1. 使用结构体struct drop\_fps targe\_fps;传参，通过类似于

targe\_fps.fps\_a = 7;

targe\_fps.fps\_b = 1;

req.rec.targe\_fps = &targe\_fps;

使能该功能

1. 只适用于帧率固定的摄像头，包括数字前视摄像头和模拟后拉摄像头（出来的帧率必须是固定的），不支持USB后拉摄像头，所以当使用USB后拉摄像头，这一路录像不支持该功能
2. 在录像时启动丢帧功能，可以参考代码（SDK里已添加）

struct drop\_fps targe\_fps;

if (req.rec.tlp\_time && (req.rec.camera\_type != VIDEO\_CAMERA\_UVC)) {

targe\_fps.fps\_a = 1000;

targe\_fps.fps\_b = req.rec.tlp\_time;

req.rec.targe\_fps = &targe\_fps;

}

在录像过程中，可以开关或者重新设置新的丢帧帧率，使用对应的函数（记住不支持USB后拉录像那一路）

static int video0\_rec\_set\_dr()

{

union video\_req req = {0};

struct drop\_fps targe\_fps;

if (!\_\_this->video\_rec0) {

return -EINVAL;

}

targe\_fps.fps\_a = 7;

targe\_fps.fps\_b = 1;

req.rec.targe\_fps = &targe\_fps;

req.rec.channel = 0;

req.rec.state = VIDEO\_STATE\_SET\_DR;

return server\_request(\_\_this->video\_rec0, VIDEO\_REQ\_REC, &req);

}

当req.rec.targe\_fps = NULL;表示关闭丢帧功能

1. 关于启动丢帧功能之后，封装器的帧率问题：封装器的帧率还是按原来的帧率（摄像头的原始帧率播放，例如30fps），所以播放时会有快进的感觉，如果想要让封装器按你实际的丢帧帧率播放，可以修改录像启动时的帧率req.rec.fps = targe\_fps.fps\_a / targe\_fps.fps\_b;
2. 关于声音的问题，一般来说，启动了丢帧功能之后，就不录声音，因为这时候的声音和视频是不同步的，当然，如果你需要在丢帧的时候录声音，也是可以的，只是声音和视频是不会同步的，因为实际的视频帧数已经减少,在使能该功能之后，SDK里默认是不录声音的。
3. targe\_fps.fps\_a是丢帧之后的帧率的分子，targe\_fps.fps\_b是丢帧之后的帧率的分母，例如你想丢帧之后的帧率为7帧，那么targe\_fps.fps\_a = 7;targe\_fps.fps\_b = 1;