Stream

NetworkStream MyNetworkStream

= new NetworkStream(MySocket, true);

Bước 3. Thực hiện truyền dữ liệu

- Nhận dữ liệu (byte)
- Ghi dữ liệu (byte)

Bước 4. Đóng kết nối

MyNetworkStream.Close();

Đọc/ghi Network Stream

■ Ghi byte Network Stream

```
byte[] Buffer = new byte[1024];
```

```
MyNetworkStream.Write(Buffer, 0, Buffer.Length);
```

■ Đọc byte Network Stream

```
int Bytes = 0;  
byte[] Buffer = new byte[1024];  
  
do {  
    Bytes = MyNetworkStream.Read(Buffer, 0, Buffer.Length);  
    Console.WriteLine("Đọc " + Bytes.ToString() + " trên Socket.");  
} while (Bytes > 0);
```

Ví dụ

- Viết chương trình thiết lập kết nối giữa Sotket của hai máy tính thông qua port 5150 và địa chỉ lookup (127.0.0.1)
Thực hiện nhận và gửi dữ liệu(byte) giữa 2 máy tính kết nối

Chương trình máy nhận (Recieve – SV)

- Bước 1. Tạo kết nối Socket qua IP: 127.0.0.1 và port 5150
- Bước 2. Đọc dữ liệu nhận được từ máy Client truyền tới

Bước 1. Tạo kết nối Socket – TCP/IP

```
string ServerName = "127.0.0.1";
int Port = 5150;

Socket ServerSocket = null;
Socket ListeningSocket = null;
// Tạo Socket có dạng truy cập TCP/IP
ListeningSocket = new Socket(AddressFamily.InterNetwork,
                             SocketType.Stream, ProtocolType.IP);

// Khai báo và gán địa chỉ IP và cổng đích cần kết nối
IPEndPoint ListeningEndPoint = new IPEndPoint(IPAddress.Parse(ServerName), Port);

// Xác nhận địa chỉ đích
ListeningSocket.Bind(ListeningEndPoint);

// Đặt thời gian chờ kết nối
ListeningSocket.Listen(10);
Console.WriteLine("Awaiting a TCP connection on IP:"
    + ListeningEndPoint.Address.ToString()
    + " Port:" + ListeningEndPoint.Port.ToString() + "...");
// Xác lập kết nối thành công
ServerSocket = ListeningSocket.Accept();
Console.WriteLine("Received a connection - awaiting data...");
Console.WriteLine("Hello, World!");
ListeningSocket.Close();
```

Bước 2 – Tạo Network stream

```
// Let's create a network stream to communicate over the connected Socket.
NetworkStream ServerNetworkStream = null;

// Setup a network stream on the server Socket
ServerNetworkStream = new NetworkStream(ServerSocket, true);

int Bytes = 0;
byte[] Buffer = new byte[1024];

do {
    Bytes = ServerNetworkStream.Read(Buffer, 0, Buffer.Length);

    Console.WriteLine("We read " + Bytes.ToString() + " byte(s) from a peer socket.");

    for (int i = 0; i < Bytes; i++)
    {
        Console.WriteLine("The byte #" + i.ToString()
            + " contains " + Buffer[i].ToString());
    }
} while (Bytes > 0);

ServerNetworkStream.Close();
```

Chương trình máy gửi

- Bước 1. Thiết lập kết nối tới máy chủ có IP: 127.0.0.1 và port=5150
- Bước 2. Thực hiện ghi dữ liệu vào NetworkStream

Bước 1 – Thiết lập kết nối

```
string ServerName = "127.0.0.1";
int Port = 5150;
Socket ClientSocket = null;
Console.WriteLine("Creating a client Socket...");
ClientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);
IPEndPoint ServerEndPoint = new IPEndPoint(IPAddress.Parse(ServerName), Convert.ToInt16(Port));

ClientSocket.Connect(ServerEndPoint);
Console.WriteLine("Connect() is OK, connecting to " + ServerName + " at " + Port + "...");

// Let's create a network stream to communicate over the connected Socket.
NetworkStream ClientNetworkStream = null;

// Setup a network stream on the client Socket
Console.WriteLine("Instantiate NetworkStream object for communication...");
ClientNetworkStream = new NetworkStream(ClientSocket, true);
```

Bước 2 – Ghi dữ liệu lên Stream

```
byte[] Buffer = new byte[1];
Console.WriteLine("Writing/sending integers 0 - 10...");
for (int i = 0; i < 10; i++)
{
    Buffer[0] = (byte)i;
    ClientNetworkStream.Write(Buffer, 0, Buffer.Length);
}
Console.WriteLine("We wrote 10 bytes to the server.");

// Close the network stream once everything is done
Console.WriteLine("Closing the NetworkStream...");
ClientNetworkStream.Close();
```

Bài tập

- Thực hiện truyền và nhận dữ liệu dạng văn bản (ví dụ tên của bạn)

Composable Streams

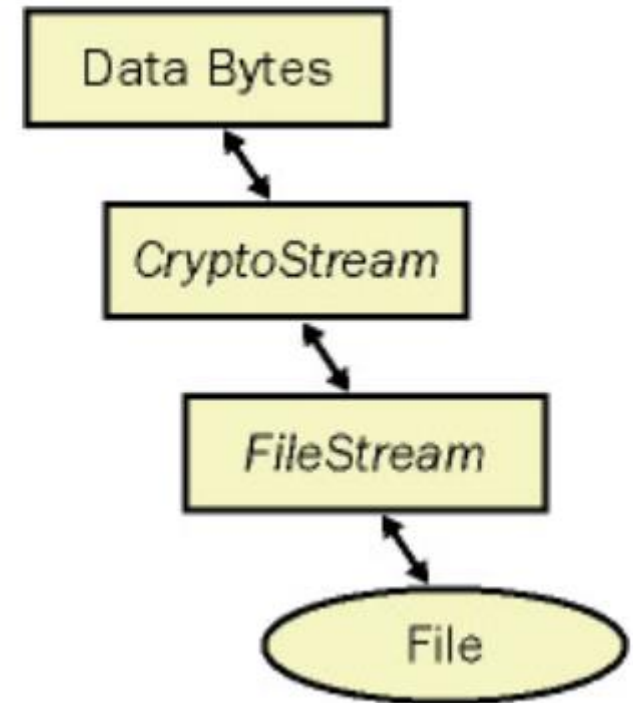
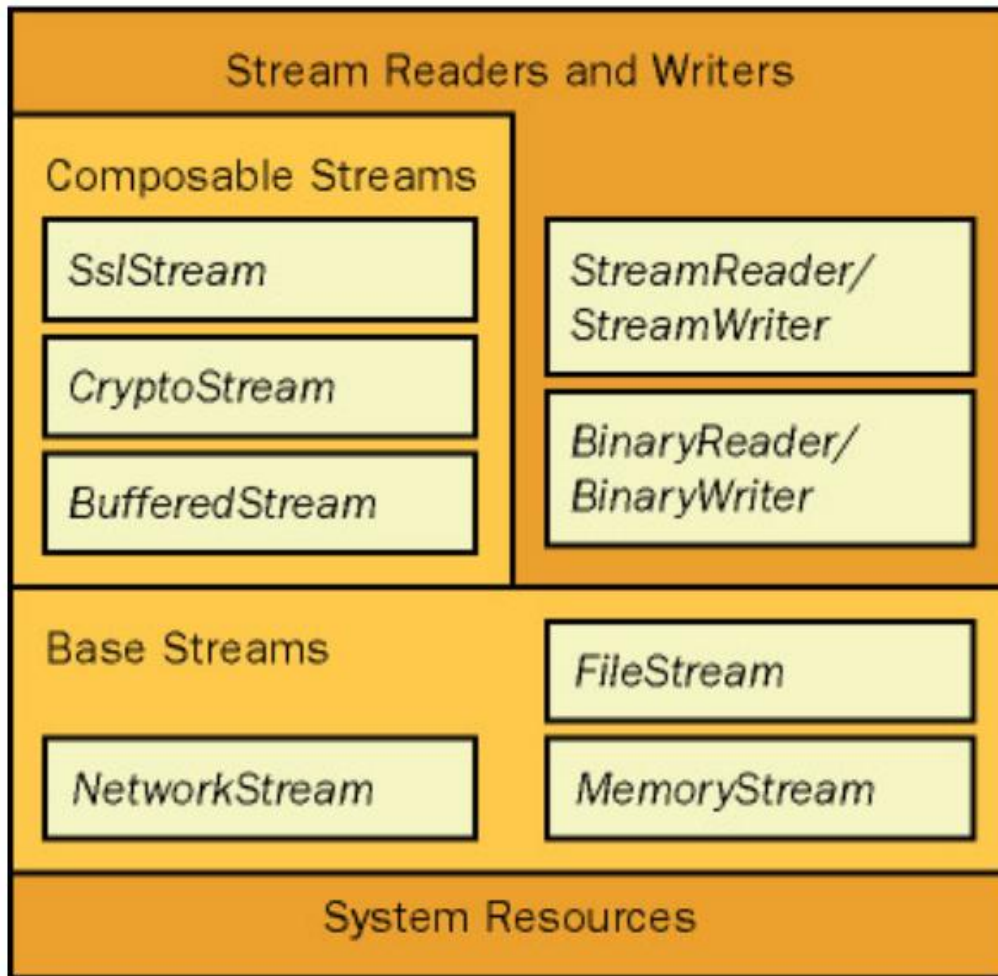


Figure 2-2: Stream relationship diagram

BufferedStream

- Tăng hiệu suất truy xuất I/O khi đọc/ghi trên stream khác
- Lớp này duy trì một bộ đệm bên trong để lưu vào bộ đệm hoặc thu thập dữ liệu và tại một thời điểm nào đó, nó sẽ tự động chuyển dữ liệu sang luồng tiếp theo hoặc giữ dữ liệu cho đến khi ứng dụng của bạn sẵn sàng đọc từ luồng.

BufferedStream

Bước 1. Thiết lập kết nối Socket (IP)

```
string ServerName = "127.0.0.1";
int Port = 5150;
Socket ClientSocket = null;

try
{
    ClientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);
    Console.WriteLine("Socket() is OK...");

    IPEndPoint ServerEndPoint = new IPEndPoint(IPAddress.Parse(ServerName), Convert.ToInt16(Port));
    ClientSocket.Connect(ServerEndPoint);
    Console.WriteLine("Connect() is OK...");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
    ClientSocket.Close();
    return;
}
```

BufferedStream

Bước 2. Tạo Network stream trên Client Socket

```
NetworkStream ClientNetworkStream = null;  
try  
{  
    try  
    {  
        // Setup a network stream on the client Socket  
        Console.WriteLine("Instantiate NetworkStream object...");  
        ClientNetworkStream = new NetworkStream(ClientSocket, true);  
    } catch (Exception ex)  
    {  
        Console.Out.WriteLine(ex.Message);  
        ClientSocket.Close();  
    }  
} catch (Exception ex)  
{  
    Console.WriteLine(ex.Message);  
    ClientNetworkStream.Close();  
    return;  
}
```

BufferedStream

Bước 3. Ghi (gửi) dữ liệu BuferedStream

```
BufferedStream ClientBufferedStream = null;
Console.WriteLine("Instantiate BufferedStream object...");
ClientBufferedStream = new BufferedStream(ClientNetworkStream);
try
{
    Console.WriteLine("Writing/sending data...");
    byte[] Buffer = new byte[1];

    for (int i = 0; i < 200; i++)
    {
        Buffer[0] = (byte)i;
        ClientBufferedStream.Write(Buffer, 0, Buffer.Length);
    }
    Console.WriteLine("We wrote 200 bytes one byte at a time to the server.");
} catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    // We are finished with the Stream so we will close it.
    ClientBufferedStream.Close();
}
```

CryptoStream Class

- Mã hóa đối xứng (symmetric)
- Mã hóa bất đối xứng (asymmetric)

Table 2-3: Available Cryptography Algorithms

Algorithm	Technique
Digital Signature Algorithm (DSA)	Asymmetric
RSA Security	Asymmetric
Data Encryption Standard (DES)	Symmetric
“Rivest’s Cipher” (RC2)	Symmetric
Rijndael	Symmetric
Triple Data Encryption Standard (TripleDES/3DES)	Symmetric

Bước 1 – Mã hóa - Khai báo và khởi tạo

```
byte[] DESKey = { 200, 5, 78, 232, 9, 6, 0, 4 };
byte[] DESInitializationVector = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

CryptoStream MyStreamEncrypter = null;
FileStream MyFileStream = null;
MyFileStream = new FileStream(@".\Jim.dat", FileMode.Create, FileAccess.Write);
Console.WriteLine("Jim.dat file created/opened successfully");

// Let's create a Symmetric crypto stream using the DES algorithm to encode
// all the bytes written to the file Jim.

Console.WriteLine("Instantiate DESCryptoServiceProvider & CryptoStream, "
    + "encrypting the file...");
DES DESAlgorithm = new DESCryptoServiceProvider();

MyStreamEncrypter = new CryptoStream(MyFileStream,
    DESAlgorithm.CreateEncryptor(DESKey, DESInitializationVector),
    CryptoStreamMode.Write);
```

Bước 2 – Mã hóa

```
byte[] MyByteArray = new byte[10];
for (short i = 0; i < MyByteArray.Length; i++)
{
    MyByteArray[i] = (byte)i;
}
MyStreamEncrypter.Write(MyByteArray, 0, MyByteArray.Length);
Console.WriteLine("Writing 10 bytes to the crypto stream...");

// Let's close the crypto stream now that we are finished writing data.
MyStreamEncrypter.Close();
```

Bước 1. Giải mã – Khai báo

```
CryptoStream MyStreamDecrypter = null;

FileStream MyFileStream2 = null;
MyFileStream = new FileStream(".\\Jim.dat", FileMode.Open, FileAccess.Read);
Console.WriteLine("Decrypted file opened successfully");

Console.WriteLine("Instantiate DESCryptoServiceProvider " +
    "& CryptoStream, decrypting the file...");

DES DESAlgorithm1 = new DESCryptoServiceProvider();

MyStreamDecrypter = new CryptoStream(MyFileStream,
    DESAlgorithm1.CreateDecryptor(DESKey, DESInitializationVector),
    CryptoStreamMode.Read);
```

Bước 2 – Giải mã

```
byte[] MyReadBuffer = new byte[1];
Console.WriteLine("Reading the decrypted file content...");

while (true)
{
    int BytesRead;
    BytesRead = MyStreamDecrypter.Read(MyReadBuffer, 0, MyReadBuffer.Length);
    if (BytesRead == 0)
    {
        Console.WriteLine("No more bytes to read");
        break;
    }
    Console.WriteLine("Read byte -> " + MyReadBuffer[0].ToString());
}
MyStreamDecrypter.Close();
```

QUESTION ?