计算机网络 第二次实验 实验报告

隋唯一 2017011430 软件学院

1. 实验环境

硬件：intel core i7

操作系统：windows10

IDE:pycharm;visual studio2010

1. 客户端

由于时间紧张，使用socket编程实现了命令行的客户端（见下文服务器端部分），另外使用python3+vc2实现了一个gui客户端，视频源是香港卫视。

运行截图：



1. 服务器端

服务器端包括了一个服务器与一个客户端，均为命令行窗口，组织在vs的同一个工程中。它们的架构由下面两个UML图给出：

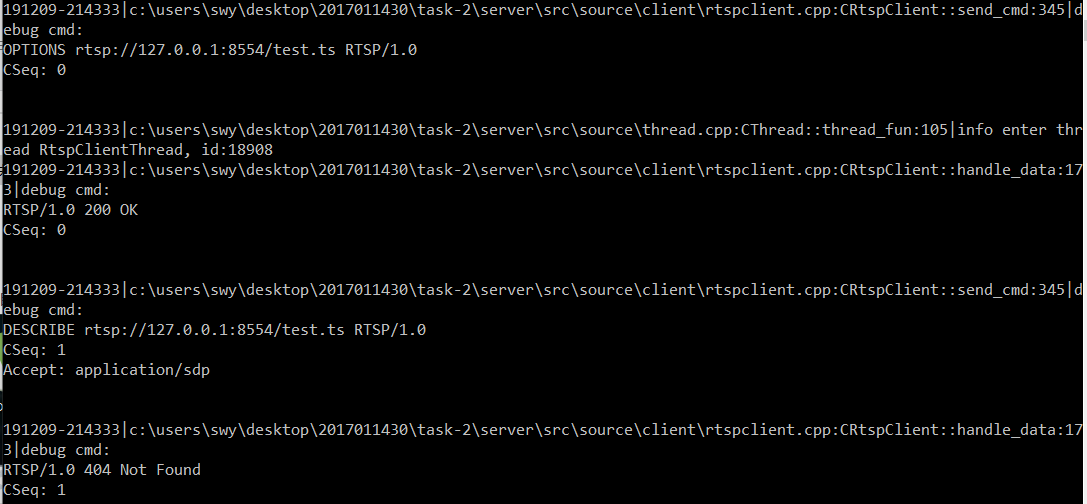
客户端架构：

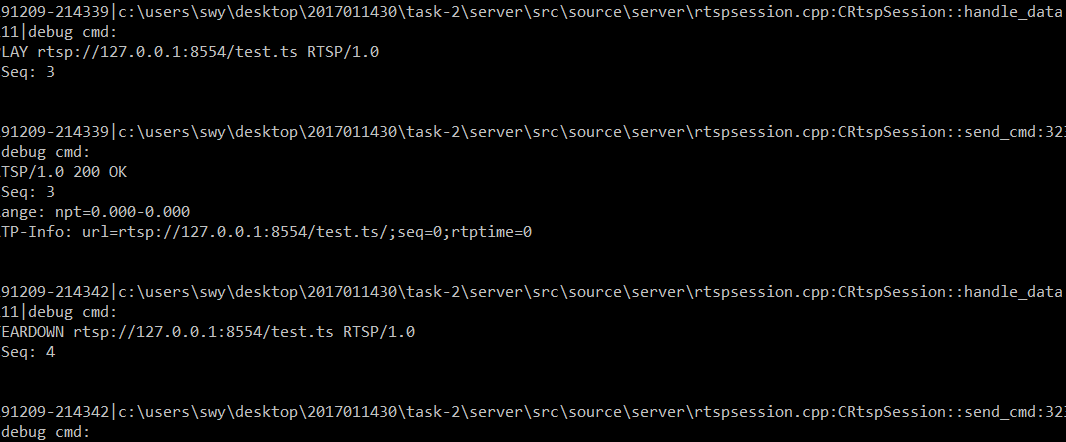


服务器端架构：



程序运行截图如下：

客户端：

服务器端：

客户端与服务器端在同一主机上运行，因为我在代码内固定了IP地址为127.0.0.1，端口为8554

特色：UDP/TCP socket实现，RTP/RTSP实现，具体实现的类以及功能如下两表：

客户端：

|  |  |
| --- | --- |
| 类名 | 功能 |
| CThread | 线程类，用于创建线程 |
| CSock | 套接字基类 |
| CTcpSock | TCP套接字类，继承自CSock，完成TCP连接、数据收发 |
| CUdpSock | UDP套接字类，继承自CSock，完成UDP数据包收发 |
| CRtspClient | RTSP客户端类，发送RTSP请求，处理RTSP响应，接收RTP包 |

服务器端：

|  |  |
| --- | --- |
| 类名 | 功能 |
| CThread | 线程类，用于创建线程 |
| CSock | 套接字基类 |
| CListenSock | 监听套接字类，继承自CSock，用于监听并创建新的套接字 |
| CTcpSock | TCP套接字类，继承自CSock，完成TCP连接、数据收发 |
| CUdpSock | UDP套接字类，继承自CSock，完成UDP数据包收发 |
| CRtspSvr | RTSP服务，创建并管理RTSP会话 |
| CRtspSession | RTSP会话类，处理RTSP请求、响应 |
| CDataSrc | 数据源类，完成RTP打包 |
| CTsFileReader | TS文件读取类，完成TS文件数据的读取 |

（TS文件是我本地的一个test文件）

1. 我的心得

在查阅资料的过程中，我发现目前RTP协议在IP摄像头领域应用广泛（一个出场频率非常高的摄像头是海康威视摄像头），当然，在其他领域也有应用，主要是直播领域。

最开始想要实现一个连接标准RTP服务器的客户端，选用live555作为服务器，总是莫名其妙的掉线，于是就放弃了。后来发现，应该是服务器需要客户端定时发送保活数据包（就是通过这个数据包通知服务器该客户端还处于工作状态），这应该并不是RTP/RTSP协议中规定的，今后选择测试服务器时要谨慎。

另外，这学期的全部socket编程都采用c/c++语言完成，感觉效率有一点低下，调试起来也比较麻烦，看来还是要掌握python的相关编程知识。

另外，TASK1的要求令人疑惑，希望以后作业的要求能够更加明确一些。

1. 参考资料

博客资源：

https://www.cnblogs.com/lidabo/p/4160138.htmlhttps://blog.csdn.net/qq\_37909553/article/details/81349156

Github链接：

https://github.com/vinichou94/RTSPVideoStreamingServerhttps://github.com/enthusiasticgeek/gstreamer-rtsp-ssl-example