**1、IPTV/UDP协议网速很慢或者无网络会导致UI卡死**

主要原因是amplayer起播时需要获取一定数据来探测节目的类型和信息，此时因为网速很慢或者无网络时amplayer起播会卡在网络等待接口，所以因为还没有完成初始化化会导致没办法立即结束播放流程。优化主要有三方面：

1. 减少没必要协议的探测

因为我们player支持和使用的协议是明确的，可以把不需要的协议探测屏蔽掉，减少协议探测的时间。主要修改是LibPlayer/amffmpeg/libavformat/utils.c文件，在switch\_table中屏蔽掉不需要支持的协议。

1. 减少网络接口等待的时间

我们当前UDP协议下载采用的不阻塞网络接口，通过selece阻塞来选取数据到达的网络接口，如果无数据1秒钟才会超时，这种设定对于网络正常时没有问题，因为UDP接收数据非常快，只要有数据到达我们就可以唤醒线程，如果此时UI需要切换节目我们可以及时检测到并退出下载，但是当网络不理想时会导致线程睡眠，最少1秒之后才能唤醒检测UI是否有退出下载的请求，这会导致UI请求退出时需要等待将近1秒，所以这里需要把超时时间减少到10ms，这样每隔10ms就可以检查一次是否有UI请求退出播放，以便能快速响应UI操作。主要修改文件：LibPlayer/amffmpeg/libavformat/udp.c, 修改等待时间如下：

FD\_ZERO(&rfds);

FD\_SET(s->udp\_fd, &rfds);

tv.tv\_sec = 0;

tv.tv\_usec = 10;

ret = select(s->udp\_fd + 1, &rfds, NULL, NULL, &tv);

1. 优化UI的切台逻辑，减少无必要的切台动作

用户有时候长按切台按键是为了快速切换的自己需要观看的节目，此时并不需要连续调用player接口去响应每一个按键去切台，因为长按切台时用户仅是想切换到最后一个自己需要观看的节目，中间的节目有没有播放用户并不关心，此时需要UI优化，用户长按切台按键时仅切换节目信息不需要实际播放节目，等待用户松开按键时，说明此时用户已选到自己需要观看的节目，这时候再实际执行切台动作。这样可以减少客户长按切台按键时中间无必要的返回节目播放。

1. **快速切台时导致内存溢出**

主要原因是因为快速切换节目时起播已经已经分配了AVFormatContext结构体，因为这时用户操作退出播放，切换到下一个节目，player会检测用户操作，检测到用户退出操作时player异常退出，这时候并没有把分配到的结构体释放或者赋值给playe导致player退出时检测不到AVFormatContext的值以为并没有分配成功，所以会导致AVFormatContext没有释放内存泄漏。主要修改文件：LibPlayer/amplayer/player/player\_ffmpeg\_ctrl.c，如果用户操作异常退出，不管是否分配成功都需要把指向AVFormatContext的指针赋值给player。Player退出时会作检测，如果AVFormatContext分配成功，则把分配的内在释放，如果不成功，则不作处理，具体修改如下：

if (url\_interrupt\_cb()) {

if (ret == 0) {

am\_p->pFormatCtx = pFCtx;

}

return FFMPEG\_OPEN\_FAILED;

}

1. **优化IPTV起播速度**

通过分析可以发现IPTV节目起播时需要花费大量时间用于检测流的信息，检测流信息花费的时间取决于读取TS流的速度，相对而言我们使用网络播放节目时一般速度都比较慢，所以导致部分节目特别是数据量较大的4k的节目起播较慢。所以我们只需要减少检测流信息的时间即可以加快IPTV的起播的速度，事实上amplayer已经提供了一个环境变量控制，即media.libplayer.fastswitch，当此环境变量设置为1时为快速起播模式，amplayer只检测播放节目必要的信息即开始播放节目，这样可以大大加快起播速度。但是存在一个问题，因为打开此模式时我们并检测节目的宽高比信息，导致有些节目使用错误的宽高比播放，所以我们需要修改，即检测到节目宽高比之后再启用快速切台模式，这样即提高起播速度也可以避免出现使用错误宽高比播放的问题。主要修改文件：LibPlayer/amffmpeg/libavformat/utils.c，在av\_find\_stream\_info函数中增加判断如果获取到了宽高比则马上进入fast\_switch。具体修改如下：

if ((!get\_aspect\_ratio) && (st->codec->sample\_aspect\_ratio.num != 0 && st->codec->sample\_aspect\_ratio.den != 0)) {

get\_aspect\_ratio = 1;

fast\_switch = 1;

}