Em ambas aplicações as mensagens foram enviadas e recebidas corretamente. Ou seja, sem dado corrompido.

O uso do UDP mostrou-se mais rápido! O que já era de se esperar, pois o pacote UDP não é verificado. Enquanto que no TCP, são feitas várias verificações.

Tempos aferidos na execução (dado em milisegundos):

	1	2	3	4	5	6	7	8	9	10	Média
TCP	0,06	0,073	0,074	0,077	0,066	0,062	0,067	0,074	0,065	0,073	0,0691
UDP	0,05	0,05	0,052	0,049	0,052	0,041	0,067	0,044	0,045	0,042	0,0492

Códigos:

Para executá-los basta rodar o arquivo server e depois o cliente correspondente.

------ Servidor UDP

```
public class ServerUDP {
    public static void main(String[] args) throws IOException{

        try{
            DatagramSocket server = new DatagramSocket(12345);

            byte[] buf = new byte[256];
            DatagramPacket packet = new DatagramPacket(buf, buf.length);
            server.receive(packet);

            String msg = "Enviando mensagem do ServidorUDP para ClienteUDP...";
            buf = msg.getBytes();

            //Enviando a mensagem
            InetAddress address = packet.getAddress();
            int port = packet.getPort();
            packet = new DatagramPacket(buf,buf.length, address, port);
            server.send(packet);

            server.close();

        } catch (SocketException ex) {
             Logger.getLogger(ServerUDP.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

------ Cliente UDP

```
public class ClientUDP {
   public static void main(String[] args) throws UnknownHostException, IOException{
      //calculando tempo
      long tempoInicial = System.currentTimeMillis();

      try{
         int port = 12345;
         DatagramSocket socket = new DatagramSocket();
         DatagramPacket packet;
         String host = "255.255.255.255";

         byte[] buf = new byte[1024];
         InetAddress address = InetAddress.getByName(host);
         packet = new DatagramPacket(buf, buf.length, address, port);
         socket.send(packet);

         //recebendo resposta do servidor
         packet = new DatagramPacket(buf, buf.length);
         socket.receive(packet);
         String msg = new String(packet.getData(), 0, packet.getLength());
         System.out.println("Mensagem recebida do servidor-->"+ msg + "<---");

         long tempoFinal = System.currentTimeMillis();
         System.out.printf("%.9f ms", (double)(tempoFinal - tempoInicial) / 1000L);
      }catch(IOException e){
         System.out.println("Erro: "+e.getMessage());
      }
   }
}
```

------ Servidor TCP

```
public class ServerTCP {

   public static void main(String[] args) throws IOException{

      try{
      // Abrindo porta 12345
      ServerSocket servidor = new ServerSocket(12345);
      Socket clientSocket = servidor.accept();
      System.out.println("Nova conexão com o cliente: " +
clientSocket.getInetAddress().getHostAddress());

      PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
```

```
        BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));

        out.println(in.readLine());
        }catch(IOException e){
            System.out.println("Erro: "+ e.getMessage());
       }
    }
}
```

------ Cliente TCP

```
public class ClientTCP {
    public static void main(String[] args){

        // Tempo Inicial
        long tempoInicial = System.currentTimeMillis();

        try {
            Socket client = new Socket("127.0.0.1", 12345);
            PrintWriter out = new PrintWriter(client.getOutputStream(), true);
            BufferedReader in = new BufferedReader(new
InputStreamReader(client.getInputStream()));

            out.println("Enviando mensagem do Cliente para o Servidor...");
            System.out.println("Mensagem recebida do Servidor --> " + in.readLine() +" <--");

            client.close();
            long tempoFinal = System.currentTimeMillis();
            System.out.printf("%.9f ms", (double)(tempoFinal - tempoInicial) / 1000L);

        }catch(Exception e){
            System.out.println("Erro: "+e.getMessage());
        }
    }
}
```