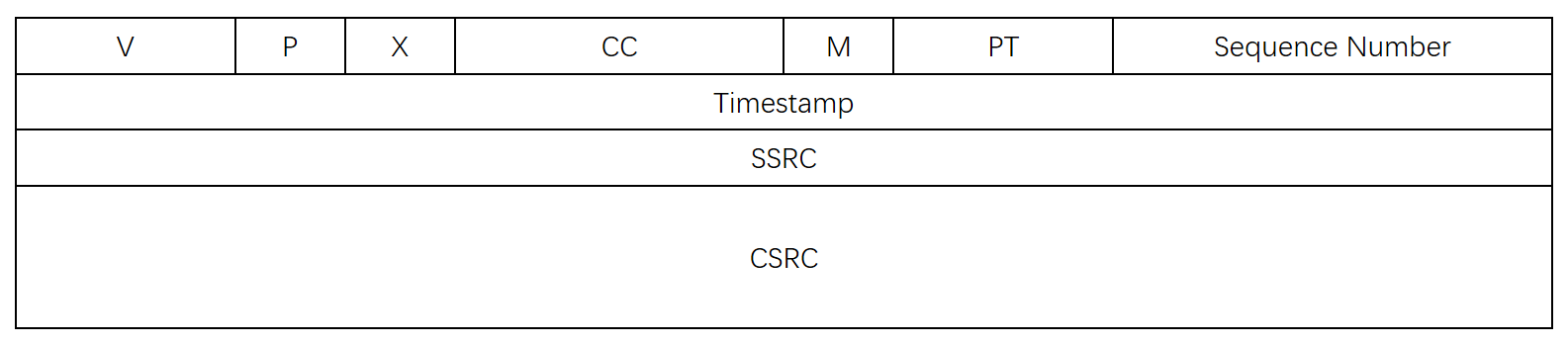
1. **RTCP包格式：**



各部分含义：

版本号（V）：2比特，用来标志使用的RTP版本

填充位（P）：1比特，如果该位置为1，则该RTP包的尾部就含附加的填充字节

扩展位（X）：1比特，如果该位为1，RTP固定头部后面就跟有一个扩展头部

CSRC计数器（CC）：4比特，含有固定头部后面跟着的CSRC数目

标记位（M）：1比特，该位的解释由配置文档来承担

载荷类型（PT）：7比特，标识了RTP载荷的类型

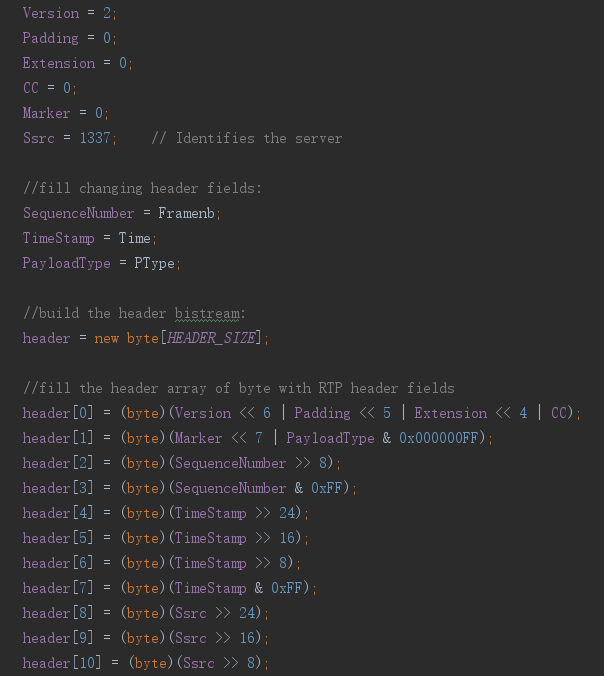
序列号（SN）：16比特，发送发在每发送一个RTP包后将该域的值增加1，接收方可用于检测包的丢失及恢复包序列，序列号的初始值是随机的

时间戳（Timestamp）：32比特，记录了该包中数据的第一个字节的采样时刻。在一次会话开始时，时间戳初始化为一个初始值。时间戳是去除抖动和实现同步不可缺少的部分

同步源标识（SSRC）：32比特，同步源就是指RTP包流的来源。在同一个RTP会话中不能有两个相同的SSRC值

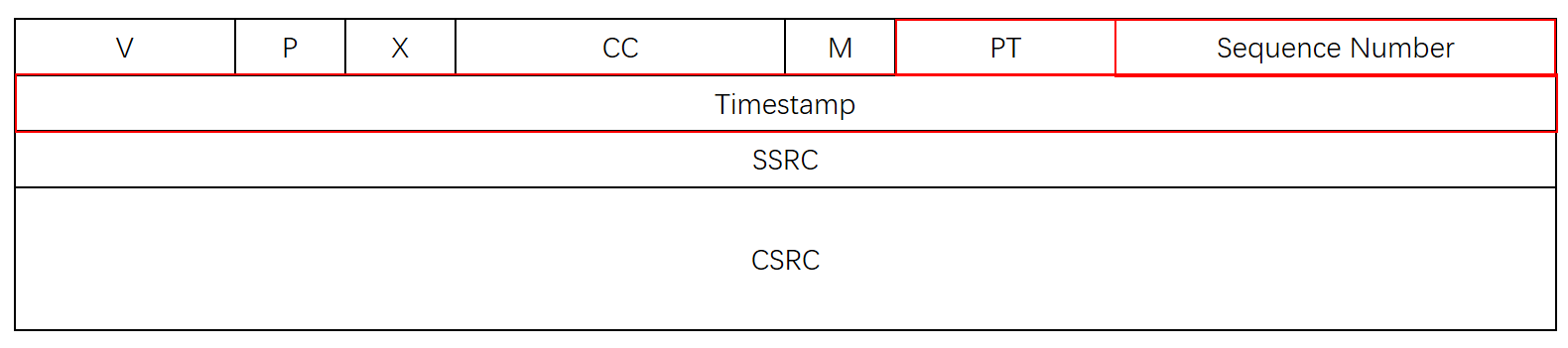
贡献资源列表（CSRC List）：0-15项，每项32比特，用来标志对一个RTP混合器产生的新包有贡献的所有RTP包的源。由混合器将这些有贡献的SSRC标识符插入表中，以便接收端能正确指出交谈双方的身份

代码中RTP包实现：



可以看出，自定义实现了RTP包，将各部分位移拼接实现RTP包格式。

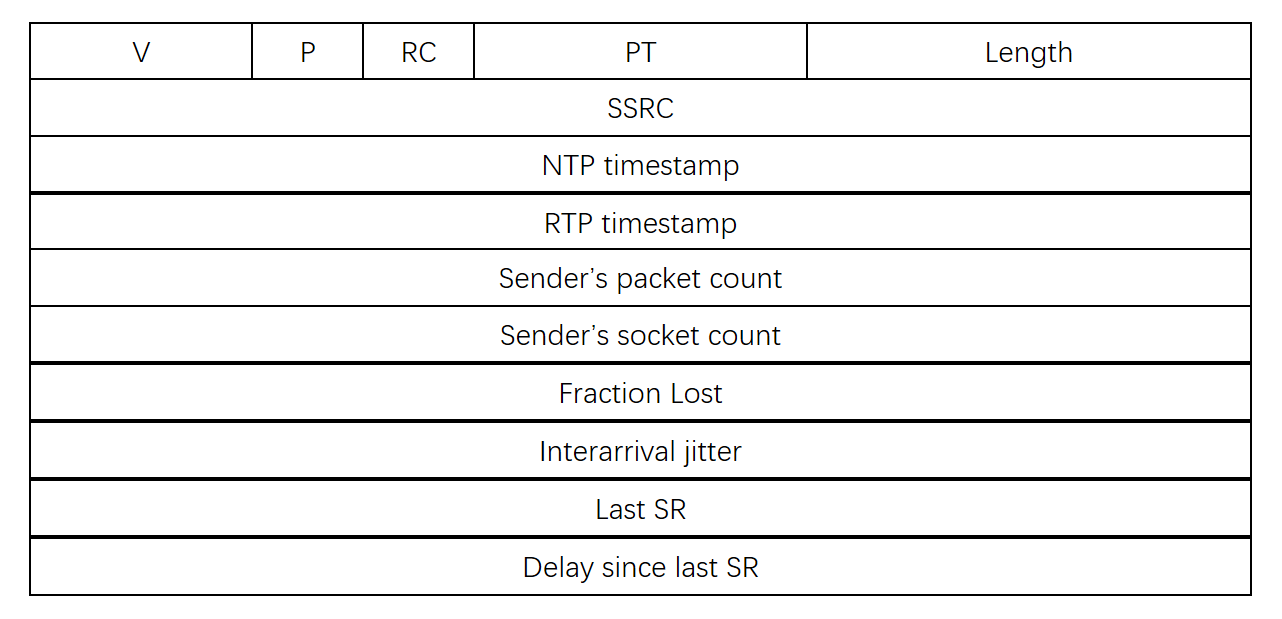
代码中RTP包格式：



其中黑色为固定不变部分，红色视实际参数而异，包含：

PT：载荷类型、Sequence Number：包序号、Timestamp：时间戳

1. **RTCP包格式**



各部分含义：

版本号（V）：2比特，用来标志使用的RTCP版本

填充位（P）：1比特，如果该位置为1，则该RTCP包的尾部就含附加的填充字节

接受报告计数器（RC）：5比特，该接收包中的接收报告块的数目，可以为零

包类型（PT）：8比特

长度域（Length）：16比特，存放的是包中以32比特为单位的总长度减一

同步源（SSRC）：发送者的同步源标识，与对应的RTP包中的SSRC一样

NTP timestamp：包发送的绝对时间，NTP的作用是同步不同的RTP媒体流

RTP timestamp：与NTP时间戳对应，与RTP数据包中的RTP时间戳具有相同的单位和随机初始值

Sender’s packet count：从开始发送包到产生该包的这段时间内，发送者发送的RTP数据包的总数，发送者改变其SSRC时，这个域要清零

Sender’s socket count：从开始发送包到产生该包的这段时间内，发送者发送的净荷数据的总字节数（不包括头部和填充），发送者改变其SSRC时，这个域要清零

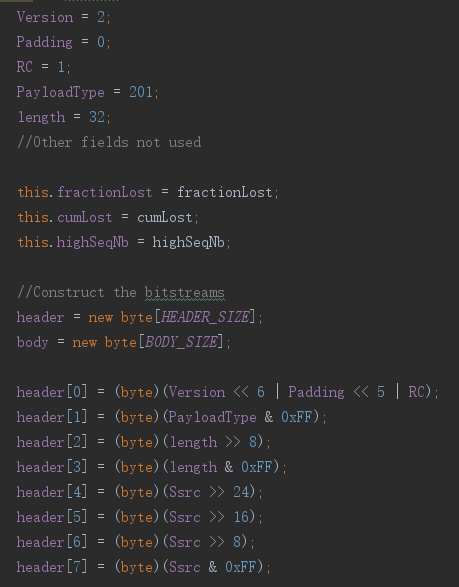
丢失率（Fraction Lost）：表明从接收到上一个RTCP包以来，从同步源（SSRC）来的RTP数据包的丢失率

接收抖动（Interarrival jitter）：RTP数据包接收时间的统计方差估计

上次SR时间戳（Last SR）：取最近从SSRC收到的RTCP包中的NTP时间戳的中间32比特

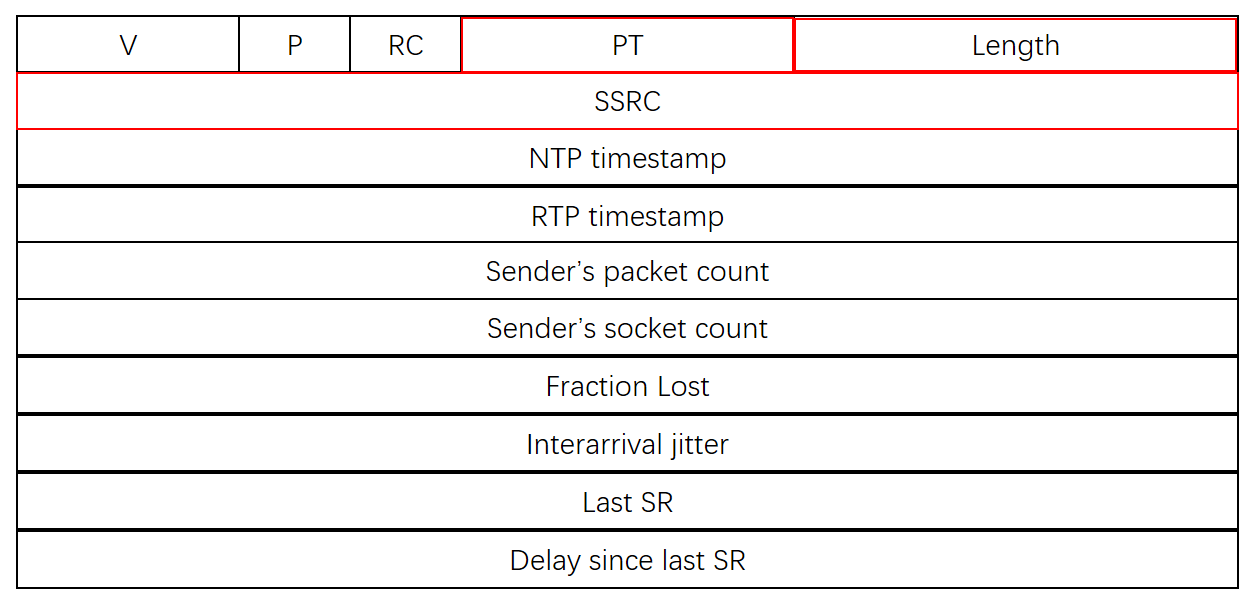
上次RTCP包以来的延时（Delay since last SR）：上次从SSRC收到RTCP包到本报告的延时

代码中RTCP包实现：



可以看出，自定义实现了RTCP包，并通过位移拼接进行组装

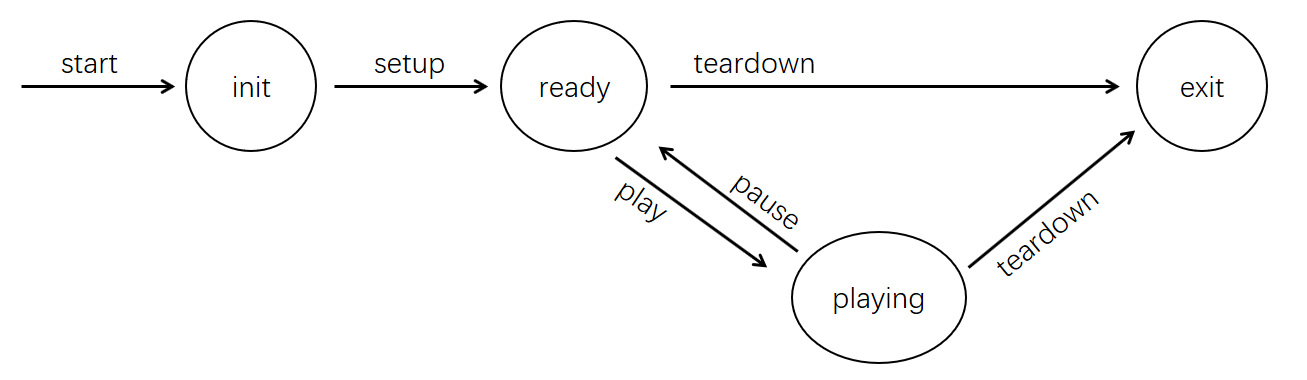
代码中RTCP包格式：



其中黑色为固定不变部分，红色视实际参数而异，包含：

PT：载荷类型、Length：包长度、SSRC：同步源标志

1. **RTSP状态转移图**



状态说明：

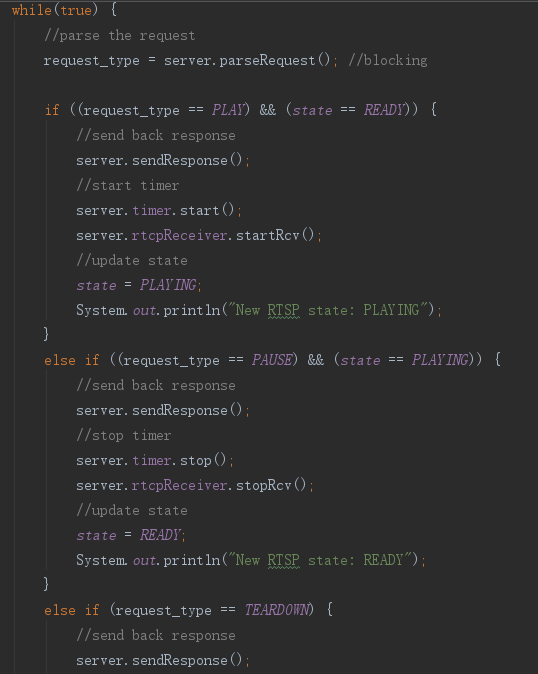
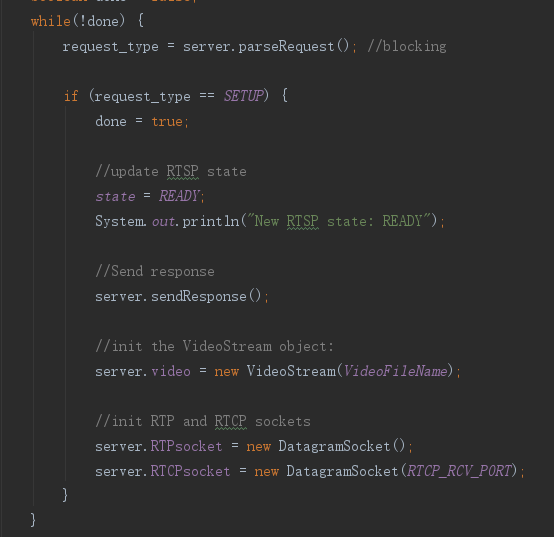
Init：初始状态；程序启动时，服务端指定UDP连接端口，客户端指定服务器ip、端口、请求的视频文件名，客户端和服务端通过UDP建立连接

Ready：就绪状态；客户端准备就绪并向服务器端发送setup请求，服务端收到后解析，进入就绪态

Playing：播放状态；处于就绪状态下时，客户端通过向服务器端发送play请求，可以实现播放的功能，进入播放状态；播放状态下客户端向服务器端发送pause请求，可是实现暂停功能，进入就绪态

Exit：退出状态；在就绪态、播放态的情况下，客户端发送teardown请求，服务器端将终止数据传输，并退出

代码截图：



1. **基本操作**

包含状态转移图中的setup、play、pause、teardown以及describe这5种操作

1. **存在问题**

目前只能播放Mjpeg格式视频文件，MJPEG是一种基于静态图像压缩技术JPEG发展

起来的动态图像压缩技术，可以生成序列化的运动图像。代码中调用库函数可以直接“切割”视频文件，将其转化为一帧一帧的图像，从而实现数据的传输。

还在研究