Lập trình socket - UDPSocket

Lớp DatagramPacket

- Lớp DatagramPacket:
  - Tạo gói tin truyền thông với giao thức UDP.
  - Kế thừa trực tiếp từ lớp Object.

public final class DatagramPacket extends Object

  - Gói tin là đối tượng của lớp này chứa 4 thành phần: địa chỉ, dữ liệu truyền, kích thước của gói tin và port number.
  - Constructor cho gói tin gửi và nhận khác nhau:
    - ✓ DatagramPacket(byte[] buf, int length)
    - ✓ DatagramPacket(byte[] buf, int length, InetAddress address, int port)

Lớp DatagramPacket

- Lớp DatagramPacket:
  - Constructor tạo gói tin (datagram) **nhận** từ mạng:

    public DatagramPacket(byte[] inBuffer, int length)

    - ✓ inBuffer: bộ đệm nhập, chứa dữ liệu gói tin nhận
    - ✓ length: kích cỡ dữ liệu gói tin nhận, xác định bằng inBuffer.length

```
byte[] inBuffer = new byte[512];
DatagramPacket packet = new DatagramPacket(inBuffer, inBuffer.length);
```

Lớp DatagramPacket

- Lớp DatagramPacket:
  - Constructor tạo gói tin (datagram) gửi:

public DatagramPacket(byte[] outBuffer , int length, InetAddress destination, int port)

  - ✓ outBuffer: bộ đệm xuất chứa dữ liệu gửi
  - ✓ length: kích cỡ dữ liệu gói tin gửi, tính bằng byte, xác định bằng outBuffer.length
  - ✓ dest: địa chỉ nhận gói tin
  - ✓ port: cổng nhận gói tin tại đích (dest)

```
String s = "Hello SGU";
byte[] outBuffer = s.getBytes();
InetAddress dest = InetAddress.getByName("localhost");
int port = 1234;
DatagramPacket packet = new DatagramPacket(outBuffer, outBuffer.length, dest, port);
```

Lớp DatagramPacket

- Lớp DatagramPacket:
  - Các phương thức đối với gói tin nhận:
    - ✓ *public InetAddress getAddress()*: trả về đối tượng InetAddress của máy gửi trong gói tin nhận.
    - ✓ *public int getPort()*: trả về port number của máy gửi trong gói tin nhận.
    - ✓ *public byte[] getData()*: trả về dữ liệu chứa trong gói tin dạng byte
    - ✓ *public int getLength()*: trả về kích cỡ dữ liệu trong gói tin, tính theo byte.
  - Các phương thức đối với gói tin gửi: 4 phương thức bắt đầu bằng set….

Lớp DatagramSocket

- Lớp DatagramSocket:
  - Tạo socket truyền thông theo giao thức UDP, gửi nhận gói tin DatagramPacket
  - Sử dụng trên cả client và server, không phân chia rõ như TCPSocket
  - Constructor:
    - ✓ *public DatagramSocket () throws SocketException*: tạo ra UDPSocket sử dụng port number ngẫu nhiên.
    - ✓ *public DatagramSocket(int port) throws SocketException*: tạo UDPSocket với port number xác định, thường dùng với server trong mô hình client/server.

Lớp DatagramSocket

- Lớp DatagramSocket:
  - Các phương thức chính:
    - ✓ *public void send(DatagramPacket dp) throws IOException*: gửi gói UDP
    - ✓ *public void receive(DatagramPacket dp) throws IOException*: nhận gói UDP
    - ✓ *public void setSoTimeout(int timeout) throws SocketTimeoutException*
    - ✓ *public void close()*

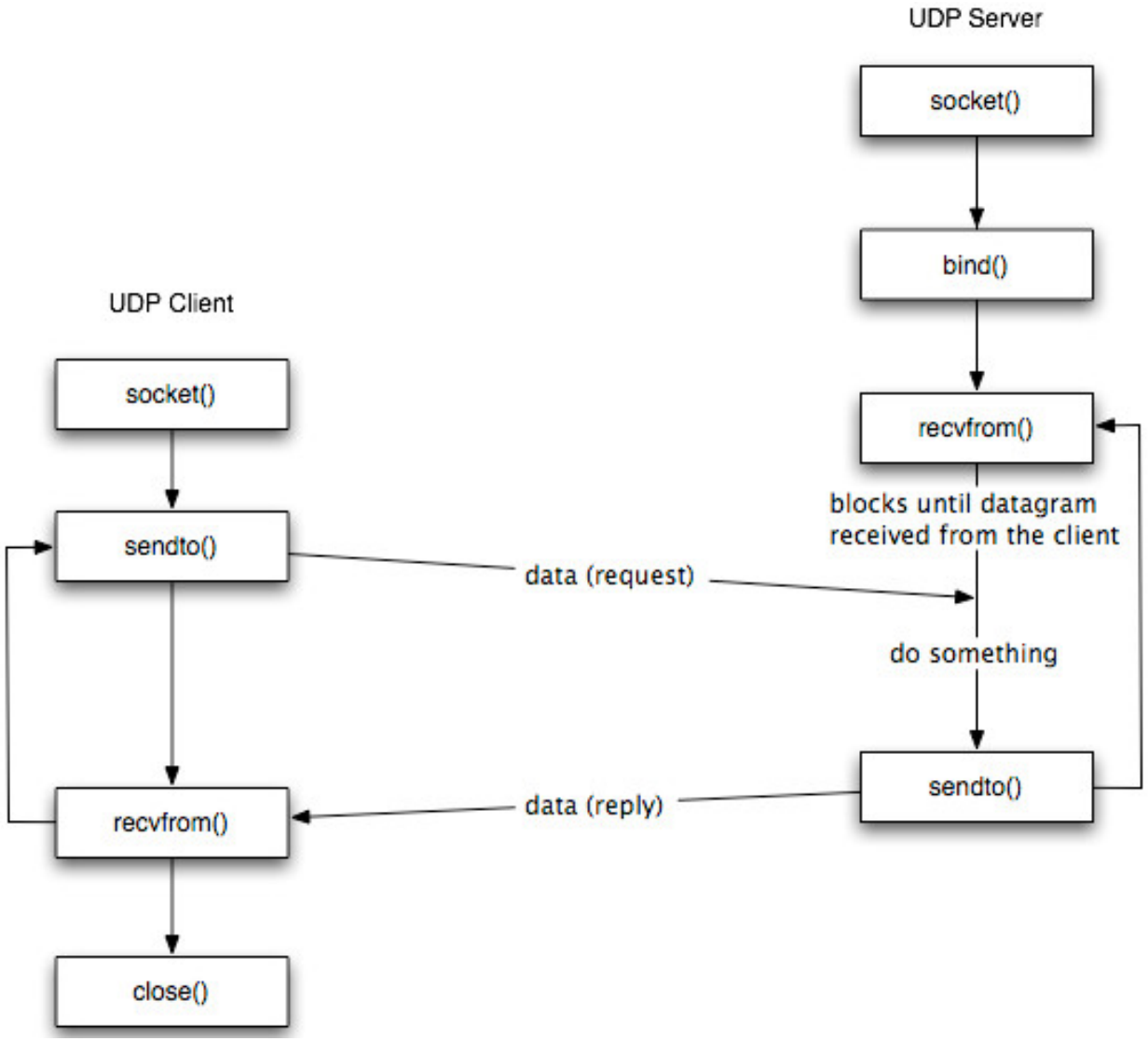
Ví dụ UDPSocket client

- Ví dụ 1: UDPSocket client QOTD

```
11          String hostname = "djxmmx.net";
12          int port = 17; // Port number for Quote of the Day (QOTD) Protocol
13          try {
14              InetAddress add = InetAddress.getByName(hostname);
15              DatagramSocket socket = new DatagramSocket();
16              // Sent UDP packet to djxmmx.net
17              DatagramPacket dpsend = new DatagramPacket(new byte[1], 1, add, port);
18              socket.send(dpsend);
19              // Received UDP packet
20              byte [] buffer = new byte[512];
21              DatagramPacket dpreceive = new DatagramPacket(buffer, buffer.length);
22              socket.receive(dpreceive);
23              String data = new String(buffer, 0, dpreceive.getLength());
24              System.out.println(data);
25              socket.close();
26          } catch (IOException e) { System.err.println(e);}
```

```
"Man can climb to the highest summits, but he cannot dwell there long."
 George Bernard Shaw (1856-1950)
```

Xây dựng ứng dụng UDPSocket



- Server/Client:
  - Tạo đối tượng DatagramSocket kèm port number
  - Tạo buffer in/out kiểu byte
  - Tạo gói tin DatagramPacket để gửi/nhận dữ liệu.
  - Gửi/nhận gói tin bằng phương thức receive()/send()
  - Quay lại bước trên hoặc đóng socket, giải phóng tài nguyên.

Xây dựng ứng dụng UDPSocket

- Lưu ý:
  - Server phải chạy trước client và trong trạng thái lắng nghe kết nối.
  - Client phải gửi DatagramPacket đến server trước
  - →Server lấy thông tin client trong DatagramPacket để phản hồi
- Ví dụ 1: chương trình gửi tin nhắn 1 chiều từ client → server sử dụng UDPSocket

Ví dụ 1: UDPSocket Server

```
 8      public static int buffsize = 512;
 9      public static int port = 1234;
10
11      public static void main(String[] args) {
12          DatagramSocket socket;
13          DatagramPacket dpreceive;
14          try {
15              socket = new DatagramSocket(1234);
16              dpreceive = new DatagramPacket(new byte[buffsize], buffsize);
17              while(true) {
18                  socket.receive(dpreceive);
19                  String tmp = new String(dpreceive.getData(), 0 , dpreceive.getLength());
20                  System.out.println("Server received: " + tmp + " from " +
21                          dpreceive.getAddress().getHostAddress() + " at port " +
22                          socket.getLocalPort());
23                  if(tmp.equals("bye")) {
24                      System.out.println("Server socket closed");
25                      socket.close();
26                      break;
27                  }
28              }
29          } catch (IOException e) { System.err.println(e);}
30      }
```

Ví dụ 1: UDPSocket Client

```
10      public static int destPort = 1234;
11      public static String hostname = "localhost";
12      public static void main(String[] args) {
13          DatagramSocket socket;
14          DatagramPacket dpsend;
15          InetAddress add; Scanner stdIn;
16          try {
17              add = InetAddress.getByName(hostname);   //UnknownHostException
18              socket = new DatagramSocket();           //SocketException
19              stdIn = new Scanner(System.in);
20              while(true) {
21                  System.out.print("Client input: ");
22                  String tmp = stdIn.nextLine();
23                  byte[] data = tmp.getBytes();
24                  dpsend = new DatagramPacket(data, data.length, add, destPort);
25                  System.out.println("Client sent " + tmp + " to " + add.getHostAddress() +
26                          " from port " + socket.getLocalPort());
27                  socket.send(dpsend);                 //IOExeption
28                  if(tmp.equals("bye")) {
29                      System.out.println("Client socket closed");
30                      stdIn.close();
31                      socket.close();
32                      break;
33                  }
34              }
35          } catch (IOException e) { System.err.println(e);}
36      }
```

Ví dụ 2: gửi tin nhắn hai chiều

- Ví dụ 2: gửi tin nhắn client ←→ server
  - Client gửi tin nhắn → server qua giao thức UDP
  - Server chuyển toàn bộ tin nhắn thành chữ hoa → client

Ví dụ 2: UDPServer

```
 8      public static int buffsize = 512;
 9      public static int port = 1234;
10      public static void main(String[] args) {
11          DatagramSocket socket;
12          DatagramPacket dpreceive, dpsend;
13          try {
14              socket = new DatagramSocket(1234);
15              dpreceive = new DatagramPacket(new byte[buffsize], buffsize);
16              while(true) {
17                  socket.receive(dpreceive);
18                  String tmp = new String(dpreceive.getData(), 0 , dpreceive.getLength());
19                  System.out.println("Server received: " + tmp + " from " +
20                          dpreceive.getAddress().getHostAddress() + " at port " +
21                          socket.getLocalPort());
22                  if(tmp.equals("bye")) {
23                      System.out.println("Server socket closed");
24                      socket.close();
25                      break;
26                  }
27                  // Uppercase, sent back to client
28                  tmp = tmp.toUpperCase();
29                  dpsend = new DatagramPacket(tmp.getBytes(), tmp.getBytes().length,
30                          dpreceive.getAddress(), dpreceive.getPort());
31                  System.out.println("Server sent back " + tmp + " to client");
32                  socket.send(dpsend);
33              }
34          } catch (IOException e) { System.err.println(e);}
35      }
```

Ví dụ 2: UDPClient

```
10      public static int destPort = 1234;
11      public static String hostname = "localhost";
12      public static void main(String[] args) {
13          DatagramSocket socket;
14          DatagramPacket dpsend, dpreceive;
15          InetAddress add; Scanner stdIn;
16          try {
17              add = InetAddress.getByName(hostname);  //UnknownHostException
18              socket = new DatagramSocket();          //SocketException
19              stdIn = new Scanner(System.in);
20              while(true) {
21                  System.out.print("Client input: ");
22                  String tmp = stdIn.nextLine();
23                  byte[] data = tmp.getBytes();
24                  dpsend = new DatagramPacket(data, data.length, add, destPort);
25                  System.out.println("Client sent " + tmp + " to " + add.getHostAddress() +
26                          " from port " + socket.getLocalPort());
27                  socket.send(dpsend);                //IOExeption
28                  if(tmp.equals("bye")) {
29                      System.out.println("Client socket closed");
30                      stdIn.close();
31                      socket.close();
32                      break;
33                  }
34                  // Get response from server
35                  dpreceive = new DatagramPacket(new byte[512], 512);
36                  socket.receive(dpreceive);
37                  tmp = new String(dpreceive.getData(), 0, dpreceive.getLength());
38                  System.out.println("Client get: " + tmp + " from server");
39              }
40          } catch (IOException e) { System.err.println(e);}
41      }
```

Tài liệu tham khảo

- https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/02.07.htm
- https://www.codejava.net/java-se/networking/java-udp-client-server-program-example
- https://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket