

RTP大作业实验报告

2017011438 曾正

TASK-1

功能说明

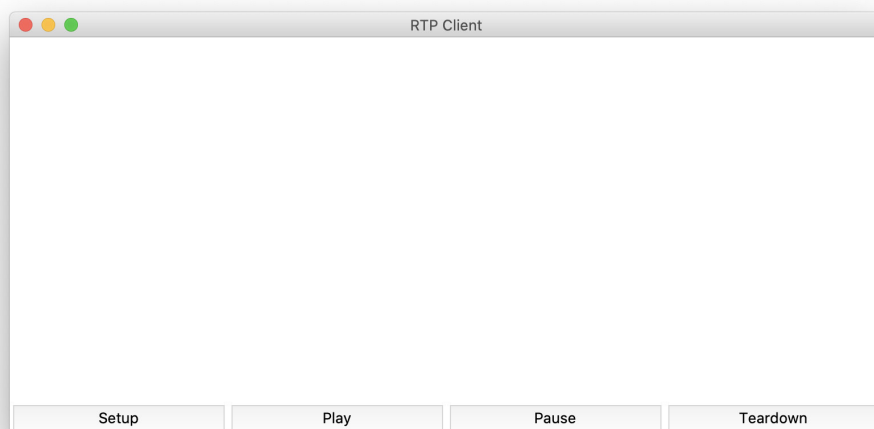
1. Server发送.jpg文件到Client接收并显示

运行方式

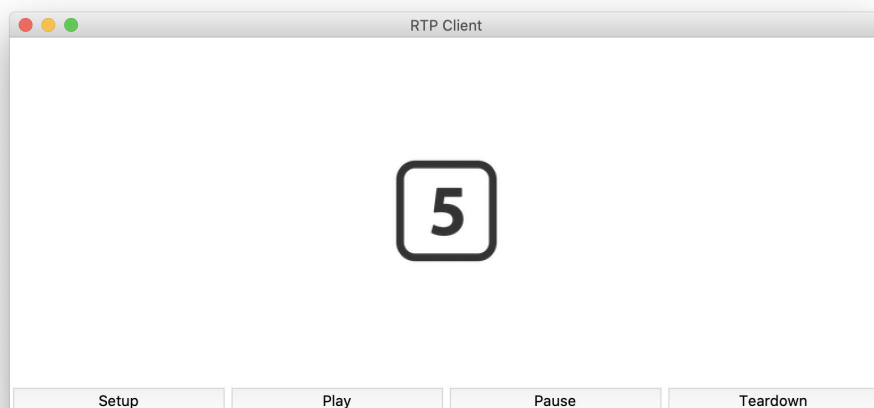
请参照TASK-1/doc.pdf

运行效果

初始界面



播放界面



Client使用说明

Button功能

- [Setup] 点击初始化，播放前先点击此按钮
- [Play] 点击开始播放，暂停后再次开始也点击该按钮
- [Pause] 点击暂停播放
- [Teardown] 点击终止传输并关闭客户端

运行原理

Server端

- 初始化建立Socket并bind，等待客户端连接
- 通过RTSPSocket接收Client端的指令并分别对SETUP、PLAY、PAUSE、TEARDOWN进行处理。
- SETUP部分先建立传输流，并设置服务器状态，再为此次连接生成Session，若成功则通过RTSPSocket返回200状态码
- PLAY部分，在通过服务器状态判断后建立（首次播放）或获取（暂停后重新播放）RTPSocket并开始传输数据，若成功则通过RTSPSocket返回200状态码。最后设置线程状态
- PAUSE部分,在通过服务器状态判断后重新设置服务器状态，并修改线程状态即暂停。若成功则通过RTSPSocket返回200状态码
- TEARDOWN部分，修改线程状态即终止并关闭RTPSocket。若成功则通过RTSPSocket返回200状态码
- 流读取会按顺序从img文件夹下持续读取.jpg文件，由1.jpg到2.jpg并持续向后读取，读到末尾n.jpg则再次读取1.jpg

Client端

由于为示例代码故省略

TASK-2

功能说明

1. Server发送视频文件到Client接收并播放
2. 支持视频格式包括但不限于.mp4 .mov .avi .flv .m4v .mkv .Mjpeg
3. 视频的暂停和播放
4. 视频文件名称的显示
5. 简单的视频列表
6. 可前后拖动的进度条，并会随视频播放的进度移动
7. 视频播放速度调整，支持2.0和0.5倍速
8. 支持全屏播放

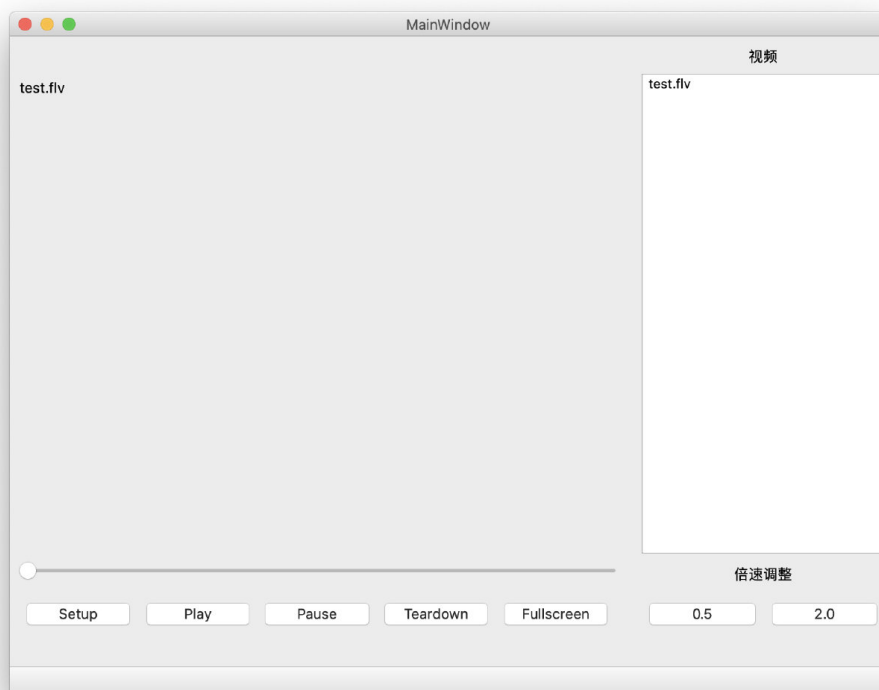
9. 全屏状态下可用[ESC]键退出全屏，并且可通过[SPACE]键控制视频的播放暂停
10. 通过分段传输的方式支持**高分辨率视频传输**（TASK-2/test.flv为可用的1080p60fps视视频文件）
11. 支持**多用户**连接播放
12. 支持**断点重连**，Client在TEARDOWN后重新通过相同的port连接即可由上次的播放断点继续播放

运行方式

请参照TASK-2/doc.pdf

运行效果

初始界面



播放界面



Client使用说明

Button功能

- [Setup] 点击初始化，播放前先点击此按钮
- [Play] 点击开始播放视频，暂停后再次开始也点击该按钮
- [Pause] 点击暂停视频播放
- [Teardown] 点击终止传输并关闭客户端
- [Fullscreen] 点击开启全屏模式
- [0.5] 点击开始以0.5倍速播放
- [2.0] 点击开始以2.0倍速播放

Slider功能

- 显示当前播放进度
- **拖动**进度条可改变播放进度

全屏时

- 按[ESC]键可退出全屏
- 全屏状态下按[SPACE]键可播放或暂停

其他

- 在点击[Teardown]结束后重新以相同的[client_port]连接即可从上次中断位置继续播放
- 不同Client通过不同port连接即可实现多用户连接

运行原理

- 初始化建立Socket并bind，等待客户端连接，对不同客户端建立不同线程
- 客户端通过RTSPSocket发送指令，服务端通过RTSPSocket接收Client端的指令并分别对SETUP、PLAY、PAUSE、TEARDOWN、SLOWER、FASTER、RELOCATE进行处理。
- SETUP部分服务端在接收到用户端指令后先建立传输流，并设置服务器状态，再为此次连接生成Session，若成功则通过RTSPSocket返回200状态码。
- PLAY部分服务端在接收到用户端指令后，在通过服务器状态判断后建立（首次播放）或获取（暂停后重新播放）RTPSocket并开始传输数据，若成功则通过RTSPSocket返回200状态码。最后设置线程状态。
- PAUSE部分服务端在接收到用户端指令后,在通过服务器状态判断后重新设置服务器状态，并修改线程状态即暂停。若成功则通过RTSPSocket返回200状态码。
- TEARDOWN部分服务端在接收到用户端指令后，修改线程状态即终止并关闭RTPSocket。若成功则通过RTSPSocket返回200状态码。
- SLOWER / FASTER部分服务端在接收到用户端指令后，通过修改发送包的速度从而达到修改播放速度的目的。
- RELOCATE部分服务端在接收到用户端指令后，通过获取特定帧的方式，确定新的播放点。
- 进度条与视频同步，通过特定帧在所有帧的位置与进度条当前长度占总长度比例绑定实现。
- 高分辨率传输，通过对每一帧图片进行分片传输实现。
- 全屏则通过修改播放区域大小以及线程管理实现，全屏下的播放暂停通过重写pyqt5的keyPressEvent完成。
- 多用户连接则通过对不同用户设置不同线程，达到互不干扰的目的完成。
- 断线重连是对不同用户分别记录其播放进度并在[SETUP]时设置对应播放点完成。