基于 QT 和 UDP 的一对一聊天室设计与实现

刘怀宇 白俊鸽
成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】 科技飞速发展，互联网逐渐步入千家万户，我们足不出户便知天下事；以前我们只能通过有线电话来联络，而就是这根线让我们无法随时随地与他人取得联系，正因如此我们对于互联网通信的需求也逐步凸显出来。本文以基于 QT 的一对一聊天室为例，介绍利用 UDP 传输消息、QT 作界面和实现的聊天室。该聊天室界面简洁、消息传送速度快，可以提供较好的用户聊天体验。

【关键词】QT；UDP；聊天室

1 引言

如今，网络遍布世界各地，只要有网络的地方我们便能使用网络通信，人们越来越习惯于使用网络通信，它能让我们更加及时便捷的收发消息，聊天室也就成为了必不可少的通信工具；目前也有许多网络聊天工具，如 QQ、微信等；聊天室可以基于即时通讯软件（如 QQ、微信）、万维网（如 imhaha、meebo）、P2P 软件等[1]，为用户提供了多种多样进行网络通信的方式，满足大家不同的需求，提供更好、更快的聊天通信体验。

2 协议的选择

对于传输协议的选择，TCP 和 UDP 各有优缺点，TCP 需要进行三次握手建立连接和四次挥手断开连接，这些都是需要一定开销和一定的时延来实现的，它是面向连接的可靠传输，且只能是点到点的，安全性高[2]，而考虑到收发消息的实际情况，收发消息不需要经过确认，即不需要建立连接；收发消息需要实时性，即消息的传输尽量少的时延和开销；聊天过程中，可以是一对一的、多对一的和多对多的聊天；发送的消息多为短消息，对信息保密性要求不高；综合上述特征选择了 UDP 作为传输协议，UDP 是面向无连接的、轻量级的、不可靠的、支持一对一，一对多，多对一和多对多的协议[2]。本文讨论的是基于 UDP 的一对一通信。

3 界面设计

3.1 聊天框区域设计

聊天框区由一个 QListWidget 对象、一个 QTextEdit 对象和两个 QPushButton 按钮组成。其中 QListWidget 对象为消息输出显示的窗口，自己发送的消息和收到的消息都将显示在此对象中，自己发送的消息显示在右侧，收到的消息显示在左侧，便于用户区分消息的来源。QTextEdit 对象为消息输入框，用户可以输入想要发送的消息（可以是单行也可以是多行）。一个按钮设置为发送按钮，用户输入完成后，点击此按钮即可将消息发送给指定的用户，另一个按钮设置为设置按钮，点击此按钮进行通信对象的选择。整体流程清晰、简洁，便于用户操作。

3.2 设置区设计

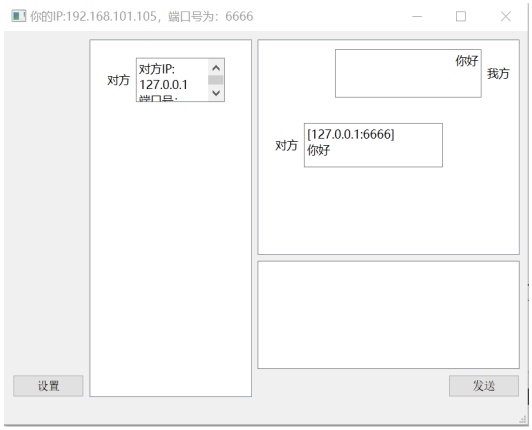
设置区的功能是选择要进行通信的用户，该区域由两个 QLable 对象、两个 QLineEdit 对象和两个 QDialogButtonBox 按钮组成。一个标签对象搭配一个 QLineEdit 对象，标签对象用于提示用户在 QLineEdit 对象中需要输入的内容，QLineEdit 对象接收用户输入的内容，内容分别为 IP 地址和端口号。一个按钮用于确认连接，另一个按钮用于取消当前窗口。

3.3 气泡设计

由一个 QLable 对象和一个 QTextEdit 对象组成。标签对象用于表明是哪一方发送的消息，对方用户发送的消息标签靠左，我方用户发送的消息标签靠右。QTextEdit 对象用于显示消息的内容。

3.4 通信对象框设计

由一个 QListWidget 对象组成，用于显示当前连接的通信用户信息。聊天界面如图 1：



图·1：聊天界面

4 功能实现

4.1 主要功能描述

本文主要讨论的是基于 QT 和 UDP 传输方式的一对一聊天室，功能简单、实用，各项功能都与实际通信方式相似，聊天室的工作流程比较简单，即：收到消息→处理消息→转发处理后的消息[3]，先输入要进行通信的用户 IP 和端口号，连接成功后就可以在输入框内输入消息，点击发送按钮发送。使用起来方便，快捷。具体流程如图 2：

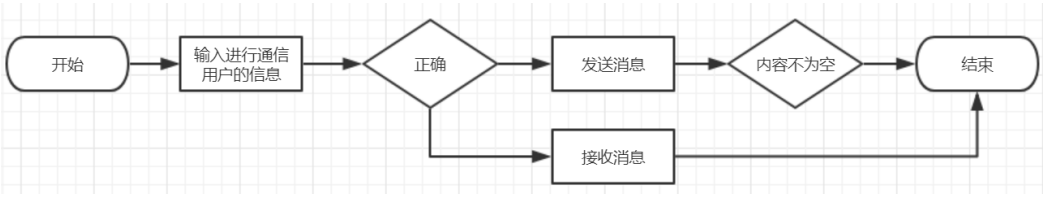


图 2：具体流程

4.2 获取本机 IP 地址功能

为获取本机 IP 地址，由于机器上可能还有无线网卡、VMware 等虚拟机软件，因此需要过滤掉虚拟网卡的信息，获取本机正在使用的 IP（筛选掉未使用的 IP 地址）。

该功能在 QString getLocalIP() 函数中实现。在此函数中，设置一个字符串类保存本机 IP 地址；一个 QProcess 类用于启动命令提示符，调用 QProcess 的 start 方法启动外部程序，并且外部程序会随着主程序的退出而退出；然后调用 QProcess 的 waitForStarted 和 waitForFinished 方法分别检测进程的开始和结束，用 QProcess 的 readAll 方法读出程序输出的内容并存储在一个字符串类中。

为找到当前正在使用的 IP 地址，设置一个 IP 地址的形式的字符串，使用正则表达式对 IP 地址进行匹配，while 循环调用正则表达式类的 indexIn 方法匹配其参数为程序输出的内容和开始位置，如果匹配成功用一个字符串类保存匹配到的内容；调用字符串类 indexOf 方法查找子网掩码，当外层循环匹配到的内容中找不到“255”，如果保存的 IP 地址不为空并且匹配的内容中匹配不到保存的 IP 地址从开始到最后一个“.”中的内容就跳出循环，反之将保存的 IP 地址内容赋值为匹配到的内容，如果外层循环匹配到的内容中寻找到“255”则调用正则表达式类的 matchedLength 方法从新的位置开始查找，直到找到当前电脑正在使用的 IP 地址为止。

4.3 选择通信用户功能

获取到了我们本地使用的 IP 地址后，我们还要连接通信对方的 IP 地址和端口，首先需要实例化自定义的设置框类，当点击设置按钮时通过 void slotShowInput() 函数调用弹出设置窗口。

窗口弹出后，在相应的输入框中输入对应的内容，并且输入的 IP 地址和端口号都需要在设置框类中通过正则表达式和 QRegExpValidator 类判断是否是正确的，若正确则使用输入框的 setValidator 方法获得手动输入的 IP 地址和端口号，然后点击确认按钮时发送一个信号来连接槽函数实现用户的创建。

用户的创建在槽函数 void slotCreateObject() 中实现，使用字符串类通过调用设置框类的 QString setIP() 和 QString setPort() 函数保存设置框中输入的 IP 地址和端口号；当 IP 地址和端口号都为空或者是 IP 地址和端口号都与上一个通信的用户相同时都不能创建新的通信用户，用一个 QMap 类保存上次的通信对象的 IP 地址和端口号，调用它的 value 方法来判断当前输入的 IP 地址和端口号是否和之前用户的相同，如果不同且都不为空，将新输入的 IP 地址和端口号通过调用 QMap<QString, QString> getInfoIP(const QString &IP, const QString &port) 函数来保存新的用户的信息，其中 const QString &IP 是传入刚才保存的新连接用户的 IP 地址的地址，const QString &port 是刚才保存的新连接用户的新连接接用户的端口号的地址；保存好后，组合一个字符串显示对方的信息，实例化一个 QListWidgetItem 和对方的气泡类，通过调用气泡类的 void setText(QString text) 函数设置内容，其中 QString text 需要传入的是对方信息的字符串，然后在通信对象框的显示窗口中调用 addItem 方法将新实例化的项目加入 QListWidget 中，并且在添加了新的项后通过调用窗口的的 scrollToBottom 方法设置滚动条到最低处。

4.4 发送消息功能

QT 默认没有添加 QtNetwork 库，为实现发送和接收消息功能，首先需要手动添加 QtNetwork 库，让 QT 程序能够使用网络。

首先需要用 QUdpSocket 类创建套接字，并且给套接字绑定端口号，然后在发送按钮的槽函数中实现发送消息功能。

在槽函数中，首先需要通过 QMap 的 value 方法获得保存的对方的 IP 地址和端口号，用户输入要发送的消息在 QTextEdit 中，通过调用 QTextEdit 的 toPlainText 方法获取到编辑区的内容保存到字符串类中。

当没有输入内容和没有设置接收方 IP 地址和端口号时都不能发送消息，都满足时才能发送消息，发送消息时，使用 QUdpSocket 类的 writeDatagram 方法给指定的 IP 地址发送数据，然后实例化一个 QListWidgetItem 类和一个我方气泡类，将输入在 QTextEdit 的内容通过调用 toPlainText 方法获取，再调用我方气泡类的 void setText(QString text) 函数，其中 QString text 需要传入的是获取到的 QTextEdit 的内容，在此函数中调用 QTextEdit 的 setText 和 setAlignment 方法分别设置内容和右对齐；在对话框的 QListWidget 中通过调用 QListWidget 的 addItem 和 setItemWidget 分别新增项和将新的 QListWidgetItem 加入并显示内容到 QListWidget 中并且调用窗口的 scrollToBottom 方法将滚动条自动设置到最低。

当发送完毕后调用 QTextEdit 的 clear 方法清空输入的内容，调用 setFocus 方法将光标重新聚集。具体代码图 3：

```
//发送信息
void MainWindow::on_btn_send_clicked()
{
    //获取接收方IP
    QString IP=infoIP.value("IP");
    quint16 port =infoIP.value("port").toUInt();

    //获取编辑区内容
    QString content =ui->textEdit->toPlainText();

    //没有输入不能发送信息
    //没有设置IP和端口号也不能发送信息
    if(content!=tr("")&&IP!=tr("")&&infoIP.value("port")!=tr(""))
    {
        //给指定IP发送数据
        m_udpSocket->writeDatagram(content.toUtf8(),QHostAddress(IP),port);
        qDebug()<<IP<<port<<content;

        QListWidgetItem *sendItem =new QListWidgetItem();
        MyBubble *sender=new MyBubble();
        sender->setText(ui->textEdit->toPlainText());
        sendItem->setSizeHint(QSize(200,100));
        ui->chatWindow->addItem(sendItem);
        ui->chatWindow->setItemWidget(sendItem,sender);

        ui->chatWindow->scrollToBottom();
    }
    //发送之后清空输入框，光标聚焦
    ui->textEdit->clear();
    ui->textEdit->setFocus();
}
```

图·3：具体代码

4.5 接收消息功能

当对方成功发送消息，触发了套接字的 readyRead 信号，然后自动调用槽函数 void slotDealMessage()，在此函数中设置一个字符数组保存数据，QHostAddress 和 quint16 分别保存 IP 地址和端口号，通过调用 QUdpSocket 类的 readDatagram 方法读取数据报，并且调用 QString toIPv4(const QHostAddress &addr) 函数将 IP 地址转化为 ipv4 的地址，const QHostAddress &addr 为保存的 IP 地址的地址，调用 QHostAddress 类的 toIPv4Address 转换，构造一个 QHostAddress，通过 toString 方法获取相对应的 IP 地址。

当有数据发送出来，将数据显示到聊天框的对应位置，显示方式同发送消息的方式相似，只需将我方气泡类换为对方气泡类。同时需要显示新的通信用户的信息在通信用户的 QListWidget，方法同选择通讯用户中的显示方法。

5 结束语

本文介绍了如何使用 QT 和 TCP 实现一对一的聊天室，　前者通过灵活的使用实现了界面的美观性和简洁性给用户良好的操作体验，后者为信息的传输提供了便利的底层逻辑，在产品的可使用度上仍有很大的进一步完善和深入的空间，如实现对用户之间通信、发送图片等功能，总体上该产品逻辑清晰、代码简单，基本功能完善。

【参考文献】

[1] 武彩红，薛艳峰，乔丽，刘凯，高志娥.UDP 聊天室系统的设计与实现 [J]. 吕梁学院学报,2014,4(02):37-39.

[2] 王龙涛．基于 UDP 协议下的局域网即时通讯 [J]. 通讯世界,2019,26(07):137-138.

[3] 窦立君，闫志刚，韩坷罕．基于 Qt 的网络聊天软件服务器端设计 [J]. 软件导刊,2016,15(11):106-108.