大家好，我是北小菜。

今天讲一个实战项目《从零开始编写一个RTSP服务器》

本次视频为系列视频第三讲，讲解内容是：**实现一个基于UDP的RTP传输音频aac的RTSP服务器，并能够拉流播放**

1. 本次实战项目系列视频，分为如下5个：
2. 从零开始编写一个RTSP服务器（1）RTSP协议讲解及代码实现
3. 从零开始编写一个RTSP服务器 （2） 实现一个基于UDP的RTP传输h264的RTSP服务器，并能够进行rtsp拉流播放
4. 从零开始编写一个RTSP服务器（3）实现一个基于UDP的RTP传输aac的RTSP服务器，并能够进行rtsp拉流播放
5. 从零开始编写一个RTSP服务器（4）实现一个基于TCP的RTP同时传输h264和aac的RTSP服务器，并能够进行rtsp拉流播放
6. 从零开始编写一个RTSP服务器（5） 基于我的开源项目BXC\_RtspServer进行详细的源码讲解，这是一个完整可用，支持多线程，基于socket编写的IO多路复用的RTSP流媒体服务器

上期视频，我们已经实现了一个基于UDP的RTP传输h264的RTSP服务器，客户端播放器能够向我们的RTSP服务端发起连接建立的请求，并且客户端在发起RTSP的Play请求以后，RTSP服务端在回复了Play请求之后，开始源源不断的通过RTP协议向客户端推送h264视频流。

本期视频我们需要实现，客户端建立与RTSP服务端的连接后，并且在RTSP服务端回复了客户端的Play请求以后，服务端需要源源不断的读取一个本地aac音频文件，并将读取到的aac音频码流封装到RTP数据包中，再推送至客户端。这样我们就实现了一个简单的支持RTSP协议的流媒体分发音频流的服务。

因为第2讲我们已经介绍过RTP协议，所以本讲就不介绍了。并且我昨天也更新了一个**《音视频开发之音频原理》**的视频教程，里面已经详细的介绍了音频的一些基本知识点，并在教程的最后我们得知麦克风在录音采样后，直接获得的就是音频原始数据pcm。但由于pcm数据较大，所以通常需要压缩之后才能传输或保存，常见的音频压缩技术有aac，g711，opus，mp3等，最常用的就是aac。并且我在**《音视频开发之音频原理》**的视频教程最后，介绍了一个音频的项目，开源地址：https://gitee.com/Vanishi/BXC\_AudioDemo，这个项目基于fdk-aac和sdl2两个依赖库，分别实现了解码音频aac然后播放，直接解析pcm然后播放。如果有想要了解pcm和aac之间转换的朋友，可以参考一下，项目的依赖库都是配置好的，下载后可以直接用vs2019打开运行。**补充一点**，目前这个项目BXC\_AudioDemo里面写了基于fdk-aac库实现aac转pcm，没有写pcm转aac的案例，其实看了aac转pcm的代码，再结合一些网上搜索fdk-aac库的使用，很容写出pcm转aac。如果写不出，可以等几天，最近几天我会抽时间补充一下fdk-aac库实现pcm转aac，同时我还会补充一个ffmpeg库实现的pcm和aac之间相互转换的案例，所以大家可以关注我的这个开源项目，等待我的后续更新。

接下来我们先讲一讲aac的码流格式，以及如何将aac封装到RTP数据包，进行传输。

AAC音频格式有2种：

ADIF(Audio Data Interchage Format)，音频数据交换格式：只有一个统一的头，必须得到所有数据后解码，适用于本地文件。

ADTS(Audio Data Transport Stream)，音视数据传输流：每一帧都有头信息，任意帧解码，适用于传输流。

我们主要介绍ADTS。

[(33条消息) C++ 解析aac-adts的头部信息\_CodeOfCC的博客-CSDN博客\_如何判断是否带有adts头](https://blog.csdn.net/u013113678/article/details/123134860)

//ffmpeg命令行 从mp4视频文件提取aac 音频文件

ffmpeg -i test.mp4 -vn -acodec aac test.aac

备注：-i 表示输入文件

-vm disable video / 丢掉视频

-acodec 设置音频编码格式

//ffmpeg 从aac音频文件解码为pcm音频文件

ffmpeg -i test.aac -f s16le test.pcm

备注：-i 表示输入文件

-f 表示输出格式

//ffplay 播放.pcm音频文件

ffplay -ar 44100 -ac 2 -f s16le -i test.pcm

备注：-i 表示指定的输入文件

-f 表示强制使用的格式

-ar 表示播放的音频数据的采样率

-ac 表示播放的音频数据的通道数

AAC共有9种规格，以适应不同的场合的需要：

MPEG-2 AAC LC 低复杂度规格（Low Complexity） 注：比较简单，没有增益控制，但提高了编码效率，在中等码率的编码效率以及音质方面，都能找到平衡点

MPEG-2 AAC Main 主规格

MPEG-2 AAC SSR 可变采样率规格（Scaleable Sample Rate）

MPEG-4 AAC LC 低复杂度规格（Low Complexity）---现在的手机比较常见的MP4文件中的音频部份就包括了该规格音频文件

MPEG-4 AAC Main 主规格 注：包含了除增益控制之外的全部功能，其音质最好

MPEG-4 AAC SSR 可变采样率规格（Scaleable Sample Rate）

MPEG-4 AAC LTP 长时期预测规格（Long Term Predicition）

MPEG-4 AAC LD 低延迟规格（Low Delay）

MPEG-4 AAC HE 高效率规格（High Efficiency）---这种规格适合用于低码率编码，有Nero ACC 编码器支持