Chương 3

Lập trình Socket không hướng kết nối (UDP)

Mục lục chương

Mô hình Client-Server không hướng kết nối

- Mô hình ứng dụng Client – Server không hướng kết nối
- Các thao tác phía server để xây dựng ứng dụng
- Các thao tác phía client để xây dựng ứng dụng
- Quá trình truyền tin giữa client và server
- Đóng socket

123

Mô hình Client-Server không hướng kết nối

1. Các thao tác để xây dựng ứng dụng client – server không hướng kết nối
   - Các thao tác phía server
   - Các thao tác phía client
   - Quá trình truyền nhận dữ liệu
   - Đóng kết nối
2. Mô hình ứng dụng client – server không hướng kết nối

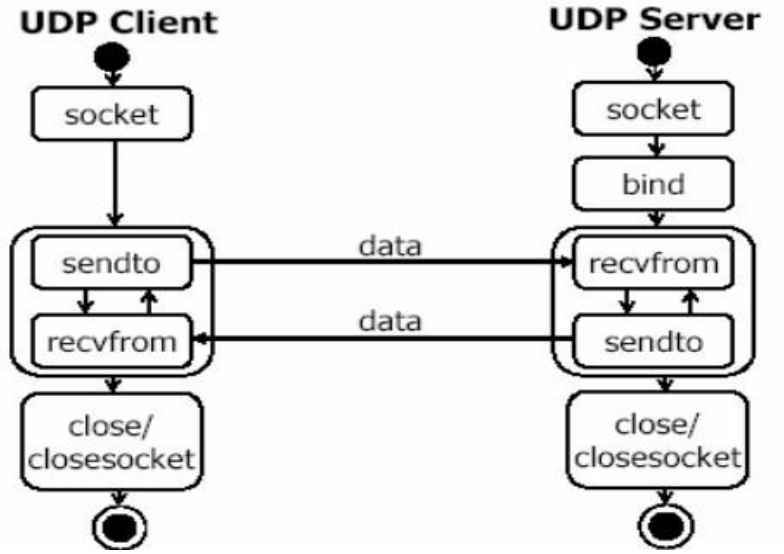
124

Các thao tác để xây dựng ứng dụng client – server không hướng kết nối

- Phía server:
  - -Tạo ra một Sockets
  - -Gắn Sockets đó với một địa chỉ cụ thể (binding)
- Phía Client:
  - - Tạo ra một Sockets
  - - Quá trình truyền nhận dữ liệu
  - - Đóng kết nối

125

Mô hình ứng dụng client – server không hướng kết nối



126

CÁC THAO TÁC PHÍA SERVER

1. Tạo một socket
2. Định danh cho socket (binding)

Tạo một socket

- Sử dụng Sockettype. Dgram và ProtocolType.Udp khi tạo socket không hướng kết nối.
- Ví dụ:

```
IPEndPoint ipep = new
IPEndPoint(IPAddress.Any, 9050);
Socket newsock =
Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
```

Định danh cho socket

- Việc này chỉ cần thực hiện đối với một máy, ta tạm gọi đó là máy chủ. Để định danh cho socket ta cũng sử dụng hàm bind.
- Ví dụ
- IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);
  Socket newsock =
- Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
  newsock.Bind(ipep);

129

CÁC THAO TÁC PHÍA CLIENT

1. Tạo ra một socket

Việc tạo ra một socket ở phía client hoàn toàn giống với phía server

2. Tiến hành truyền nhận dữ liệu

130

QUÁ TRÌNH TRUYỀN, NHẬN DỮ LIỆU GIỮA CLIENT VÀ SERVER

1. Quá trình truyền dữ liệu
2. Quá trình nhận dữ liệu

131

Quá trình truyền dữ liệu

- Để truyền dữ liệu ta sử dụng hàm sendto thay vì hàm send.
- Nguyên mẫu của hàm sendto như sau:
  - SendTo(byte[] *data*, EndPoint *Remote*)
  - SendTo(byte[] data, SocketFlags Flags, EndPoint *Remote*)
  - SendTo(byte[*data*], int Size, SocketFlags Flags, EndPoint *Remote*)
  - SendTo(byte[] *data*, int Offset, int Size, SocketFlags Flags, EndPoint *Remote*)

132

Quá trình nhận dữ liệu

- Để nhận dữ liệu ta sử dụng hàm ReceiveForm().
- Nguyên mẫu tương tự hàm SendTo() với một điểm khác là không có tham số EndPoint.
- Dạng hàm cơ bản là:
  - ReceiveFrom(byte[] data, ref EndPoint Remote)

133

Sử dụng Connect() trong UDP Client

- Chúng ta có thể nhận thấy là ứng dụng UDP sử dụng SendTo() và ReceiveFrom() khá phức tạp
- Lý do là vì ứng dụng UDP được xây dựng để có thể gửi và truyền dữ liệu đến bất kỳ máy nào
- Nếu chỉ truyền và nhận dữ liệu đến một máy cụ thể, ta có thể sử dụng hàm Connect()

134

Ví dụ UDPServer, UDPClient

- Ví dụ về UDPServer
- Ví dụ về UDPClient
- Ví dụ về sử dụng hàm Connect() trong UDP.

135

Phân biệt các thông điệp UDP

- Một trong những đặc điểm quan trọng của UDP
- là chúng bảo toàn ranh giới giữa các thông điệp
- Tuy nhiên UDP server, sau khi được tạo ra, thì có thể nhận thông điệp từ nhiều UDP Client.
- Vậy làm thể nào để UDP server phân biệt được
- các thông điệp đến từ các Client khác nhau?

136

Ví dụ Test UDPServer, UDPClient

- Ví dụ TestUdpSrvr
- Ví dụ TestUdpClient

Xử lý một số vấn đề trong truyền thông UDP

- Trong khi xử lý được vấn đề phân biệt ranh giới giữa các thông điệp, truyền thông UDP lại gây ra các vấn đề mới.
- Hai vấn đề quan trọng là
  - Vấn đề mất data
  - Vấn đề xác định các packet bị mất

Xử lý vấn đề mất data

- Một số ưu điểm của truyền thông TCP là:
  - Dữ liệu được đặt trong bộ đệm trước khi truyền và nhận
  - Mỗi lần gọi hàm Receive sẽ đọc một số lượng dữ liệu nhất định. Phần dữ liệu còn lại vẫn ở trong bộ đệm
  - Điều này hạn chế tối đa việc mất dữ liệu
- Tuy nhiên với truyền thông UDP thì những điều trên không được đảm bảo

139

Xử lý vấn đề mất data

- Trong truyền thông UDP
  - Không dùng bộ đệm
  - Toàn bộ dữ liệu đến sẽ được đưa đến hàm ReceiveFrom()
  - Nếu hàm ReceiveFrom() không nhận hết dữ liệu, thì dữ liệu sẽ bị mất

140

Ví dụ

- Ví dụ về BadUdpClient.cs.
- Ví dụ về BetterUdpClient.cs

141

Xử lý vấn đề mất các gói tin

- Một vấn đề khác trong truyền thông UDP là nó có khả năng mất các gói tin (packets)
- Lý do là vì trong truyền thông UDP không có cơ chế để biết rằng một gói tin khi được truyền đi có đến được đích hay không

142

Xử lý vấn đề mất các gói tin

- Để xử lý vấn đề mất gói tin thì ta cần cài đặt cơ chế báo nhận, có nghĩa là khi một gói tin nhận được từ phía máy khách thì nó sẽ báo lại cho máy gửi
- Nếu sau một thời gian máy gửi không nhận được báo nhận từ client thì nó sẽ gửi lại gói tin

143

Xử lý vấn đề mất các gói tin

- Có hai phương pháp để thực hiện cơ chế báo nhận
  - Sử dụng socket không đồng bộ cùng với đối tượng thời gian
  - Sử dụng socket đồng bộ cùng với thiết lập giá trị time-out

144

Sử dụng Socket time-out

- Hàm ReceiveFrom() là một hàm blocking, có nghĩa là khi gọi đến hàm này thì nó sẽ đứng chương trình cho đến khi hàm đó nhận được dữ liệu.
- Nếu vì một lý do nào đó mà dữ liệu không đến được thì chương trình sẽ bị treo

145

Sử dụng Socket time-out

- Để giải quyết vấn đề này ta cần dùng socket time-out
- Ví dụ:
  - server.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.ReceiveTimeout, 3000);
- Ví dụ về TimeoutUdpClient

146

Bắt ngoại lệ

- Ta biết rằng khi dùng socket time-out sẽ tạo ra ngoại lệ
- Vì vậy chúng ta phải bắt các ngoại lệ đó
- Ví dụ về ExceptionUdpClient

147

Xử lý vấn đề truyền lại dữ liệu

- Để cài đặt cơ chế truyền lại dữ liệu ta cần thực hiện các bước sau:
  - 1. Send a message to a remote host.
  - 2. Wait for an answer from the remote host.
  - 3. If an answer is received, accept it and exit the method with the received data and the size of the data.
  - 4. If no answer is received within a time-out value, increment a retry value.
  - 5. Check the retry value. If it is less than the number of retries desired, go to step 1 and start over. If it is equal, abort the retransmission attempt and report the results to the customer.

148

- 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

Chương 4

Sử dụng các lớp trợ giúp