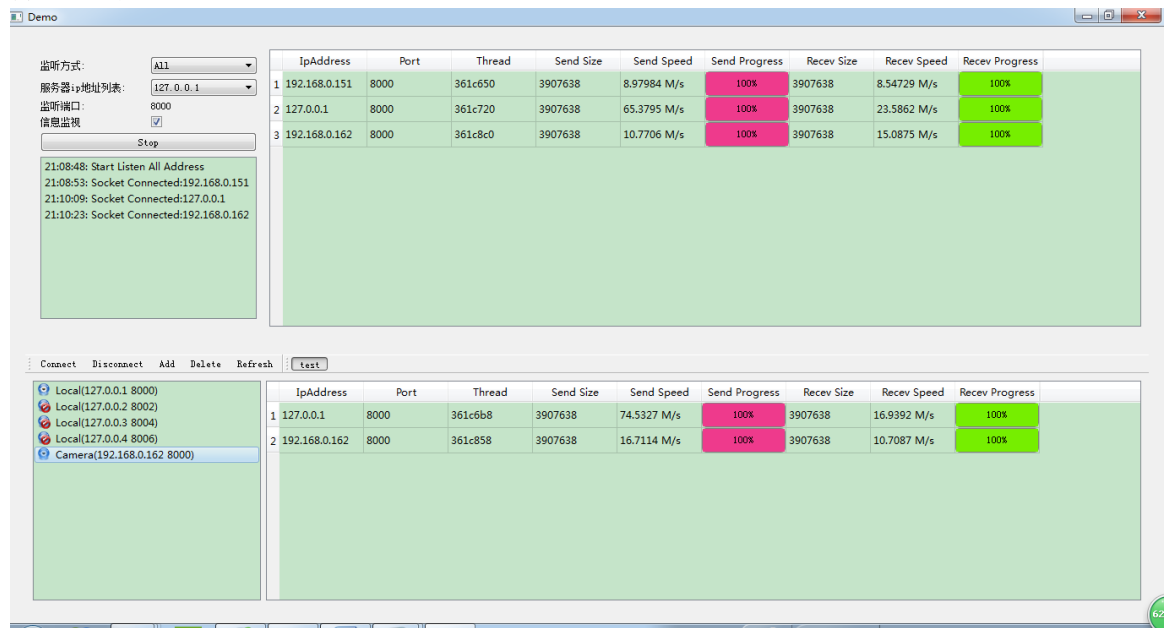


设计说明

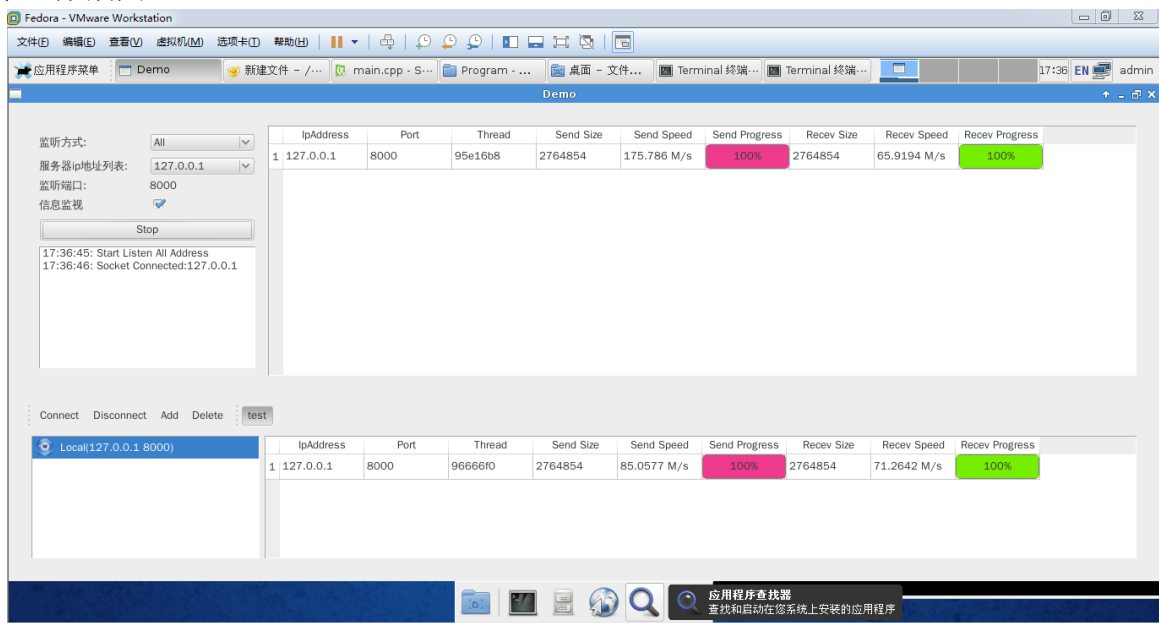
概要:

此demo主要用于服务端及客户端的多线程socket通讯管理问题，为每一个socket连接建立一个线程进行管理，使用此demo为蓝本可快速的构建出多线程socket通讯的服务端及客户端，数据传输中使用了一个简单的自定义报文头，用于解决tcp的粘包问题。服务端及客户端接收及发送数据由两部分构成，一部分为命令（cmd）quint32型，一部分为需传输的数据data QByteArray型。客户端及服务端可以根据CMD的不同来约定对数据的不同解析方式。该demo经过Qt4.8.6及Qt5.3.1在windows、Fedora、及嵌入式Arm的linux多平台混合连接测试并持续24小时以上测试，无内存泄漏，稳定运行，可用于轻量级工程实例中作为网络数据传输的解决方案。

windows软件测试截图:



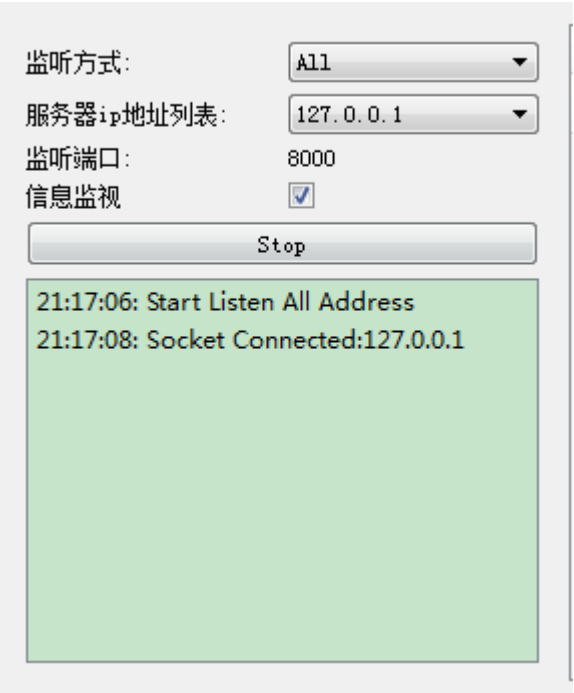
Linux软件运行截图:



软件的demo把服务端和客户端放在一起显示出来了，上半部分为服务端，下半部分为客户端，在真正使用时，拆开两部分分别在不同的应用中初始化即可。同时提供多种初始化风格，可对详细信息表格显示或者隐藏。

一. 服务端设计说明

1. 界面预览



这里对功能进行简单设置：

监听方式：All：监听所有本地能找到的ip地址

One：只监听ip地址列表选中的地址

服务器ip地址列表：可以识别到的所有服务器ip地址

监听端口：现在默认的是8000，提供了端口可更改

信息监视：要不要显示下面的一些常用连接信息。

连接信息表

显示连接上来的服务器ip地址、本地监听的端口、socket的指针地址、发送的数据大小、发送的速度、发送进度、接收数据的大小、接收的速度、接收进度。

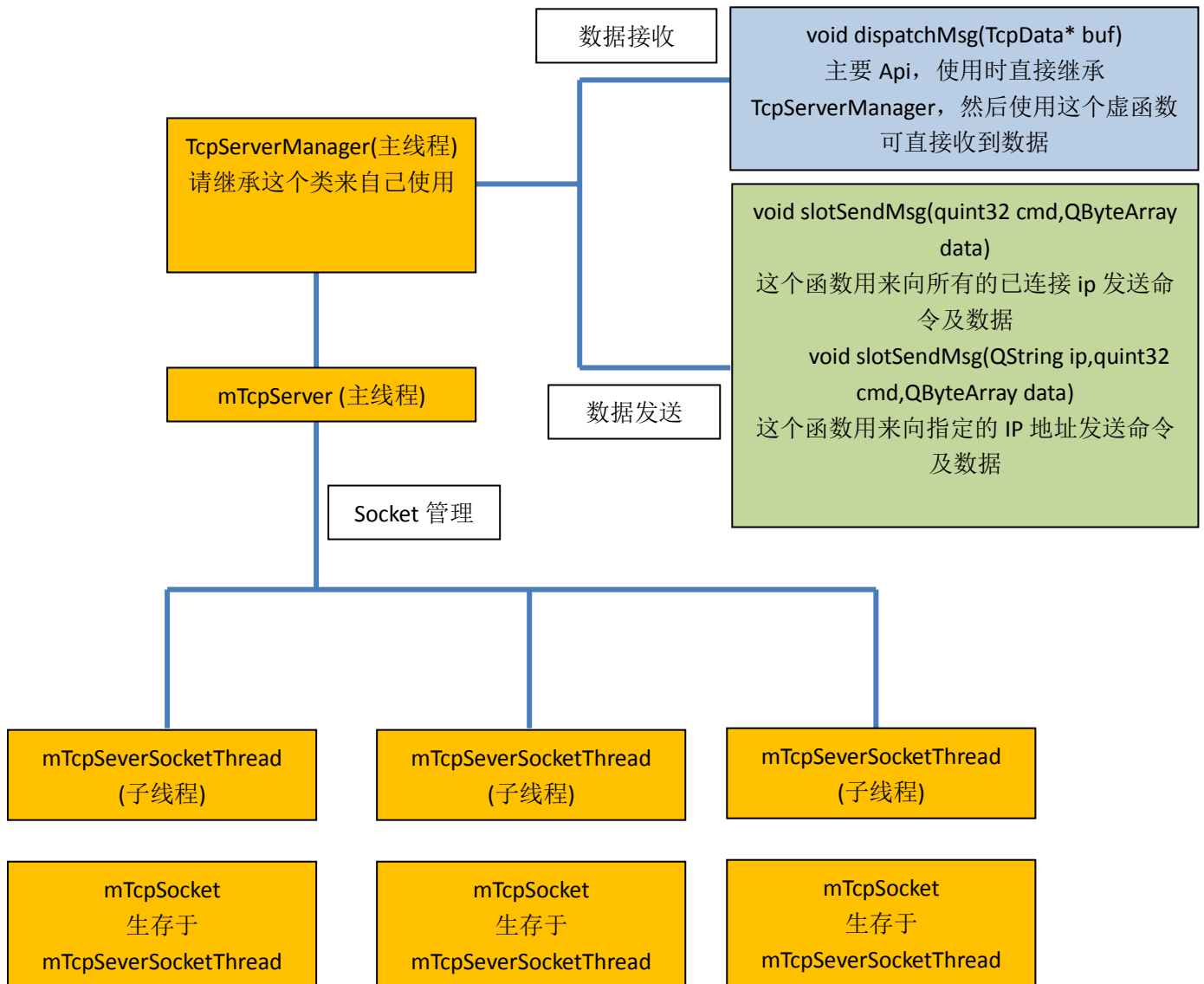
每当有一个客户端连接进来后会创建一个线程，当客户端的socket连接断开后线程销毁。

	IpAddress	Port	Thread	Send Size	Send Speed	Send Progress	Recev Size	Recev Speed	Recev Progress	
1	127.0.0.1	8000	adc730	3907638	47.1726 M/s	100%	3907638	38.8189 M/s	100%	

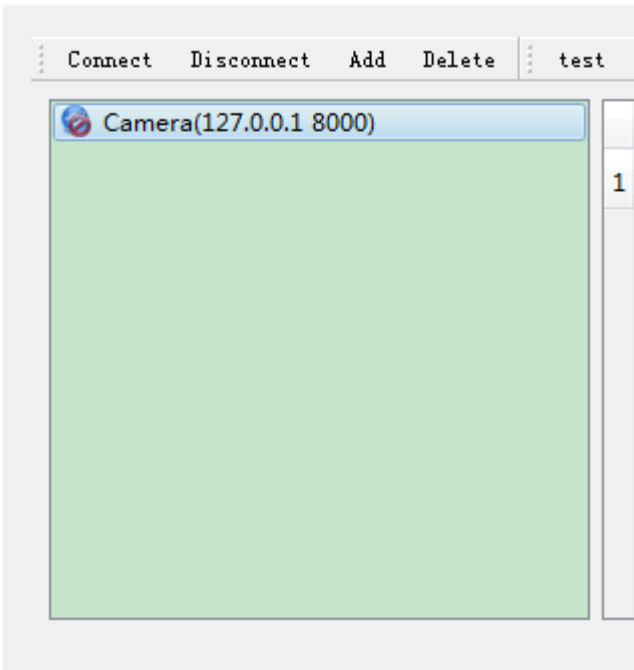
2.程序结构（根据程序中的类名）

2.1 服务端使用请继承TcpServerManager类后使用继承后的类，重写dispatchMsg虚函数，在此对接收的数据进行解析，注意，这是在主线程中！使用buf中cmd来区分你自己定义的各种类型的数据解析。

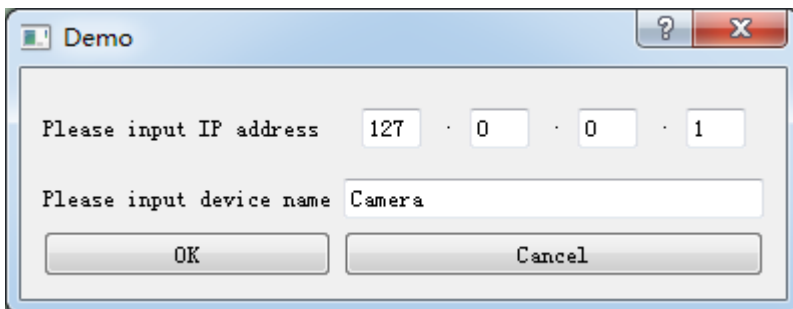
2.2使用slotSendMsg函数来发送数据，带ip和不带ip的方式两种，带ip的发送到指定ip地址，不带ip的发送到所有连接的客户端，由于服务端只能监听一个port，所以不需要填port号。



二. 客户端设计说明



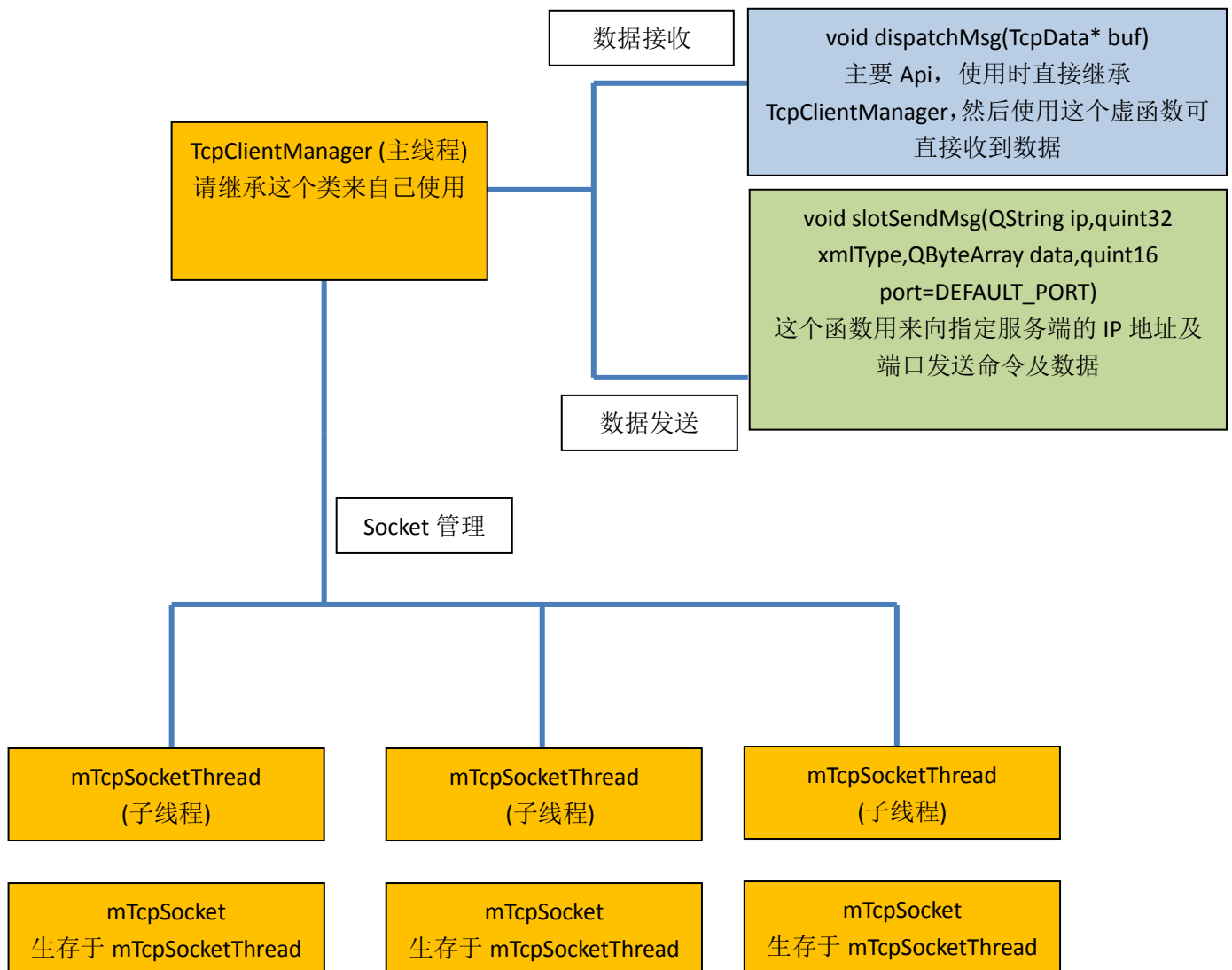
1. **connect**: 根据选中的ip地址及端口发出连接请求，连接成功后图标状态改变（连接状态根据socket真实状态反应）。
2. **disconnect**: 断开选中的socket连接。
3. **add**: 添加一个ip地址及这个地址的名字，默认8000端口。



4. **delete**: 删除选中的socket连接，并清除线程（socket连接一定得断开才能删除，不然qt的内核会出错崩溃，原因不知。。。）。
5. **test**: 测试功能，进行压力测试，客户端发送测试数据，现在测试数据是读取App文件夹TestImage下的图片，服务端收到数据后马上把收到的数据原样返回客户端，服务端再收回数据后等2秒后读取下一张图片的数据发给服务端，简单来说就是踢皮球一样的测试，客户端和服务端之间反复的收发数据。

客户端程序结构

- 1.服务端使用请继承TcpClientManager类后使用继承后的类，重写dispatchMsg虚函数，在此对接收的数据进行解析，注意，这是在主线程中！使用buf中cmd来区分你自己定义的各种类型的数据解析。
- 2.使用slotSendMsg函数来发送数据，发送数据至指定的ip及端口的服务器。



除了上述版本外，在此基础上还做了一个测试版本，方便大家更直观的看到效果。将测试服务端及客户端中流转的图片显示出来。

