

## 8 实验八 Socket 网络编程实验

### 8.1 实验目的

- 1) 掌握 Sockets 的相关基础知识，学习 Sockets 编程的基本函数和模式、框架。
- 2) 掌握 UDP、TCP 协议及 Client/Server 和 P2P 两种模式的通信原理。
- 3) 掌握 Socket 编程框架

### 8.2 实验内容

下面给出两个题目，供大家选择。

#### 8.2.1 实现一个简单的聊天程序

1) 功能类似于 QQ 的简单聊天程序，不一定需要图形界面；具体的通信模式不限，可以是单纯的客户机/服务器（C/S）模式，也可以是混合 C/S 和 P2P 模式；具体协议不限，可以是基于 TCP 的实现，也可以是混合 TCP 和 UDP 协议的实现。

2) 基本功能包括：A. 验证用户登录；B. 两个用户的文字聊天；C. 用户之间传输文件，包括二进制的大文件（比如 100MB）。

3) 高级功能包括：A. 支持 NAT 穿透；B. 离线文件和断点续传的支持；C. 语音聊天。

#### 8.2.2 实现基于 TCP 协议的 FTP 客户端和服务端程序

1) 基于标准的 FTP 协议，分析 TCP 的粘包问题并给出解决思路。

2) 基本功能包括：A. 用户和口令的认证；B. 目录的各种操作（进入、创建、列表）；C. 文件上传和下载；

3) 高级功能包括：A. PASSIVE 模式；B. 断点续传。

4) 测试：在本地运行 FTP 客户端程序，测试网络中某个可用的 FTP 服务器；再将 FTP 服务器程序放在云服务器上，在本地运行客户端程序，测试各项功能。

## 8.3 实验原理

### 8.3.1 熟悉 TCP, UDP 协议原理

略。

### 8.3.2 Socket 网络编程原理

下面简单介绍互联网的 Client/Server 模式的工作原理，以 TCP 服务器为例说明，UDP 服务器略有不同。客户端也是如此。

#### 1) 服务器

服务器先创建一个套接字 (Socket)，并将该套接字和特定端口绑定，然后服务器开始在此套接字上监听，直到收到一个客户端的连接请求，然后服务器与客户端建立连接，连接成功后和该客户端进行通信 (相互接收和发送数据)，进行用户信息验证，并返回验证信息。最后，服务器和客户端断开连接，继续在端口上监听。

#### 2) 客户端

客户端创建一个套接字，里面包含了服务器的地址和端口号，客户端的端口号由系统自动分配，不需要指明。和服务器建立连接，如果连接成功则Socket创建成功。然后客户端发送用户名和密码，等待验证。通信结束后主动断开连接，释放资源。

## 8.4 实验要求

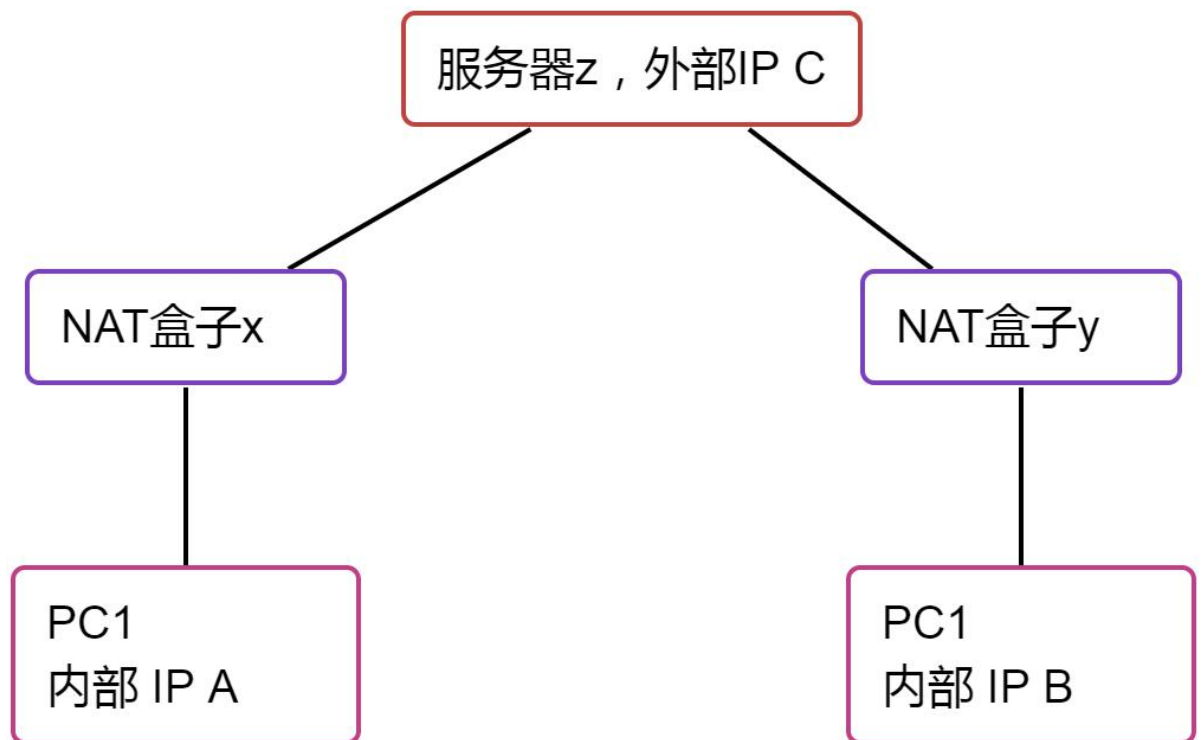
- 1) 撰写实验报告；
- 2) 编程语言：建议 C/C++，JAVA 或 Python。

## 8.5 实验环境和分组

1) 每 2 位同学一组，共同编写程序 (可以一人写客户端程序，另外一位写服务器程序)。

2) 进行调试和测试，写出简单的测试报告。服务器程序和客户端程序不在同一局域网内。

典型的网络拓扑如下，服务器 z 一般选云服务器，比如腾讯云、华为云等的服务器：



## 8.6 实验步骤

- 步骤 1: 阅读或设计网络通信系统的协议
- 步骤 2: 设计程序的人机界面, 设计程序框架
- 步骤 3: 编写 server 端和 client 端程序
- 步骤 4: 启动分组捕获软件 (如 Wireshark、Tcpdump 等), Client 端和 Server 端实现互联通信, 测试并分析捕获的分组, 分析协议实现情况,
- 步骤 5: 根据测试和分析结果, 编写测试报告。