移动设备硬件编码直播实践

如何用mp4文件进行直播

一下科技 邓铮

如何理解标题

- 标题：移动设备硬件编码直播实践
- 移动设备——智能手机 Android / iPhone
- 硬件编码——利用移动设备自带的编码芯片录制视频
  - 耗电少
  - 视频清晰（码流可以很高）
- 直播——正在拍什么，别人就能看什么

名词解释

名词解释

- 大B和小b：B for Byte，b for bit，1字节＝8位
- 码流/码率：单位是bps(bit pre secord),一秒钟有多少数据
- 2/10/16进制：二进制 0000 1111 =>0x0F=>十进制的15
- 帧率：单位是fps(frame pre secord),一秒钟有多少幅画面
- I、P、B帧：关键帧、参考帧、双向参考帧

IP

第一帧

第二帧

I帧

P帧

视频格式

Mp4究竟是什么？

- ISO发布的标准
  - MPEG-1 VCD
  - MPEG-2 DVD
  - MPEG-4 ISO 14496
- ITU 发布的标准
  - H261
  - H263 ≈ ISO 14496-2,DivX,Xvid
  - H264/AVC/ISO 14496-10 蓝光DVD
- Mp4究竟是什么？
  - ISO 14496-14：MPEG-4文件格式，这是常见的mp4的说法

音频格式

- AMR / AMR-NB(4-10k) / AMR-WB(6-23k)
- MP3(MPEG Audio Layer 3)
- AAC(Advanced Audio Coding)/HE-AAC

文件（容器）格式

- AVI（MS）
- RM/RMVB（Real）
- FLV（Adobe）
- 3GP / MOV / MP4格式（MPEG-4 Part 14）
  - 3GP是MP4格式的一种简化版本
  - MOV（QuickTime，Apple）
  - 他们都是使用ISO 14496-12定义的文件格式

直播技术标准

- MMS（Windows Media Player）
- RTSP（类似HTTP交互方式，Real）
- RTMP（Adobe）
- HTTPFLV

天下武功出少林

- 都使用某种视音频格式进行编码
- 可封装到不同的文件格式中
- 各种直播协议都以TCP发数据为主，并加入辅助的流控
- 各种直播流格式和文件格式基本都可以互相转换
- 在不涉及到重编码的情况下，这种转封装的效率非常高

FLV一大简至美

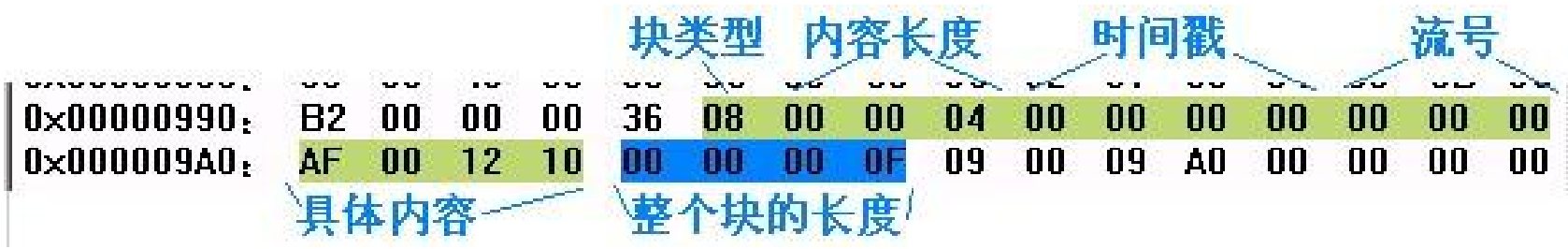
FLV一大简至美

- 每个FLV文件开头都是这13个字节
- 属性05表示文件中既有视频，也有音频

R	R	R	R	R	A	R	V
0	0	0	0	0	1	0	1

FLV—块定义

文件头	块头	内容	块尾	块头	内容	块尾	块头	内容	块尾



- 块头是固定的11字节，因此内容长度＋11＝块长度
- 类型只有08音频 09视频 12文本数据（metadata，cuepoint）

FLV—视频块

- 帧类型：
  - 0x1 P帧　　0x2 I帧　　0x3 B帧
- 编码格式：
  - 0x1 JPG　　0x2 H263　　0x4 On2 VP6
  - 0x7 AVC / H264

FLV—音频块

```
0x00000990:  B2  00  00  00  36  08  00  00  04  00  00  00  00  00  00  00
0x000009A0:  AF  00  12  10  00  00  00  0F  09  00  09  A0  00  00  00  00
```

编码格式 属性设置

- 编码格式：
  - 0x2、0xE MP3　0x4~6 Nellymoser　0x7~8 G711　0xA AAC
- 属性设置：

采 样	率	位长	声道

  - 采样率： 0x0 5.5kHz　0x1 11kHz　0x2 22kHz　0x3 44kHz
  - 位长： 0x0 8bit　0x1　16bit
  - 声道：0x0 mono　0x1　stereo

FLV—H264的NALU

- 由于H264编解码时需要设置很多基本参数，因此flv格式封装的h264特别空出第一个字节（图中红字）来描述数据内容是什么。
  - 其中00 后面的数据是NALU单元
  - 01 后面的是普通视频数据
- NALU里装着H264必须的SPS和PPS（序列、图像参数集）,可参照ISO 14496-15

FLV—AAC的SeqHeader

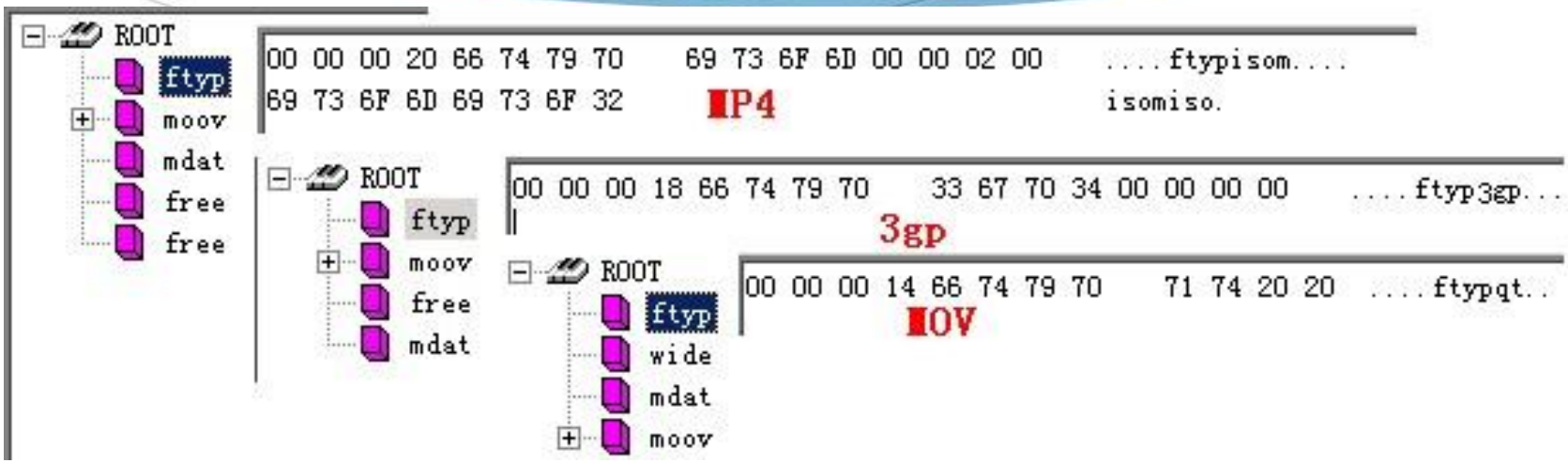
- AAC编解码时也需要设置一些基本参数，因此flv格式封装的AAC也空出第一个字节（图中红字）来描述数据内容是什么。
  - 其中00 后面的数据是Sequence Header
  - 01 后面的是普通音频数据
- SeqHeader定义可参照ISO 14496-3

MP4—有容乃大

MP4—Boxs

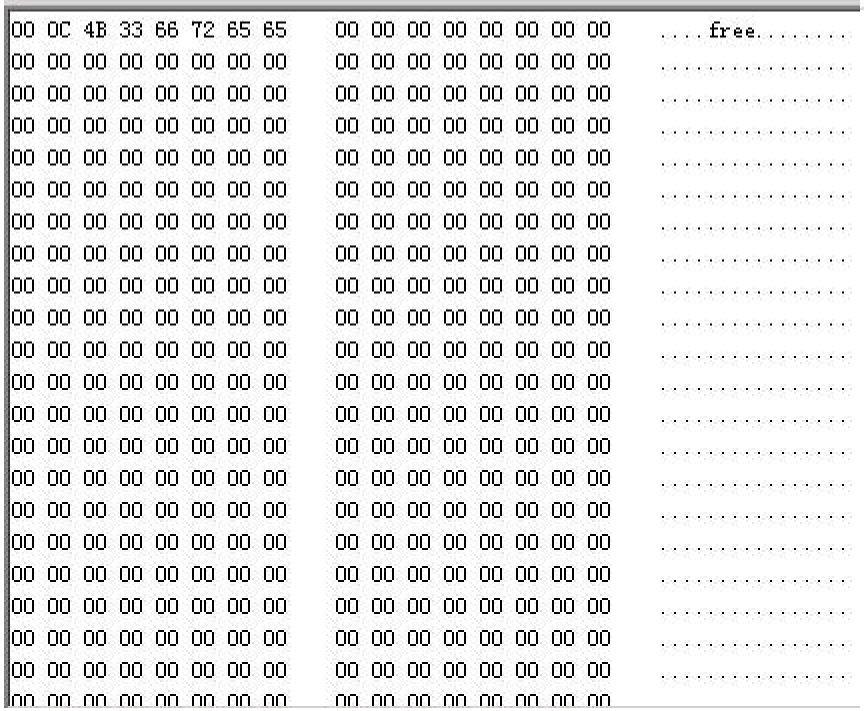
- Mp4文件由N个Box组成
- 每个Box都有一个4个字节的标识(atom)，该标识一般为易理解的4个英文字母
- 目前规定的Box种数已经有171种，详见 http://mp4ra.org/atoms.html

MP4—ftyp



- Ftyp为Mp4的第一个Box，标识文件是什么类型的文件，图中可见MP4，3GP，MOV的数据内容都是差不多的。

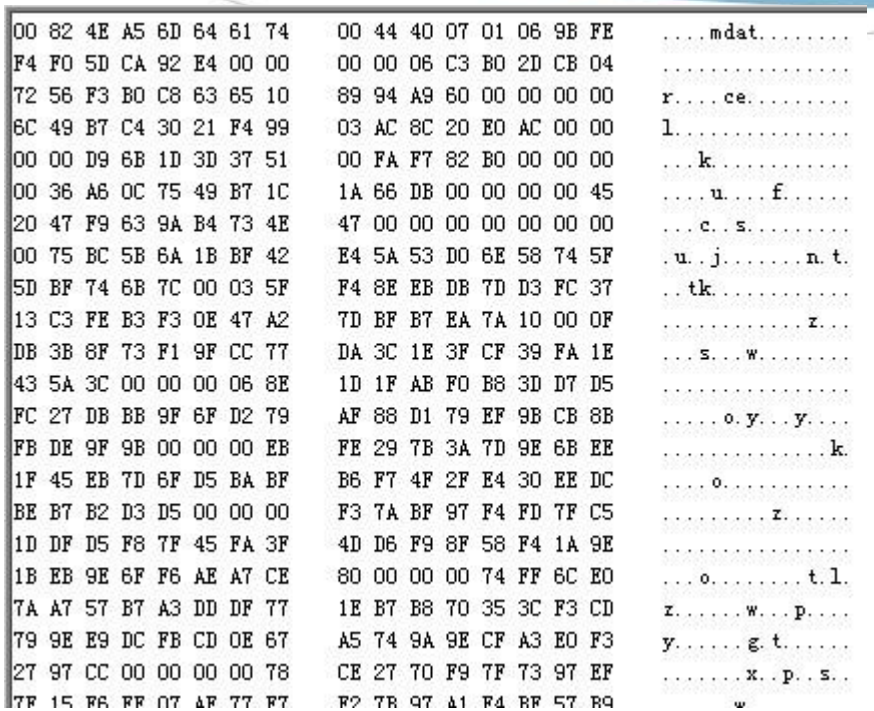
MP4—free



```
00 0C 4B 33 66 72 65 65  00 00 00 00 00 00 00 00   ....free........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

- Free box就是占坑用的。
- 在修改mp4文件时，这个box用处非常大，后面将会提到。

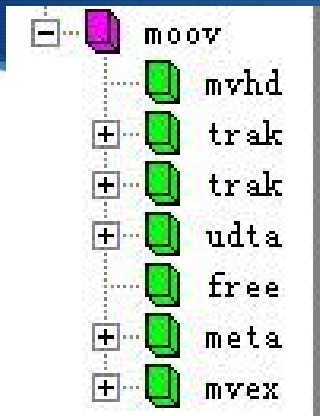
MP4—mdat



```
00 82 4E A5 6D 64 61 74   00 44 40 07 01 06 9B FE   ....mdat........
F4 F0 5D CA 92 E4 00 00   00 00 06 C3 B0 2D CB 04   ................
72 56 F3 B0 C8 63 65 10   89 94 A9 60 00 00 00 00   r....ce.........
6C 49 B7 C4 30 21 F4 99   03 AC 8C 20 E0 AC 00 00   l..............
00 00 D9 6B 1D 3D 37 51   00 FA F7 82 B0 00 00 00   ...k............
00 36 A6 0C 75 49 B7 1C   1A 66 DB 00 00 00 00 45   ....u....f......
20 47 F9 63 9A B4 73 4E   47 00 00 00 00 00 00 00   ...c..s.........
00 75 BC 5B 6A 1B BF 42   E4 5A 53 D0 6E 58 74 5F   .u..j.....n.t.
5D BF 74 6B 7C 00 03 5F   F4 8E EB DB 7D D3 FC 37   ..tk............
13 C3 FE B3 F3 0E 47 A2   7D BF B7 EA 7A 10 00 0F   ............z...
DB 3B 8F 73 F1 9F CC 77   DA 3C 1E 3F CF 39 FA 1E   ...s...w........
43 5A 3C 00 00 00 06 8E   1D 1F AB F0 B8 3D D7 D5   ................
FC 27 DB BB 9F 6F D2 79   AF 88 D1 79 EF 9B CB 8B   .....o.y...y....
FB DE 9F 9B 00 00 00 EB   FE 29 7B 3A 7D 9E 6B EE   .............k.
1F 45 EB 7D 6F D5 BA BF   B6 F7 4F 2F E4 30 EE DC   ....o.....
BE B7 B2 D3 D5 00 00 00   F3 7A BF 97 F4 FD 7F C5   .........z......
1D DF D5 F8 7F 45 FA 3F   4D D6 F9 8F 58 F4 1A 9E   ................
1B EB 9E 6F F6 AE A7 CE   80 00 00 00 74 FF 6C E0   ...o........t.l.
7A A7 57 B7 A3 DD DF 77   1E B7 B8 70 35 3C F3 CD   z.....w...p....
79 9E E9 DC FB CD 0E 67   A5 74 9A 9E CF A3 E0 F3   y.....g.t.......
27 97 CC 00 00 00 00 78   CE 27 70 F9 7F 73 97 EF   .......x..p..s..
7F 15 F6 FF 07 AF 77 F7   F2 7B 97 A1 F4 BF 57 B9   ..........w
```

- 顾名思义，用于存放多媒体数据的Box。
- 这个Box的大小一般和文件的大小差不多。
- 文件的视频，音频数据都杂乱无章地，穿插着存放在这里！！

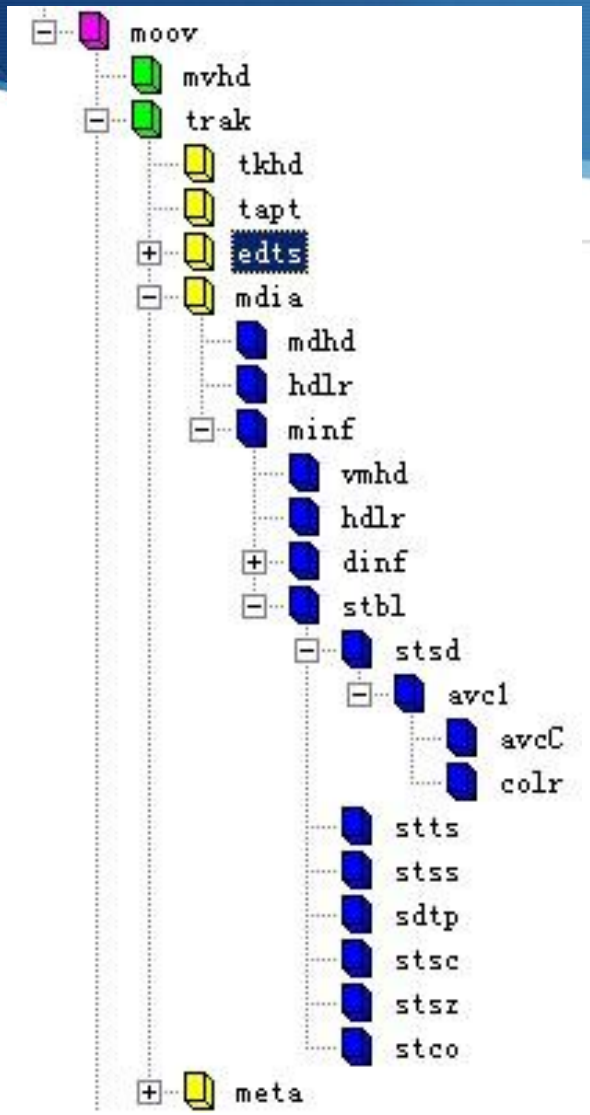
MP4—moov




```
00 00 1A 51 6D 6F 6F 76    00 00 00 6C 6D 76 68 64    ....moov...lmvhd
00 00 00 00 CA DF D4 E0    CA DF D4 E7 00 00 02 58    ...............X
00 00 0E 95 00 01 00 00    01 00 00 00 00 00 00 00    ................
00 00 00 00 00 01 00 00    00 00 00 00 00 00 00 00    ................
00 00 00 00 00 01 00 00    00 00 00 00 00 00 00 00    ................
00 00 00 00 40 00 00 00    00 00 00 00 00 00 00 00    ....@...........
00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 00    ................
00 00 00 03 00 00 08 2E    74 72 61 6B 00 00 00 5C    ........trak...\
74 6B 68 64 00 00 00 0F    CA DF D4 E0 CA DF D4 E7    tkhd............
00 00 00 01 00 00 00 00    00 00 0E 95 00 00 00 00    ................
00 00 00 00 00 00 00 00    00 00 00 00 FF FF 00 00    ................
00 00 00 00 00 00 00 00    00 00 00 00 FF FF 00 00    ................
00 00 00 00 05 00 00 00    02 D0 00 00 40 00 00 00    ............@...
05 00 00 00 02 D0 00 00    00 00 00 44 74 61 70 74    ...........Dtapt
00 00 00 14 63 6C 65 66    00 00 00 00 05 00 00 00    ....clef........
02 D0 00 00 00 00 00 14    70 72 6F 66 00 00 00 00    ........prof....
05 00 00 00 02 D0 00 00    00 00 00 14 65 6E 6F 66    ............enof
```

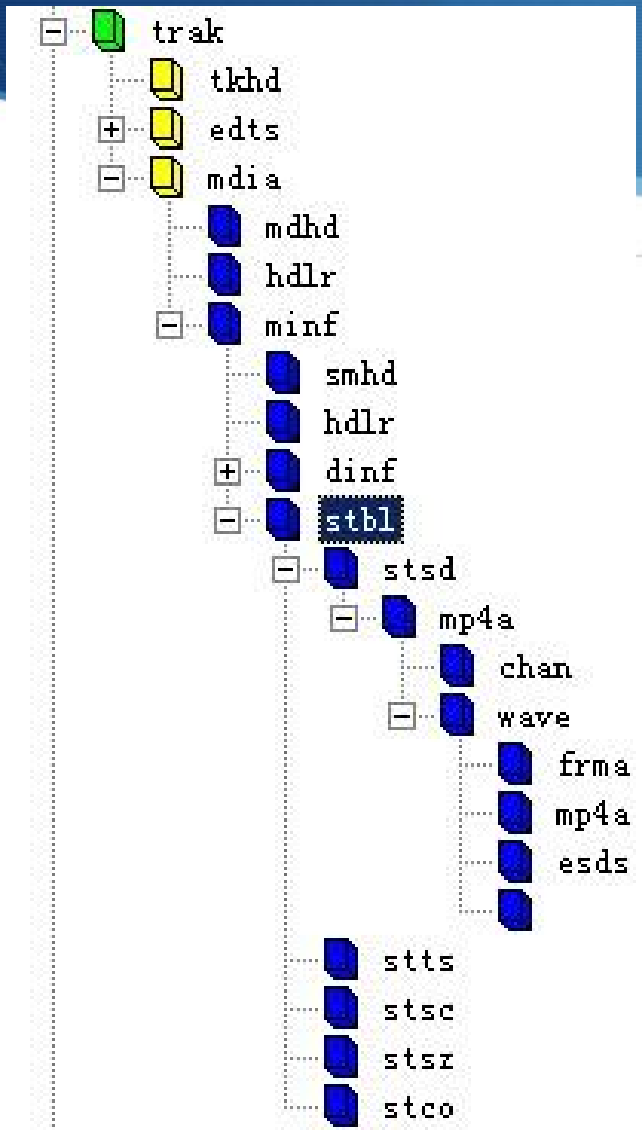
Moov记录了视频文件中的关键信息，里面包含了很多Box。只需要处理mvhd，trak两个Box即可，mvhd是moov的头部，保存时长等信息，trak是mp4里的多条轨道，一般是视频一条，音频一条

MP4—视频trak



- tkhd是trak的头部，保存轨道时长，宽高等信息
- mdia里保存了轨道中保存的媒体的信息
- mdhd是mdia的头部，保存TimeScale等
- hdlr表明本轨道类型，是视／音频等
- minf是具体的媒体信息
- minf里的stbl是最重要的Box！
- 其中stsd里的avc1里的avcC，保存了刚才我们提到的类似NALU的单元，里面放着SPS和PPS！！

MP4—音频trak



- 同样深入地，aac的seq头在trak->mdia->minf->stbl->stsd->mp4a里！！

MP4—最重要的stbl

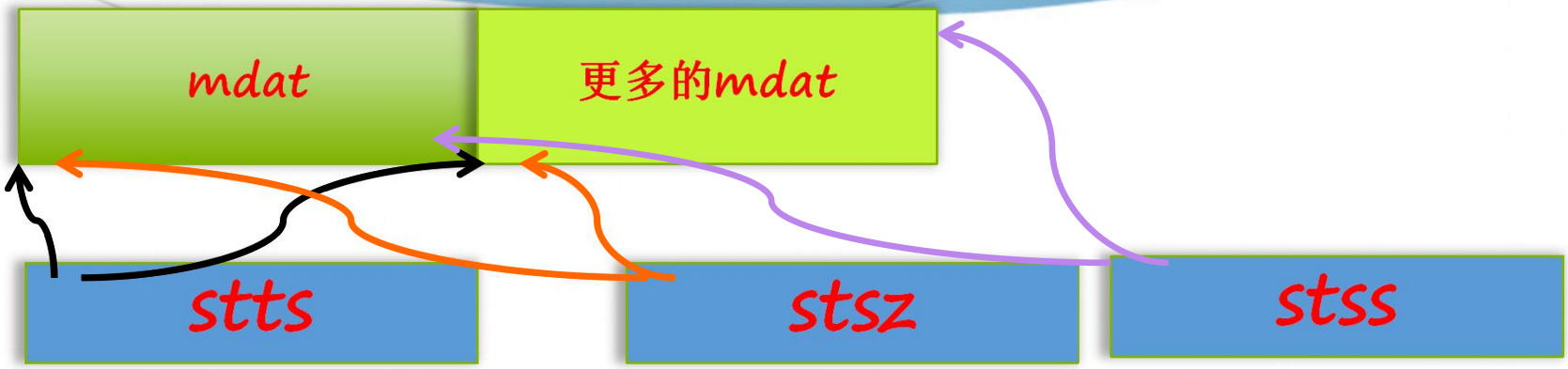
- Stts：time-to-sample，通过时间戳定位sample
- Stsc：sample-to-chunk，通过sample找到对应的块
- Stsz：sample size，每个sample的大小
- Stz2：sample size，更省空间的stsz表示
- Stss：chunk offest，每个chunk的偏移
- Co64：chunk offest，64位的偏移，超过4g的文件需要

硬编码的各种不可能

手机硬件支持的编码格式

- Android／iPhone
  - 都支持H.264编码
    - iPhone有底层接口能取到H.264裸流，生成MOV格式
    - Android只能设置输出文件名，通过Socket方式可不保存文件到磁盘，只能生成MP4／3GP格式
  - 都支持AAC和AMR编码

MP4格式—天生点播的命



- 直播模式下，文件内容是不停顺序生成的，mdat块还好，直接写入实时视频，音频数据就可以了，最后再修改一下mdat头的长度即可。
- 可是stbl里的stts，stsz，stss等要悲剧了，每个都是不停增加的主，那怎办呢？

MP4格式一天生点播的命

- 在解码的时候，如果没有moov信息，解码器对mdat的数据是无能为力的。

系统相机是如何运作的？

- Android早期的机型，在进行录制时，moov整个box先不生成了，就放在内存里，在录制完成时，才把整个moov写到文件的最后。
- 但是这种文件moov在文件尾，那就非常不适合网络上点播，因此有专门的程序处理这个，例如：Mp4Box/ffmpeg选项 -movflags faststart

系统相机是如何运作的(2)?

- Android后期的机型，在开始的时候，保留了很大一段空间（例如0xc5c20），到最后保存moov的时候，就把它写到这里，然后多余的空间用free box顶替

iPhone呢？

ftyp
moov
mdat
…….
hoov(hoof)
mdat
…….

- iPhone也是一开始保留一段空间用于保存moov，但是比较小，只够一段使用。
- 用完了后，又重新生成一段moov和mdat，但是这时的box不叫moov了，叫hoov或hoof。
- 无论Android还是iPhone，对应数据的moov都是最后才写的，甚至没写之前，文件头还是坏的。

总结一下

- 硬件都能且只能生成mp4 / 3gp / mov格式的视频文件
- 在直播期间，只有不断生成的mdat数据，数据是视频 / 音频混合在一起的
- 用于指示mdat数据块的区间，编码必须的SPS、PPS、AAC seq header，无一例外地都保存在最后才生成的moov box

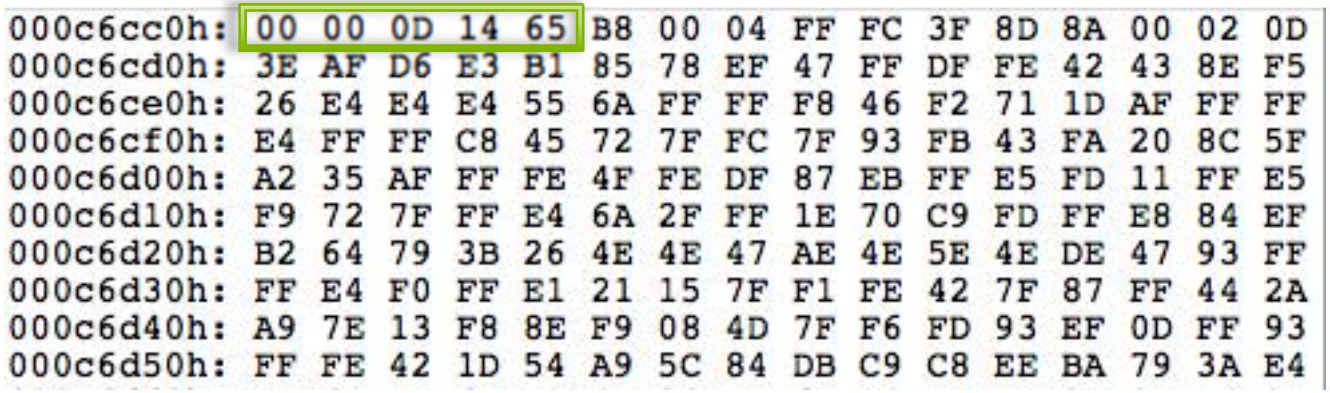
艰难地实现直播

艰难一——SPS，PPS等

- SPS，PPS等的生成算法非常复杂，而且需要知道各种编码器默认参数是怎设置的，基本上可以放弃自行计算
- 同样的参数，同样的机型，得到的取值是一样的
- 因此在第一次直播完成后，重新扫描一遍文件，即可把参数保存下来
- 对于热门机型，通过文件上传后，在服务器上批量分析相同机型的原视频，达到大致准确的比例后，把参数录入到系统中，则该种机型都可以一开始就支持了

艰难二——如何获取直播数据

- 生成mp4文件的时候，可以通过文件实时监测得到最新写入的数据
- 通过mp4封装的h264裸流格式的分析，可以得出一些规律：
  - H264每个块都以4字节的数据描述该块有多长开始，最大的大小可达0xFFFFF
  - 每帧的数据开始的一个字节，会描述帧的一些设置，和是否为关键帧等，可能出现的值只有61 65 41 45 21 25 B8几种



艰难三——不确定的状态机

- 开始扫描文件，寻找mdat box
- 找到mdat后，判断开始是视频还是音频
- 如果是视频数据，直接等待该块完成
- 如果是音频数据怎办？

艰难四——怎样判断音频块

- AMR，MP3格式都是定长的，处理起来非常方便
- 但是AAC是使用bit计算的，没有任何整个字节的头尾特征
- 只能完全实现AAC的解码器，尝试对数据进行解码，如果能够解码成功，说明就是音频数据
- 从ffmpeg的AACParser代码进行提炼，大概5000行代码，实现了一个黑盒处理器，输入是原始数据，输出是其中多少字节是音频数据。
- 如果到了截止的数据，刚好符合刚才视频的判断规则，则接下来是视频帧，否则继续等待新的数据，解出音频数据

艰难五——音频块的时间戳

- 音频数据是不间断的，音频的时间戳会非常准确。
- 因为AAC通用配置都是44100kHz，双声道，16bit采样，生成的数据是均匀的
- 计算得出音频块每次时间间隔递增为0x17毫秒
- 在每增加5次和50次的时候需要补偿一下，自增1毫秒，即可基本准确实现时间戳递增

艰难六——视频块的时间戳

- 视频数据的时间戳，由于不同机型每次吐出的数据帧数不一样，只能在两次吐数据的间隔进行平均

  前两次生成块的间隔时间—>预测下一块间隔时间

- 在单次吐的数据里面，如果出现多个数据块，每次增加40毫秒

- 如果视频时间戳离音频时间戳偏移过大，则用音频时间戳慢慢增加或减少40毫秒的增量，达到一个动态平衡

万事俱备

- 大简至美——重新封装FLV发布到服务器上
- web环境下直接使用HTTPFLV播放
- Android下使用ffmpeg软解码FLV播放，直播完成后，转封装出MP4文件给Android系统播放
- iPhone下使用hls播放，服务器实时把HTTPFLV转封装为TS流供iPhone使用，直播完成后，转码出MP4进行播放

谢谢！