****

**课程实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程编号** |  | **课程名称** | **计算机网络** |
| **实验学分** |  | **任课教师** | **叶松涛** |
| **学生学号** | **190320020** | **行政班级** | **信科1901** |
| **学生姓名** | **田章** | **联系方式** | **17680314099** |

**20 21 -2022学年度2学期**

**教务处制**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验** | | | | |
| **实验编号** | **实验5** | **实验名称** | **ICMP Ping程序** | |
| **指导教师** | **叶松涛** | **实验日期** | **2022.06.18** | |
| **实验同组人** | **未分组** |  |  | |
| 1. **实验目的和要求**   **实验目的：**在这个实验中，您将更好地理解因特网控制报文协议（ICMP）。您会学习使用ICMP请求和响应消息实现Ping程序。  **实验要求：**用python开发自己的Ping程序。程序将使用ICMP，但为了保持简单，将不完全遵循RFC 1739中的正式规范。只需要编写程序的客户端，因为服务器端所需的功能几乎内置于所有操作系统中。Ping程序能将ping请求发送到指定的主机，间隔大约一秒钟。每个消息包含一个带有时间戳的数据包。每个数据包发送完后，程序最多等待一秒，用于接收响应。如果一秒后服务器没有响应，那么客户端应假设ping数据包或pong数据包在网络中丢失（或者服务器已关闭）。 | | | | |
| 1. **实验设备、软件**   编辑器VScode  Python环境：3.10.1 | | | | |
| 1. **实验内容（步骤、计划、方案）**   import socket  import os  import sys  import struct  import time  import select  import binascii  ICMP\_ECHO\_REQUEST = 8  def checksum(strCheck):  *'''校验和'''*      csum = 0      countTo = (len(strCheck) / 2) \* 2      count = 0      while count < countTo:          thisVal = strCheck[count + 1] \* 256 + strCheck[count]          csum = csum + thisVal          csum = csum & 0xffffffff          count = count + 2      if countTo < len(strCheck):          csum = csum + strCheck[len(strCheck) - 1]          csum = csum & 0xffffffff      csum = (csum >> 16) + (csum & 0xffff)      csum = csum + (csum >> 16)      answer = ~csum      answer = answer & 0xffff      answer = answer >> 8 | (answer << 8 & 0xff00)      return answer  def receiveOnePing(mySocket, ID, timeout, destAddr):  *'''收到一个 Ping'''*      timeLeft = timeout      while 1:          startedSelect = time.time()          whatReady = select.select([mySocket], [], [], timeLeft)          howLongInSelect = (time.time() - startedSelect)          if whatReady[0] == []:  # Timeout              return "Request timed out."          timeReceived = time.time()          recPacket, addr = mySocket.recvfrom(1024)          # 从IP报文中获取ICMP头          header = recPacket[20:28]          header\_type, header\_code, header\_checksum, header\_packet\_ID, header\_sequence = struct.unpack(              "bbHHh", header)          if (header\_type != 0 or header\_code != 0 or header\_packet\_ID != ID or header\_sequence != 1):              return "Receive error."          timeLeft = timeLeft - howLongInSelect          if timeLeft <= 0:              return "Request timed out."          return timeLeft  def sendOnePing(mySocket, destAddr, ID):  *'''发送一个 ping'''*      # Header is type (8), code (8), checksum (16), id (16), sequence (16)      myChecksum = 0      # 创建一个校验和为0的虚拟头。      # struct——将字符串解释为压缩的二进制数据      header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, ID, 1)      data = struct.pack("d", time.time())      # 计算数据和虚拟头的校验和。      myChecksum = checksum(header + data)      # 得到正确的校验和，并放入头部      if sys.platform == 'darwin':          myChecksum = socket.htons(myChecksum) & 0xffff          # 将16位整数从主机字节顺序转换为网络字节顺序。      else:          myChecksum = socket.htons(myChecksum)      header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, ID, 1)      packet = header + data      mySocket.sendto(packet, (destAddr, 1))  # AF\_INET地址必须是元组，而不是str      # Both LISTS and TUPLES consist of a number of objects      # which can be referenced by their position number within the object  def doOnePing(destAddr, timeout):  *'''做一个 ping'''*      icmp = socket.getprotobyname("icmp")      # SOCK\_RAW是一个强大的套接字类型。更多信息请参见:http://sock-raw.org/papers/sock\_raw      mySocket = socket.socket(socket.AF\_INET, socket.SOCK\_RAW, icmp)      myID = os.getpid() & 0xFFFF  # 返回当前进程i      sendOnePing(mySocket, destAddr, myID)      delay = receiveOnePing(mySocket, myID, timeout, destAddr)      mySocket.close()      return delay  def ping(host, timeout=1):  *'''向 host 主机发送一个 ping 请求, 默认超时时长为 1s'''*      # timeout=1的意思是:如果一秒钟没有收到服务器的回复，客户端就会认为客户端的ping或者服务器的pong丢失了      dest = socket.gethostbyname(host)      print("Pinging " + dest + " using Python:")      print("")      # 向间隔大约一秒的服务器发送ping请求      for i in range(4):          delay = doOnePing(dest, timeout)          print(delay)          # 睡眠一秒钟          time.sleep(1)  ping("www.hutb.edu.cn") | | | | |
| 1. **实验原始记录、结果分析（可以附页）** | | | | |
| 1. **实验思考题及其解答**   **练习1.** 目前，程序计算每个包的往返时间，并逐个打印出来。修改此命令以符合标准ping程序的工作方式。你需要在所有ping完成后，报告最小，最大和平均RTT，以及数据包丢失率（百分比）。    **练习2.** 你的程序只能检测ICMP超时。修改Ping程序，解析ICMP响应错误代码，并向用户显示相应的错误结果。ICMP响应错误码示例有0：目标网络无法到达可达，1：目标主机无法到达。 | | | |
| 1. **实验评分**   **教师签名：**    **年 月 日** | | | |