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

2. 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

3. 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

4. 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;

5. 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。

6. 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;