

## 套接字编程作业 2-2: UDP ping 程序

本作业要求掌握使用 Python 进行 UDP 套接字编程的基础知识：如何使用 UDP 套接字发送和接收数据报，以及如何设置适当的套接字超时，并熟悉 Ping 应用程序及其在计算统计信息（如丢包率）中的作用。

在这个编程作业中，你需要研究一个用 Python 编写的简单的 ping 服务器程序，然后实现对应的客户端程序。该客户端程序将向 ping 服务器发送简单的 ping 报文，接收服务器返回对应的响应报文，并确定从该客户发送 ping 报文到收到响应报文为止的时延。该时延称为往返时延（RTT）。这些 ping 客户端程序和服务器程序提供的功能类似于现代操作系统中可用的标准 ping 程序功能，但是本作业的程序使用更简单的 UDP 协议，而不是标准 ping 程序使用的互联网控制消息协议（ICMP）。

### 任务 1、学习 ping 服务器代码

“UDPPingerServer.py”是 Ping 服务器的完整代码。

ping 服务器程序在一个无限循环中监听到来的 UDP 数据报。UDP 为应用程序提供了不可靠的传输服务，报文可能因为路由器队列溢出、硬件错误或其他原因而在网络中丢失。因此，ping 服务器程序使用一个随机整数来模拟网络丢包的影响。当数据报到达时，服务器将报文中的内容转为大写，并仅当生成的随机整数大于或等于 4 时才将其发送回客户端。

### 任务 2、编写 ping 客户端代码

根据 ping 服务器代码，编写一个 ping 客户端代码，向 ping 服务器发送简单的 ping 报文，接收服务器返回对应的响应报文，并确定从该客户发送 ping 报文到收到响应报文为止的时延。该时延称为往返时延（RTT）。

ping 客户端程序需具有如下功能：

1) 客户端程序的命令格式：

***client.py server\_address server\_port***

其中：***client.py*** 是客户端程序文件名；***server\_address*** 是 ping 服务器的 IP 地址，如果 ping 服务器和客户端运行在同一个主机中，则可以是该主机的 IP 地址或 127.0.0.1；***server\_port*** 是 ping 服务器正在监听的端口。

2) 客户端向服务器发送 10 个 ping 报文，每个报文的内容由以下格式的 ASCII 字符组成：

***Ping sequence\_number send\_time***

其中：***sequence\_number*** 是 ping 报文的序号（1~10）；***send\_time*** 是以年月日

时分秒表示发送 ping 报文的时间。

3) 因为 UDP 是不可靠的协议, 所以从客户端发送到服务器的分组可能在网络中丢失。因此, 客户端不能无限期地等待响应报文。客户端程序等待服务器回应的时间至多为 1 秒 (请查找 Python 文档, 以了解如何在 UDP 套接字上设置超时值)。

如果在 1 秒内没有收到服务器返回的响应报文, 则客户端程序应该假定分组在网络传输期间丢失, 并显示如下信息:

Request timed out

4) 客户端收到服务器返回的响应报文时将显示如下信息:

Reply from *server\_address*: *ping\_message*, RTT=*rtt\_value* ms

其中: *server\_address* 是发送响应报文的 ping 服务器的 IP 地址; *ping\_message* 是响应报文的内容; *rtt\_value* 是以毫秒 (ms) 为单位的往返时间。

5) 客户端要统计并显示丢包率 (百分比), 以及最小、最大和平均 RTT。

### 任务 3\*、编写 UDP Heartbeat 客户端和服务端 (选做)

UDP Ping 的另一个类似应用是 UDP Heartbeat (心跳)。心跳可用于检查应用程序是否已启动并运行, 并报告单向丢包。客户端在 UDP 数据报中将一个序列号和当前时间戳发送给正在监听客户端心跳的服务器。服务器收到 UDP 数据报后, 计算时差, 报告丢包 (若发生)。如果心跳数据报在指定的一段时间内丢失, 就可以假设客户端应用程序已经停止。实现 UDP Heartbeat 客户端和服务端, 你需要修改任务 2 中编写的 UDP ping 客户端代码和任务 1 中给出的 UDPPingerServer.py 代码。

### 作业提交要求

任务 1 和 2 必做, 任务 3 选做。

将以下内容压缩打包后提交, 压缩文件的命名规则为 “作业 2-2-学号-姓名”:

1) 任务 2 的完整 ping 客户端代码文件

2) 运行结果报告文件, 包含:

- Python 版本
- 任务 2 的 ping 客户端程序运行窗口截图

3) 可选提交: 任务 3 的 UDP Heartbeat 客户端和服务端代码文件各 1 份, 以及运行结果报告 1 份 (包含客户端和服务端程序运行窗口截图)

## Ping 服务器的完整代码（UDPPingerServer.py）

```
# We will need the following module to generate randomized lost packets import random
from socket import *
import random

# Create a UDP socket
# Notice the use of SOCK_DGRAM for UDP packets
serverSocket = socket(AF_INET, SOCK_DGRAM)
# Assign IP address and port number to socket
serverSocket.bind(('', 12000))

while True:
    # Generate random number in the range of 0 to 10
    rand = random.randint(0, 10)
    # Receive the client packet along with the address it is coming from
    message, address = serverSocket.recvfrom(1024)
    # Capitalize the message from the client
    message = message.upper()
    # If rand is less is than 4, we consider the packet lost and do not respond
    if rand < 4:
        continue
    # Otherwise, the server responds
    serverSocket.sendto(message, address)
```