RTMP(flv) 封装H264、AAC

RTMP和FLV格式友好的兼容性，主要体现在RTMP封装可播放的音视频流时，仔细研究一下，你会发现，RTMP Packet中封装的音视频数据流时，其实和FLV/tag封装音频和视频数据的方式是相同的，所以，我们只需要按照FLV封装H264和AAC的方式，即可生成可播放流。只不过视频相关包包括：onMetaData，AVCDecoderConfigurationRecord，h264数据。和flv文件中的区别是flv文件用tag头，rtmp中相关信息是放在rtmp头中。至于内部内容是一样的。友好的兼容性带了的便捷，比如RTMP流媒体服务器可以直接将RTMP Packet作为FLV/tag写入文件，只需要稍微处理一些FLV头既可以保存成FLV文件。

1.RTMP/flv 的封装视频、音频数据相同。

RTMP 封装 Message Header + 音视频数据

Flv 封装 Tag头+音视频数据。

1. RTMP封装H264 视频数据格式
   1. 消息格式:

注: fmt为chunk第一个字节高两位

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 域名 | 域名1 | Video Data | 值 | 注释 |
| RTMP Message 头 | RTMP消息头 | 3bytes: timestamp delta, |  | fmt=0,1,2 |
| 3bytes: payload length, |  | fmt=0,1 |
| 1bytes: message type, | SetChunkSize 0x01  AbortMessage 0x02  Acknowledgement 0x03  UserControlMessage 0x04  WindowAcknowledgementSize 0x05  SetPeerBandwidth 0x06  EdgeAndOriginServerCommand 0x07  AMF3CommandMessage 17 // 0x11  AMF0CommandMessage 20 // 0x14  AMF0DataMessage 18 // 0x12  AMF3DataMessage 15 // 0x0F  AMF3SharedObject 16 // 0x10  AMF0SharedObject 19 // 0x13  AudioMessage 8 // 0x08  VideoMessage 9 // 0x09  AggregateMessage 22 // 0x16 | fmt=0,1 |
| 4bytes: stream id, |  | fmt=0 |
| RTMP Message  消息负载 | 负载头 | 4bits FrameType | 1 : key frame (for AVC, a seekable frame)  2 : inter frame (for AVC, a non-seekable frame)  3 : disposable inter frame (H.263 only)  4 : generated key frame (reserved for server use only)  5 : video info/command frame | 负载头   1. 帧类型 2. 编解码ID |
| 4bits CodecId | 1:jpeg(当前没有使用)  2 : Sorenson H.263  3 : Screen video  4 : On2 VP6  5 : On2 VP6 with alpha channel  6 : Screen video version 2  7 : AVC |
| 负载 | 1Byte AVCPacketType | 0:AVC sequence header  1:AVC NALU  2: AVC end of sequence (lower level NALU sequence ender is not required or supported) |  |
| 3Bytes CompositionTime | If(AVCPacketType==1)  时间偏移  else  0 |
| n字节Video Data | H264视频数据 |  |
|  |

2.RTMP封装AAC 音频数据格式

2.1 消息格式：

注: fmt为chunk第一个字节高两位

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 域名 | 域名1 | AAC Data | 值 | 注释 |
| RTMP Message 头 | RTMP消息头 | 3bytes: timestamp delta, |  | fmt=0,1,2 |
| 3bytes: payload length, |  | fmt=0,1 |
| 1bytes: message type, | SetChunkSize 0x01  AbortMessage 0x02  Acknowledgement 0x03  UserControlMessage 0x04  WindowAcknowledgementSize 0x05  SetPeerBandwidth 0x06  EdgeAndOriginServerCommand 0x07  AMF3CommandMessage 17 // 0x11  AMF0CommandMessage 20 // 0x14  AMF0DataMessage 18 // 0x12  AMF3DataMessage 15 // 0x0F  AMF3SharedObject 16 // 0x10  AMF0SharedObject 19 // 0x13  AudioMessage 8 // 0x08  VideoMessage 9 // 0x09  AggregateMessage 22 // 0x16 | fmt=0,1 |
| 4bytes: stream id, |  | fmt=0 |
| RTMP Message 消息负载 | 负载头 | 4bits soundFormat | 0: Linear PCM, platform endian  1: ADPCM  2: MP3  3: Linear PCM, little endian  4: Nellymoser 16 kHz mono  5: Nellymoser 8 kHz mono  6: Nellymoser  7: G.711 A-law logarithmic PCM  8:G.711 mu-law logarithmic PCM  9: reserved  10: AAC  11:Speex  14: MP3 8 kHz  15:Device-specific sound | 负载头   1. 帧类型 2. 编解码ID |
| 2bits SoundRate | 0:5.5 kHz = 5512 Hz  1:11 kHz = 11025 Hz  2:22 kHz = 22050 Hz  3:44 kHz = 44100 Hz |
| 1bit SoundSize | 0:8-bit samples  1:16-bit samples |
| 1bit SoundType | 0:Mono sound  1:Stereo sound |
| 负载 | 1Byte AACPacketType | 0:AAC sequence header  1:AAC raw |  |
| n字节AAC Data | If(AVCPacketType==0)  AudioSpecialConfig  else  Raw AAC Data |
|  |
|  |