

华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2020 级

上机实践成绩：

指导教师：张召

姓名：张熙翔

学号：10205501427

上机实践名称：UDPping 程序

上机实践日期：2022/5/29

一、实验目的

了解系统命令 ping 用法
实现简单的 UDPping 程序

二、实验任务

ping 命令的简单使用
使用 IDEA 实现简单的 UDPping 程序

三、实验过程

Task: 在这个编程作业中，你将用 Java 编写一个客户 ping 程序。该客户将发送一个简单的 ping 报文，接收一个从服务器返回的对应 pong 报文，并确定从该客户发送 ping 报文到接收到 pong 报文为止的时延。该时延称为往返时延（RTT）。由该客户和服务器提供的功能类似于在现代操作系统中可用的标准 ping 程序。然而，标准的 ping 使用互联网控制报文协议（ICMP）（我们将在第 5 章中学习 ICMP）。此时我们将创建一个非标准（但简单）的基于 UDP 的 ping 程序。

你的 ping 程序经 UDP 向目标服务器发送 10 个 ping 报文。对于每个报文，当对应的 pong 报文返回时，你的客户要确定和打印 RTT。因为 UDP 是一个不可靠的协议，由客户发送的分组可能会丢失。为此，客户不能无限期地等待对 ping 报文的回答。客户等待服务器回答的时间至多为 1 秒；如果没有收到回答，客户假定该分组丢失并相应地打印一条报文。

实现思路：

调用 java JDK 中 java.net 包下的 DatagramSocket 和 DatagramPacket 类，可以实现对用户数据报文的控制，DatagramSocket 类用于创建接收和发送 UDP 的 Spcket 实例，调用 DatagramPacket 类用于处理报文，因为它可以将 Byte 数组、目标地址、目标端口等数据包装成报文或者将报文拆卸成 Byte 数组。通过随机函数模拟数据的丢失。通过调用 DatagramSocket 和 DatagramPacket 类来实现操作系统提供的 ping 命令的类似功能。最后对 10 个 ping 报文进行统计。

完整代码:

Server:

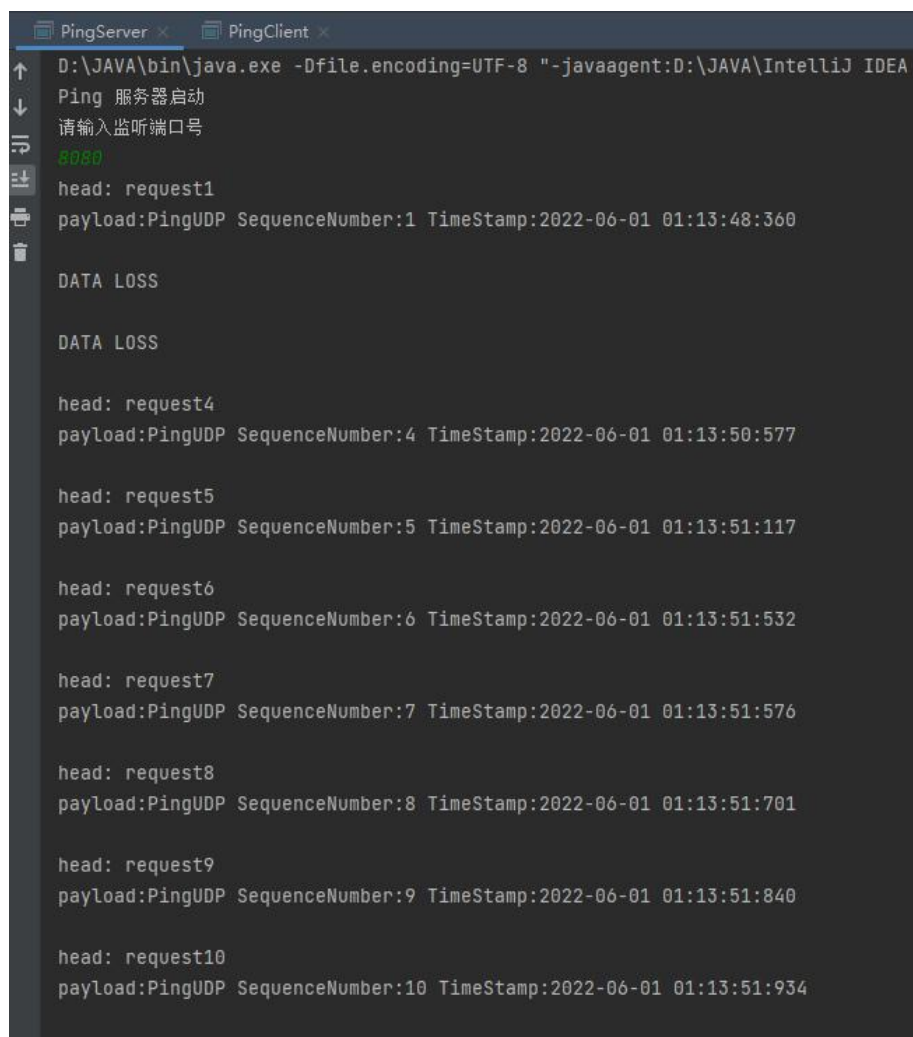
```
package UDPPing;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.nio.charset.StandardCharsets;
import java.util.Scanner;
//服务器
public class PingServer {
    public static void main(String[] args) throws Exception {
        System.out.println("Ping 服务器启动");
        System.out.println("请输入监听端口号");
        Scanner sc;
        sc = new Scanner(System.in);
        int initport = sc.nextInt(); //输入端口号
        DatagramSocket serverSocket = new DatagramSocket(initport);
        //根据端口生成 server 端 socket 实例
        byte[] receiveData = new byte[1024];
        //用于接收数据
        byte[] sendData;
        //用于发送数据
        while(true) {
            //死循环
            DatagramPacket receivePacket = new
DatagramPacket(receiveData, receiveData.length);
            //生成接收数据包实例
            serverSocket.receive(receivePacket);
            //用于接收数据
            String sentence = new String(receivePacket.getData());
            //将数据从 byte 数组转回 string
            InetAddress IPAddress = receivePacket.getAddress();
            //获得客户端 ip
            int port = receivePacket.getPort();
            //获得客户端 port
            long randomTime = (long) (Math.random()*1000);
            //生成随机数 模拟传输延迟
            Thread.sleep(randomTime);
            if(randomTime>650) {
                sentence="DATA LOSS \n";
                //如果随机数大于 650 则数据丢失
            }
            System.out.println(sentence);
            //显示数据
            sendData = sentence.getBytes(StandardCharsets.UTF_8);
            //请求数据转换 返回 client
            DatagramPacket sendPacket = new
DatagramPacket(sendData, sendData.length, IPAddress, port);
            //生成数据包
            serverSocket.send(sendPacket); //发送回 client
        }
    }
}
```

Client:

```
package UDPPing;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
import java.text.SimpleDateFormat;
import java.util.Date;
public class PingClient {
    public static void main(String[] args) throws Exception{
        Scanner sc;
        sc = new Scanner(System.in);
        System.out.println("请输入主机号");
        String address = sc.nextLine();
        System.out.println("请输入端口号");
        int port = sc.nextInt();
        long[] rtt = new long[10];
        //存储 rtt 并统计
        for(int i = 1; i <= 10; i++)
        { //模拟发送 10 条
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
hh:mm:ss:SS");
            //格式
            String sentence = "head: request"+i+"\n"+"payload:PingUDP
SequenceNumber:"
                +i+" TimeStamp:"+sdf.format(new Date())+"\n";
            DatagramSocket clientSocket = new DatagramSocket();
            //生成 client 的 socket 实例
            InetAddress IPAddress = InetAddress.getByName(address);
            //生成 ip 地址实例
            byte[] sendData;
            //接收数据
            byte[] receiveData = new byte[1024];
            sendData = sentence.getBytes();
            DatagramPacket sendPacket = new
DatagramPacket(sendData,sendData.length,IPAddress,port);
            Date sendBefore = new Date();//记录发送前的时间
            clientSocket.send(sendPacket);
            DatagramPacket receivePacket = new
DatagramPacket(receiveData,receiveData.length);
            clientSocket.receive(receivePacket);
            //接收从 server 发回的数据包
            Date receiveAfter = new Date();
            String modifiefSentence = new String(receivePacket.getData());
            rtt[i-1] = receiveAfter.getTime()-sendBefore.getTime();
            //计算 rtt
            System.out.println("rtt:"+rtt[i-1]);
            System.out.println(modifiefSentence);//显示数据
            clientSocket.close();
        }
        //统计
        long sum = 0;
        long max = 0;
        long min =rtt[0];
        byte loss = 0;
```

```
for(int i=0;i<10;i++){
    if(rtt[i]>650){
        loss++;
    }
    if(rtt[i]>max){
        max=rtt[i];
    }
    if(rtt[i]<min){
        min=rtt[i];
    }
    sum = sum+rtt[i];
}
byte received = (byte)(10-loss);
double rate= (double)loss/10;
System.out.println("统计结果如下: ");
System.out.println("10 packets transmitted,"+received+" packets
received,"
+rate*100+"% packet loss");
System.out.println("max rtt:"+max+" min rtt:"+min);
System.out.println("avg rtt:"+ (sum/10));
}
```

实验结果:



```
PingServer x PingClient x
D:\JAVA\bin\java.exe -Dfile.encoding=UTF-8 "-javaagent:D:\JAVA\IntelliJ IDEA
Ping 服务器启动
请输入监听端口号
8080
head: request1
payload:PingUDP SequenceNumber:1 TimeStamp:2022-06-01 01:13:48:360

DATA LOSS

DATA LOSS

head: request4
payload:PingUDP SequenceNumber:4 TimeStamp:2022-06-01 01:13:50:577

head: request5
payload:PingUDP SequenceNumber:5 TimeStamp:2022-06-01 01:13:51:117

head: request6
payload:PingUDP SequenceNumber:6 TimeStamp:2022-06-01 01:13:51:532

head: request7
payload:PingUDP SequenceNumber:7 TimeStamp:2022-06-01 01:13:51:576

head: request8
payload:PingUDP SequenceNumber:8 TimeStamp:2022-06-01 01:13:51:701

head: request9
payload:PingUDP SequenceNumber:9 TimeStamp:2022-06-01 01:13:51:840

head: request10
payload:PingUDP SequenceNumber:10 TimeStamp:2022-06-01 01:13:51:934
```

```
请输入主机号
localhost
请输入端口号
8080
rtt:462
head: request1
payload:PingUDP SequenceNumber:1 TimeStamp:2022-06-01 01:13:48:360

rtt:919
DATA LOSS

rtt:799
DATA LOSS

rtt:539
head: request4
payload:PingUDP SequenceNumber:4 TimeStamp:2022-06-01 01:13:50:577

rtt:415
head: request5
payload:PingUDP SequenceNumber:5 TimeStamp:2022-06-01 01:13:51:117

rtt:42
head: request6
payload:PingUDP SequenceNumber:6 TimeStamp:2022-06-01 01:13:51:532

rtt:124
head: request7
payload:PingUDP SequenceNumber:7 TimeStamp:2022-06-01 01:13:51:576

rtt:138
head: request8
payload:PingUDP SequenceNumber:8 TimeStamp:2022-06-01 01:13:51:701

rtt:93
head: request9
payload:PingUDP SequenceNumber:9 TimeStamp:2022-06-01 01:13:51:840

rtt:522
head: request10
payload:PingUDP SequenceNumber:10 TimeStamp:2022-06-01 01:13:51:934

统计结果如下:
10 packets transmitted,8 packets received,20.0% packet loss
max rtt:919 min rtt:42
avg rtt:488
|
```