# 1概述

本软件在实现了TCP通信和udp通信，客户端可以自主选择是通过什么协议发送信息；对于服务器端，它可以接收多个客户端的连接请求，并能够区分不同的客户端，并将所有的链接客户端显示出来；连接成功后，服务器能接收客户端发送过来的文本信息并显示，客户端在本地能够显示自己发送的信息，信息的发送模式为TCP和UDP。

* + 1. **开启服务器**

用户在服务器端输入端口号，点击“开启服务”按钮，开启服务器。

如果端口和ip输入有误，则会报错。

* + 1. **TCP连接**

用户在客户端输入服务器的IP地址和端口号，点击“连接服务器”，待左下角显示“连接成功”时，完成TCP连接。

* + 1. **发送信息**

采用TCP发送信息时首先要完成服务器开启和TCP连接，之后用户可在客户端的文本编辑区输入信息，点击“发送”。

* + 1. **通过udp**

采用UDP发送信息时只需打开服务器，不需要进行连接，然后在客户端的文本编辑区输入信息，点击“发送”。通过udp发送时，必须勾选udp复选框，这样客户端则会通过udp协议发送信息，否则客户端是通过tcp进行连接后发送信息，则会报错。

1. **开发目标**

**服务器端：**

服务器端可以接收多个客户端的连接请求，并能够区分不同的客户端，显示并打印已连接的客户端，如果收到信息则需要将信息打印数来；

打印信息时不区分udp或者tcp，通过sockted收到信息后，如果可以peername则是有链接的tcp，用recv收信息，否则使用recvfrom收取udp的信息。

**客户端：**

在客户端的文本编辑区输入信息，点击“发送”。通过udp发送时，必须勾选udp复选框，这样客户端则会通过udp协议发送信息，否则客户端是通过tcp进行连接后发送信息，则会报错。

1. **参考文档**

《软件需求分析说明书》

《Windows程序设计（王艳平，第二版）》

《Visual C++\_6.0程序设计从入门到精通》

1. **开发工具和平台**

操作系统：Windows 10

工具：Visual Studio 2013

Visual C++ 6.0

1. **测试工具**

Visual Studio 2013

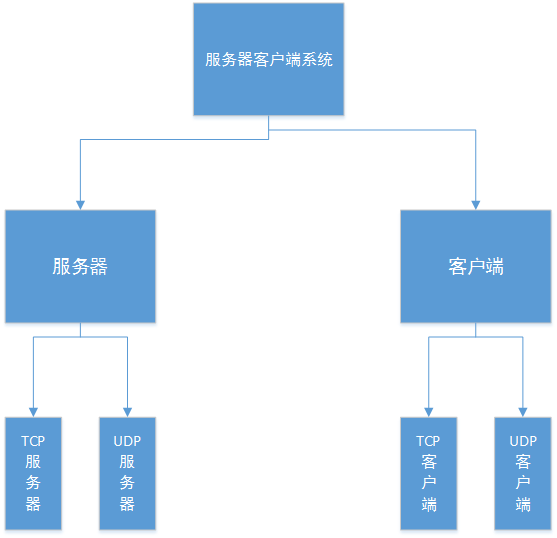
Visual C++ 6.0

1. **程序结构**

这里要描述软件系统的总体结构，可以使用结构图、层次分解图或包图来描述，并应说明系统结构划分的原则（例如，基于标准、协议所规定的体系结构，来自于分析模型的方案，或者基于原有体系结构的限制）。对于使用分析模型的体系结构，应说明分析类的职责及相互关系。

服务器和客户端是通过TCP协议交流信息。信息没有经过加密处理。

系统功能结构图如下图所示：



本系统的流程图见下图所示：

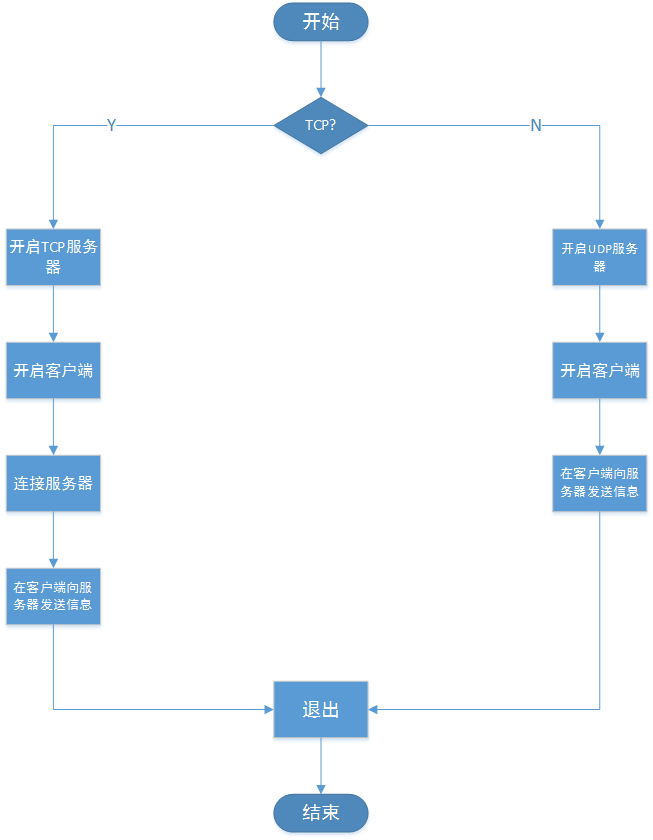


图1-2 系统流程图

1. **功能模块详细设计**

本节描述系统中的子系统和模块。

在线主机IP显示和刷新

基本功能：

打开客户端，显示界面立即显示出在线主机的ip，系统可以自动刷新或者通过按键来实现刷新。

基本方案：

采用网络中的广播方式来代替遍历最终来实现功能。文本信息的发送和接收

基本功能：

用户打开客户端，找到接收方的ip，点击后选择要发送的文件，然后点击发送文件按钮，等待接收方接收。

基本方案：

每一个客户端也是一个服务器，发送方作为客户端与接收方进行connect连接，建立TCP套接字连接之后通过send和receive函数来发送接收数据包,文本信息的发送和接收

基本功能：

用户打开客户端，找到接收方的ip，点击后在编辑区编辑信息，然后点击发送消息按钮。

基本方案：

发送方作为客户端与接收方进行connect连接，接受方返回一个是否接收的消息，如果接受，客户端在另一个端口上与接收方进行tcp连接，然后进行文件传输。