

Java UDP Socket Programming

□ UDP chatting Program

- UDP Echo Program
- **UDP Chatting Program**
 - Main
 - no-GUI vs. GUI
 - Socket Interface
 - Chatting Protocol

UDP Chatting programming

Goal: UDP를 사용하여 메시지를 주고 받는 Chatting progr.
(unreliable Packet Transmission)

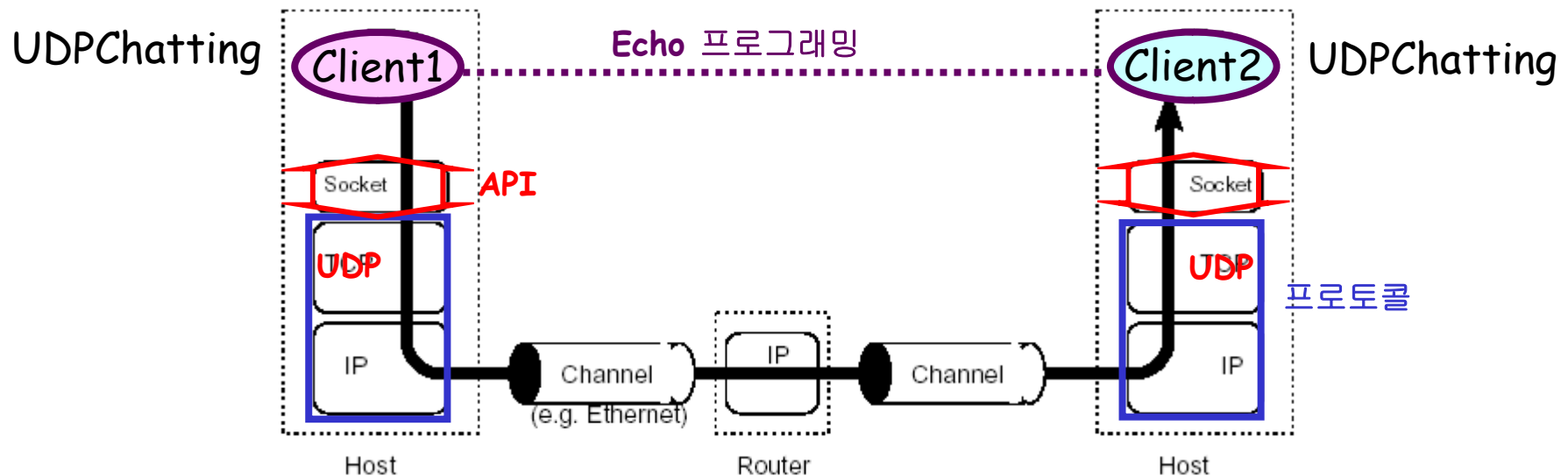


Figure 1.1: A TCP/IP Network

UDPChatting Program Structure

```
UDP chatting() {  
    DatagramSocket socket = new DatagramSocket();  
    rcThread = new RcvThread(socket, p); rcThread.start();  
    BufferedReader br = new BufferedReader(new  
        InputStreamReader(System.in));  
  
    while (true) {  
        String data = br.readLine();  
        socket.send (send_packet);  
    }  
}
```

```
class RcvThread extends Thread { ....  
    public void run() {  
        while (true) {  
            rcv_packet = new DatagramPacket /* */;  
            socket.receive(rcv_packet);  
            System.out.println("\n Receive Data : " + result);  
        }  
    }  
}
```

Client2



Operation Mode

□ UDPChatting server mode 실행

```
D:\User\jin\Util\Eclipse\UDPChatting>java -jar UDPChatting.jar
사용법: java UDPChatting localhost port or port

D:\User\jin\Util\Eclipse\UDPChatting>java -jar UDPChatting.jar 8000
Datagram my address on 0.0.0.0my port 8000
Input Data :
```

□ UDPChatting client mode 실행

```
D:\User\jin\Util\Eclipse>cd UDPChatting

D:\User\jin\Util\Eclipse\UDPChatting>java -jar UDPChatting.jar 127.0.0.1 8000
Datagram my address on 0.0.0.0my port 4426
Input Data : Hello
Input Data : _

D:\User\jin\Util\Eclipse\UDPChatting>java -jar UDPChatting.jar 8000
Datagram my address on 0.0.0.0my port 8000
Input Data :
Receive Data : Hello
```

UDPChatting Thread

```
try {
    if(myport==0) {socket = new DatagramSocket();}
    else {socket = new DatagramSocket(myport);}
    System.out.println("Datagram my address on "+socket.getLocalAddress().getHostAddress()+"my port "+socket.getLocalPort());

    rcThread = new RcvThread(socket);
    rcThread.start();

    DatagramPacket send_packet;// 송신용 데이터그램 패킷
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    while (true) {
        // 키보드 입력 읽기
        System.out.print("Input Data : ");
        String data = br.readLine();
        if (data.length() == 0){ // no char carriage return
            System.out.println("grace out call");
            break;
        } // else System.out.println("read line="+data);
        byte buffer[] = new byte[512];
        buffer = data.getBytes();// 스트링을 바이트 배열로 바꿈
        // 데이터 송신
        if((remoteaddr!=null)) {
            send_packet /* Fill in the Blank */ ;
            socket.send /* Fill in the Blank */;
        } else System.out.println("Server mode: unable to send until receive packet");
    }

} catch(IOException e) {
    System.out.println(e);
}
```

rcvThread Thread

```
class RcvThread extends Thread {
    DatagramSocket socket;
    boolean      sem=true;
    DatagramPacket rcv_packet;// 수신용 데이터그램 패킷

    RcvThread (DatagramSocket s) {
        socket = s;
    }
    public void run() {
        while (sem) {
            byte buff[] = new byte[100];
            rcv_packet /* Fill in the Blank by using buff[] */;
            try {
                socket /* Fill in the Blank to receive packets */;
                UDPChatting.remoteport = rcv_packet.getPort(); // 임의의 소켓에 대한 응답을 위해
                UDPChatting.remoteaddr = rcv_packet.getAddress(); // 임의의 소켓에 대한 응답을 위해
            } catch (IOException e) {
                System.out.println("Thread exception "+e);
            }
            String result = new String(buff);
            System.out.println("\n Receive Data : " + result);
        }

        System.out.println("grace out");
    }
    public void graceout(){
        sem=false;
    }
}

} // end ReceiverThread class
```

Mission 2: UDP Chatting ACKED

- Goal: UDP packet을 보내고 응답 패킷이 올때까지 무작정 기다림. (단, ACK는 아무 패킷이나 수신될 때까지 임)
- 동작:

Mission 3: UDP Chatting with Timeout Retransmission

- Goal: UDP packet을 보내고 응답이 없으면 일정 시간 후 다시 전송하기 위한 timeout 에 의한 재전송 기능. (단, 일정 회수 이상의 Timeout이 발생하면 전송 중지하고 keyboard 입력을 다시 기다림. 또 ACK는 아무 패킷이나 수신될 때까지 임)
- 동작:

BackUp

```

import java.io.*;
import java.net.*;
public class ChatJin implements Runnable, ActionListener {
....
    public ChatJin(int sport) {
        //Find Local IP Address & Name
        try {
            InetAddress myInet=InetAddress.getLocalHost();
            System.out.println("My InetAddress "+myInet.getHostAddress()+" name "+myInet.getHostName());
        } catch(UnknownHostException e) { System.out.println("InetAddress Error on Port "+8000+" Error code "+e);
            e.printStackTrace();    }
        try { //Listening //Test procedure로 임의의 datagramSocket을 생성할 경우 Inet이 0.0.0.0으로 설정됨, 여러개의 interface를 가진
            경우
                //선택의 여지에 따라 패킷을 받지 못하는 경우가 있음, 이를 방지해주기 위해 특정 IP 주소를 할당해야 할 필요가 있음
                //이것을 여러번 바꾸어도 못받는 LGIBM 노트북이 있음: 원인 불명 (추후 파악 필요)
                server = new DatagramSocket(sport,myInet);
                System.out.println("Datagram Server on "+server.getLocalAddress().getHostAddress()+" port "+server.getLocalPort());
                // DatagramSocket의 경우 port 번호와 InetAddress.getLocalHost()를 줄 경우 임의의 UnknownException을 지시해야 함 //
        } catch(UnknownHostException e) { System.out.println("Server Unknown Exception error"+e);
        } catch(SocketException e) { System.out.println("Server Socket Error on Port "+dport+" Error code "+e);
            e.printStackTrace();    }
        try { //Sending
            client = new DatagramSocket(dport,InetAddress.getLocalHost());
            System.out.println("Datagram Client on "+client.getLocalAddress().getHostAddress()+" port "+client.getLocalPort());
        } catch(IOException e) { System.out.println("Client Socket Error on Port "+dport+" Error code "+e);
            e.printStackTrace();    }
    }
    public static void main(String args[]) {
        if (args.length != 2) {System.out.println("사용법: java ChatJin port_number"); System.exit(0);}
        int arg_port = Integer.parseInt(args[0]);// 포트 번호1
        ChatJin m = new ChatJin(arg_port);
        m.go();
        Thread t = new Thread(m);
        t.start();
    }
    .....
}

```

```

public DatagramPacket EtherFrame(String infor) { //나중에 class로 만들 것, 지금은 단순한 서브루틴
    InetAddress inet = null; byte[] outdata=new byte[1024];
    byte[] da=new byte[6];      da[0]=0x0f;      System.arraycopy(da,0,outdata,0,6);      byte[] sa=new byte[6];      sa[0]=0x2f;
    System.arraycopy(sa,0,outdata,6,6);      byte[] ty=new byte[2];      ty[0]=0x03;      System.arraycopy(ty,0,outdata,12,2);
    byte[] llc=new byte[4];      ty[0]=0x04;      System.arraycopy(llc,0,outdata,14,4);
    String ip = (String) Smodel.getText(); //getText()과.getSelectedText()와의 차이는 return 값의 차이

    if(ip.equals("")) // ip=(String) Smodel.getSelectedText();
    { System.out.println("Server IP not setting");
      try { ip=InetAddress.getLocalHost().getHostAddress();
        } catch(UnknownHostException ee) { ee.printStackTrace(); }
    } // catch(NullPointerException ex)
    System.out.println("Server IP string"+ip+".");      addElement(ip.trim()); // set Server IP
    String up =(String) jcb.getSelectedItem();
    if(up!=null) {
        if(up=="\n") System.out.println("destination IP setting"+up+"."); // addElement((String) up.trim());
        else System.out.println("what string"+up+"."); //check error for up
    } /* if */
    String dportstring = (String) Smodel2.getText();
    if(dportstring.equals(""))
        dportstring = (String) "8000"; //default port number
    dport = (int) Integer.parseInt(dportstring); //string output -> convert string to int ? //System.out.println(dportstring);
    int len=infor.getBytes().length;      System.arraycopy(infor.getBytes(),0,outdata,18,len);
    byte[] crc=new byte[4];      crc[0]=0x05;      System.arraycopy(crc,0,outdata,18+len,4);
    try {      inet = InetAddress.getByName(ip);
    } catch(Exception ex) {      ex.printStackTrace();      }
    return(new DatagramPacket(outdata,len+18+4,inet,dport)); }

public void go() {
    try { f = new JFrame(InetAddress.getLocalHost().getHostAddress()); //WIndow Frame
    } catch(UnknownHostException e) { e.printStackTrace(); }
    f.addWindowListener(new WindowAdapter() { //Window icons: 최소화, 최대화, 끄기
        public void windowClosing(WindowEvent e) {
            if(client.isClosed()!=true) client.close();
            if(server.isClosed()!=true) server.close();
            System.exit(0);
        } //Socket.close를 호출하면 항상 socket의 exception이 발생함 따라서 현재 receive하는 socket이 있을 경우 exception이 발생함
    });
}

```

```

Container c = f.getContentPane(); //윈도우 frame내에 사용자 영역 (jp+jsp+jps)로 구성
    EtchedBorder eb = new EtchedBorder(); //각 패널의 겉 모양은 EtchedBoard로 구성 할 예정
    JPanel mp = new JPanel(); mp.setLayout(new GridLayout(1,3)); // 송신, 수신, 모니터링 (s+r+m)로 구성"Message Sent Received Monitoring"
    s = new JTextArea(); s.setEditable(false); // "Sent"전체 프레임의 중간에 있는 수신 메시지를 표시하는 프레임
    jsp = new JScrollPane(s);
    r = new JTextArea(); r.setEditable(false); // "Received"전체 프레임의 중간에 있는 수신 메시지를 표시하는 프레임
    jrp = new JScrollPane(r);
    m = new JTextArea(); m.setEditable(false); // "Monitered"수신된 메시지를 표시하는 프레임
    jmp = new JScrollPane(m);
    mp.add(jsp,"Left"); mp.add(jrp,"Center"); mp.add(jmp,"Right"); mp.setBorder(eb);

    JPanel jp2 = new JPanel(); // 윈도우 기본 틀내에 존재하는 상위 North 패널
    jp2.setLayout(new GridLayout(1,4)); //첫번째 프레임내 서브 프레임으로 사용 (I+jcb)로 구성
    JLabel l = new JLabel("Destination");
    model = new DefaultComboBoxModel(); jcb = new JComboBox(model); jcb.setEditable(true);
    JCheckBox checkBox = new JCheckBox("ARP"); //나중에 global 변수로 지정해서 다른 곳에서 활용가능하도록 함
    jp2.add(l,"West"); jp2.add(jcb,"Center"); jp2.add(checkBox,"East");
    jp2.setBorder(eb); // 프레임 주변의 음양으로 가는 줄을 표시

    JPanel Sjp = new JPanel(); // 윈도우 기본 틀내에 존재하는 상위 North 패널
    Sjp.setLayout(new GridLayout(1,4)); //첫번째 프레임내 서브 프레임으로 사용 (S+jcb)로 구성
    JLabel S = new JLabel("Server IP");
    Smodel = new JTextField(16);
    JLabel S2 = new JLabel(" UDP port");
    Smodel2 = new JTextField(4);
    Sjp.add(S); Sjp.add(Smodel); Sjp.add(S2); Sjp.add(Smodel2);
    Sjp.setBorder(eb); // 프레임 주변의 음양으로 가는 줄을 표시

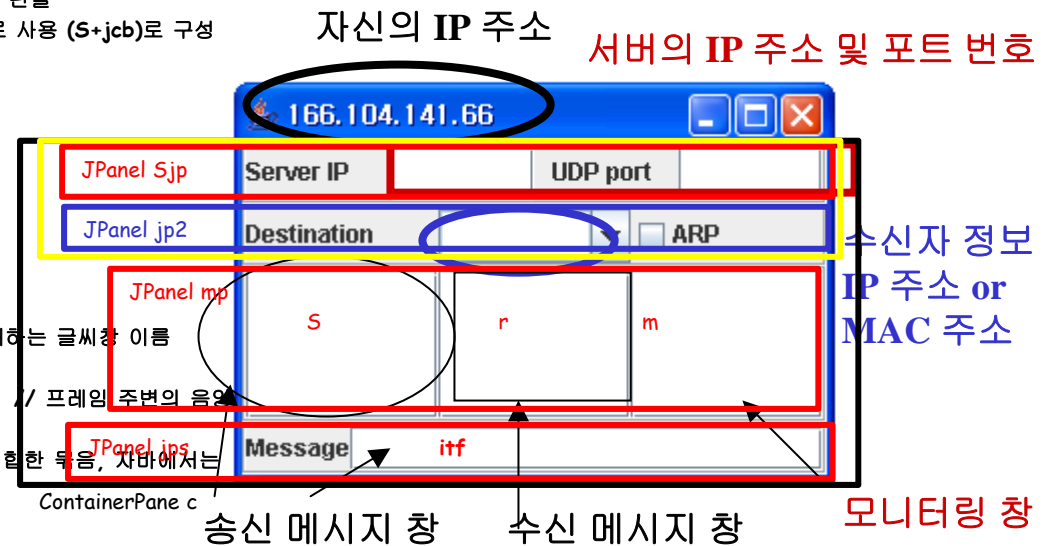
    JPanel jps = new JPanel(); // 전체 패널 하단의 패널 (Im+jtf)로 구성
    jps.setLayout(new BorderLayout());
    JLabel lm = new JLabel("Message"); //jps 패널 내의 글씨쓰는 옆에 존재하는 글씨창 이름
    jtf = new JTextField(); //jps 패널 중 글씨를 넣는 부분
    jps.add(lm,"West"); jps.add(jtf,"Center"); jps.setBorder(eb); // 프레임 주변의 음양으로 가는 줄을 표시

    JPanel jp = new JPanel(); //Server IP window와 Destination window를 합한 다음, 자바에서는
    jp.setLayout(new GridLayout(2,1)); //두개의 가로줄로 구성
    jp.add(Sjp); jp.add(jp2);

    c.add(mp); c.add(jp,"North"); c.add(jps,"South");

    jtf.addActionListener(this); //waiting for keyboard input
    f.setSize(300,200);
    f.setVisible(true);
}

```



```

//TextField의 내용을 packet에 저장 하여 보내는 구현...
public void actionPerformed(ActionEvent e) {
    //Key board input에 의한 message 작성
    String mssg = jtf.getText();
    s.append(">> " + mssg+"\n");
    m.append(count++ + ">> " + mssg+"\n");
    soutData = null;
    try {
//        inet = InetAddress.getByName(ip);
        soutData = EtherFrame(mssg);
        client.send(soutData);
        System.out.println("Send IP= "+soutData.getAddress()+" , port= " +soutData.getPort());
    } catch(Exception ex) {
        ex.printStackTrace();
    }
    jtf.setText("");
}

synchronized public void addElement(String ip) {
    System.out.println(ip);
    for(int i=0;i<model.getSize();i++) {
        if(model.getElementAt(i).equals(ip)) { //        model.setSelectedItem(ip);
            if(jb==null)        jb = jsp.getVerticalScrollBar();
            jb.setValue(jb.getMaximum());        return;        }
        }
    model.addElement(ip);
    if(jb==null)        jb = jsp.getVerticalScrollBar();
    jb.setValue(jb.getMaximum());
}

```

```

public void run() { //UDP packet을 Listen하는 Thread
    byte[] rmsg = new byte[1024];
    byte[] rsvmsg = new byte[1024];
    byte[] da=new byte[6];
    byte[] sa=new byte[6];
    byte[] ty=new byte[2];
    byte[] llc=new byte[4];
    byte[] crc=new byte[4];
    while(true) {
        sinData = new DatagramPacket(data,data.length);
        System.out.println("Listening...");
        try {
            server.receive(sinData);
        } catch(IOException e) {
            //I/O Exception if(sinData.getAddress().getHostName()+
            System.out.println("Packet received I/O Error..."+" "+e);
            e.printStackTrace();
            //break; //Break While loop
        }

        System.out.println("Packet received from..." +sinData.getAddress().getHostName()+" "+sinData.getPort());
        String addr = sinData.getAddress().getHostName();
        int pt =sinData.getPort();
        int len = sinData.getLength();
        rmsg=sinData.getData();
        System.arraycopy(rmsg,12,ty,0,2);
        System.arraycopy(rmsg,18,rsvmsg,0,len-18-4);
        String rsvData = new String(rsvmsg,0,len-18-4);
        System.out.println("Rsv Packet =..." +rsvData);
        //System.out.println("hostname =..." +addr+" port= "+pt+"Data Type 0x" +ty[1]+ty[0]+"dest addr=0x" +rmsg[5]+rmsg[4]+rmsg[3]+rmsg[2]+rmsg[1]+rmsg[0]);
        System.out.println("hostname =..." +addr+" port= "+pt+"Data Type 0x" +ty+"dest addr=0x" +rmsg[5]+rmsg[4]+rmsg[3]+rmsg[2]+rmsg[1]+rmsg[0]);
        r.append("[ " + addr + " ] " + rsvData + "\n");
        m.append(count++ + "[ " + addr + " ] " + rsvData + "\n");
        addElement(addr);
    }
}
}

```