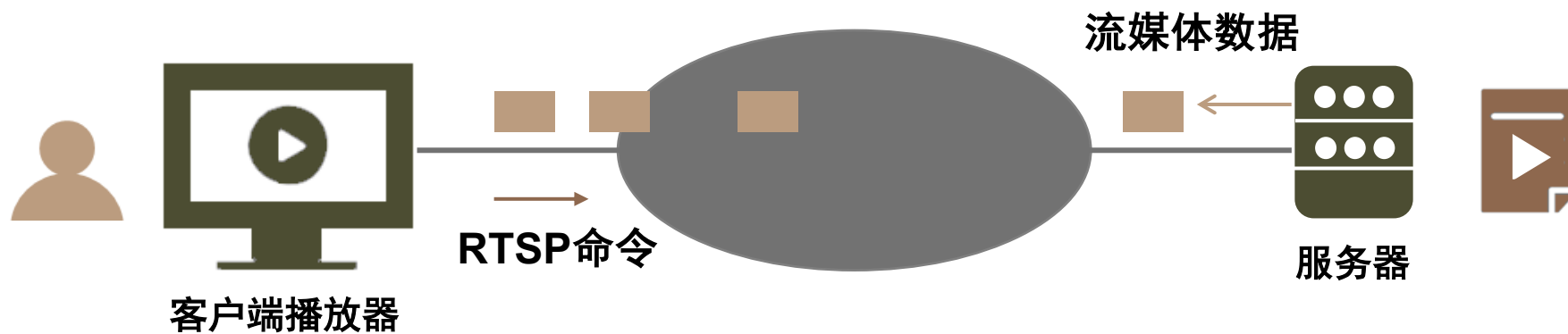


流媒体应用中的 播放器



流媒体应用中的播放器



- 播放器边播放流媒体边下载流媒体数据
- 媒体流数据通过流媒体传输协议从服务器传输到客户端
- 用户像播放本地音视频一样播放流媒体

流媒体播放器

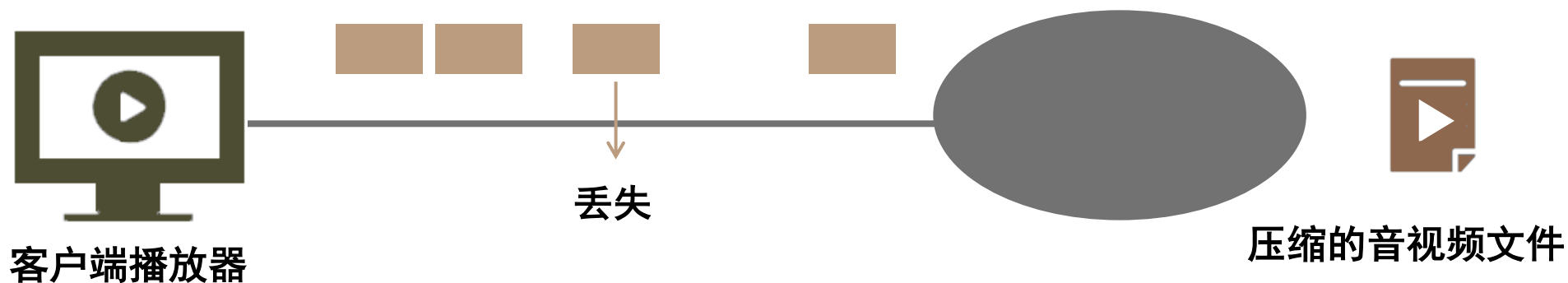
- 用户界面
- 解码流媒体
- 处理传输错误
- 消除抖动

媒体播放器的解压

解码：播放时把数据编码文件转为模拟的音视频信号的过程。

两大问题

- 若后面的流媒体数据早于之前的数据到达客户端，则无法解压
- 前面数据的丢失可能导致媒体流数据无法解码



压缩过程必须设计成不管丢不丢包都能进行解码。

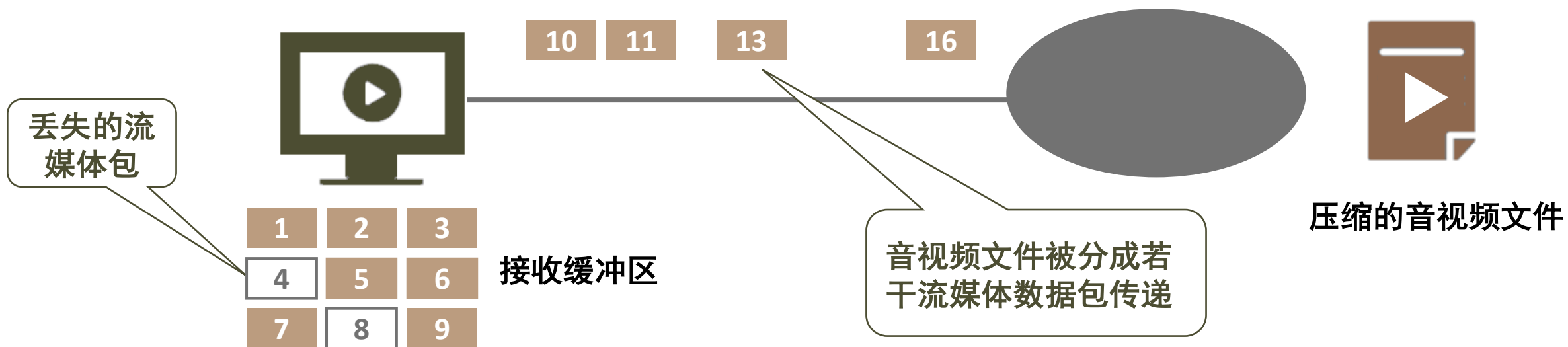
例如：MPEG压缩输出I帧、P帧和B帧



媒体播放器的差错控制

差错控制目的：主要克服大量丢包造成的图像不连续。

- ① 流媒体数据的少量丢失不会对用户体验带来太大影响
- ② 网络拥塞可能带来数据突发性丢失



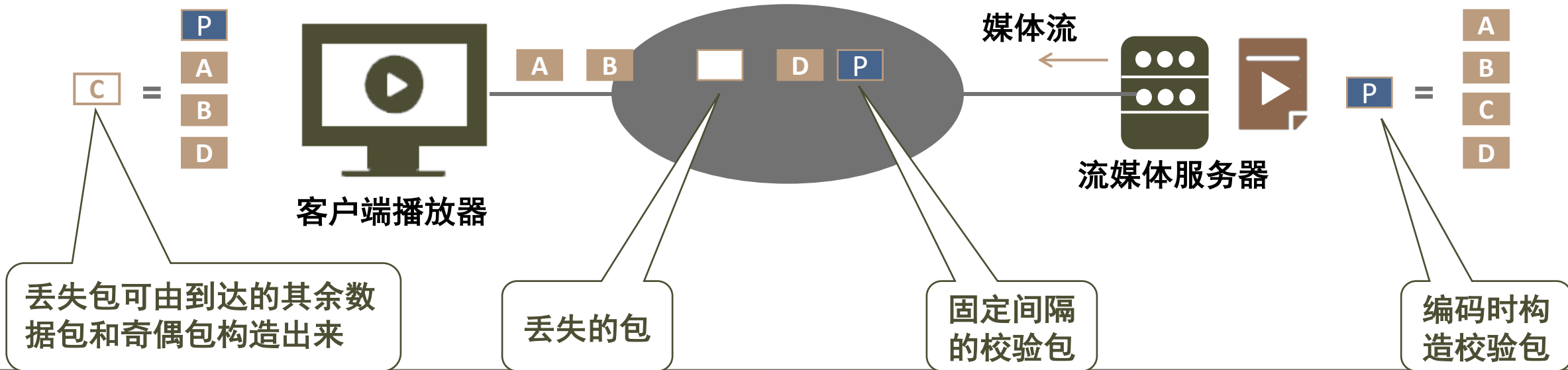
前向纠错策略

前向纠错（FEC）：一种简单的纠错编码。

- 定期构造由数据包生成的奇偶包
- 将数据包和奇偶包一起发送

前向纠错代价

- 只有奇偶包抵达才能重构丢失包
- 增加了额外的带宽开销



FEC: Forward Error Correction

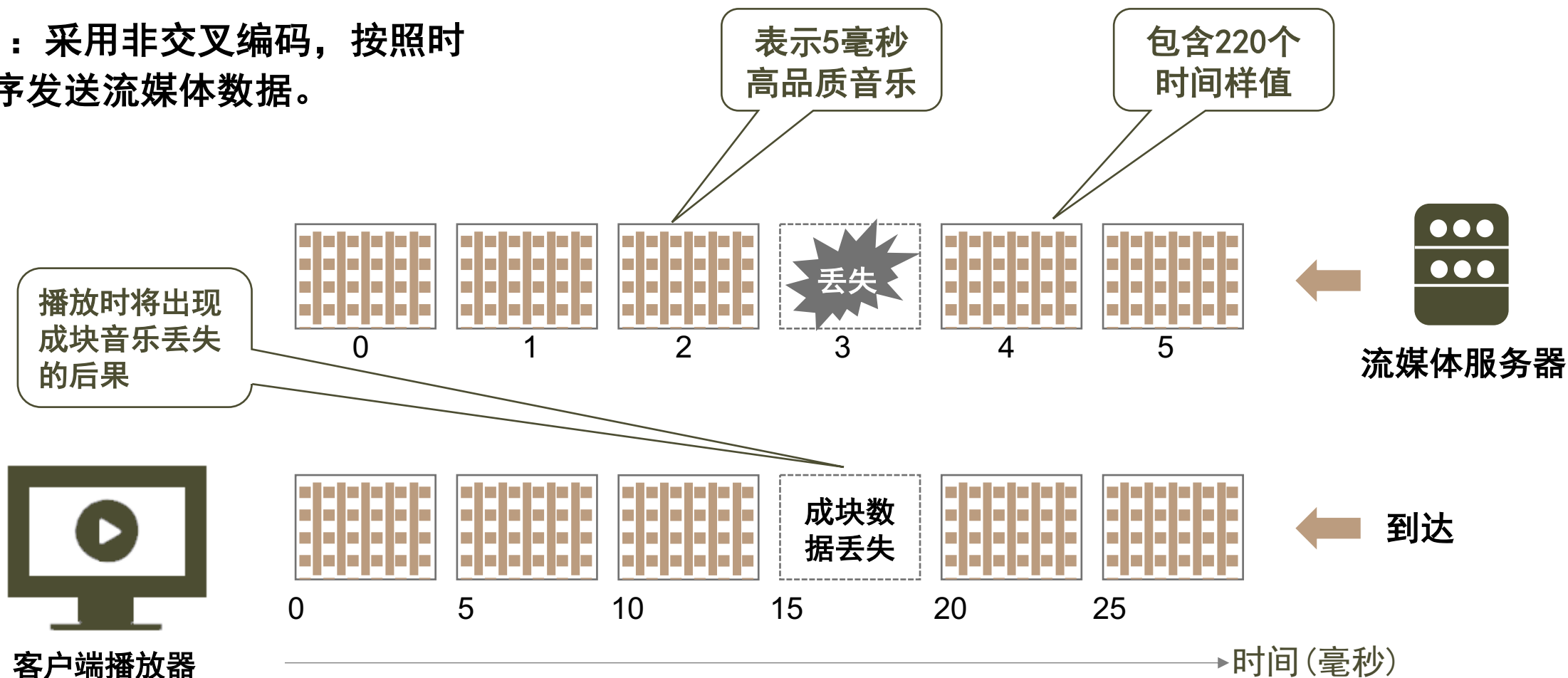
■ 流媒体包



北京大学

示例I：非交错编码

示例1：采用非交叉编码，按照时间顺序发送流媒体数据。



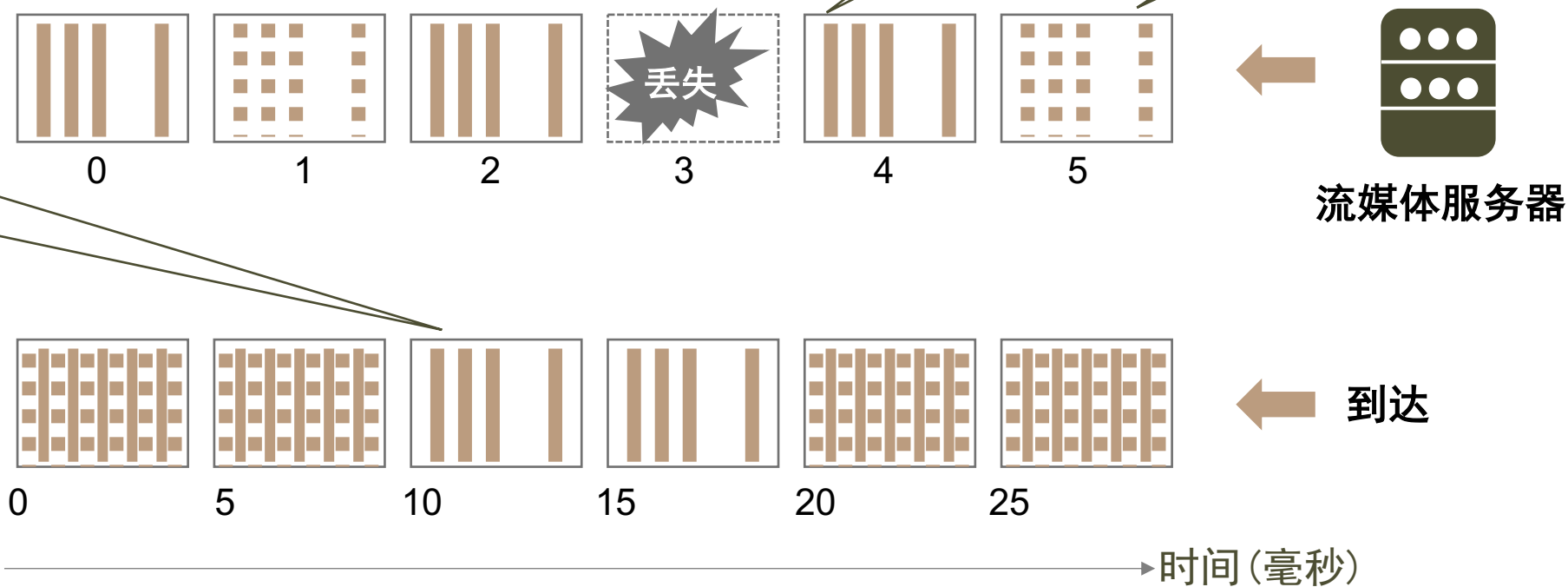
交错编码策略

交错编码：发送端将媒体流混合或者交叉编码，接收端做相反操作还原原始媒体流。

播放效果不会出现成帧丢失，只是降低播放质量。



客户端播放器



媒体流数据的实时性

媒体流数据的实时传输

- 服务器以播放速率通过UDP发送音视频，媒体播放器一旦收到媒体流立即解压播放。
- 通过TCP（尽可能快地）发送媒体流，把到达的媒体缓存在播放缓冲区中

每种媒体数据有自身的播放速率需求，为防止出现“无米可炊”的窘态，媒体播放器通常要延迟5~10秒再播放。

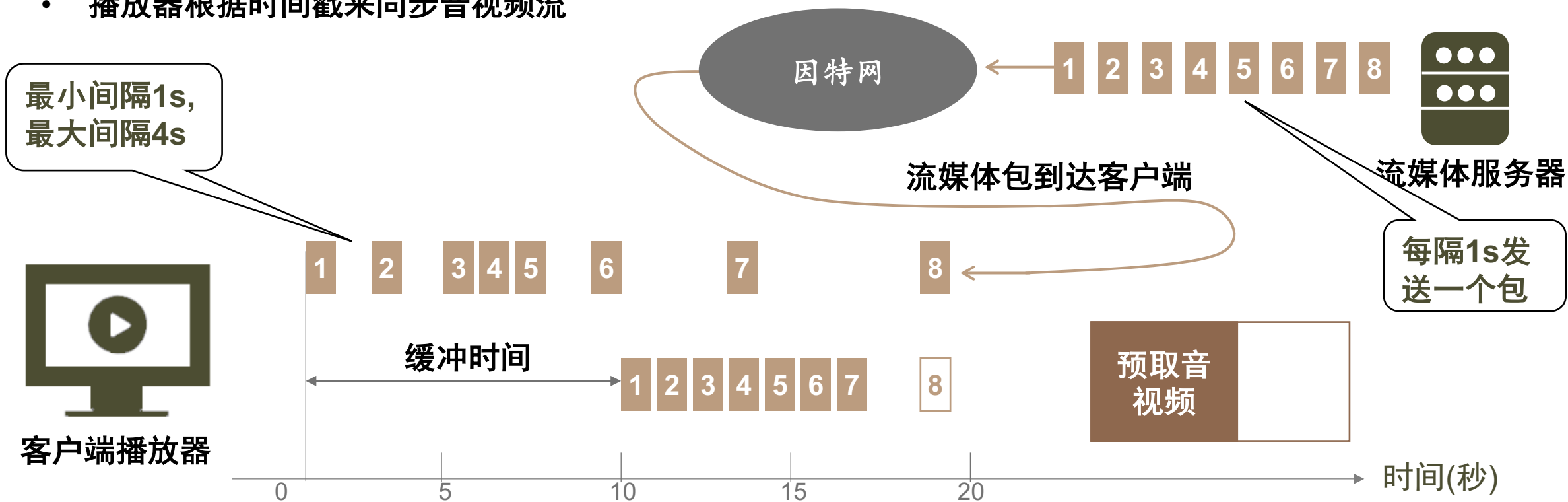


抖动对播放的影响

抖动 (jitter) : 前后两个包的延迟差

- 每个包都有一个相对于媒体流中第一个样本值的时间戳
- 播放器根据时间戳来同步音视频流

抖动消除 : 接收端通过延迟播放并缓冲包来消除包传输时延的变化。

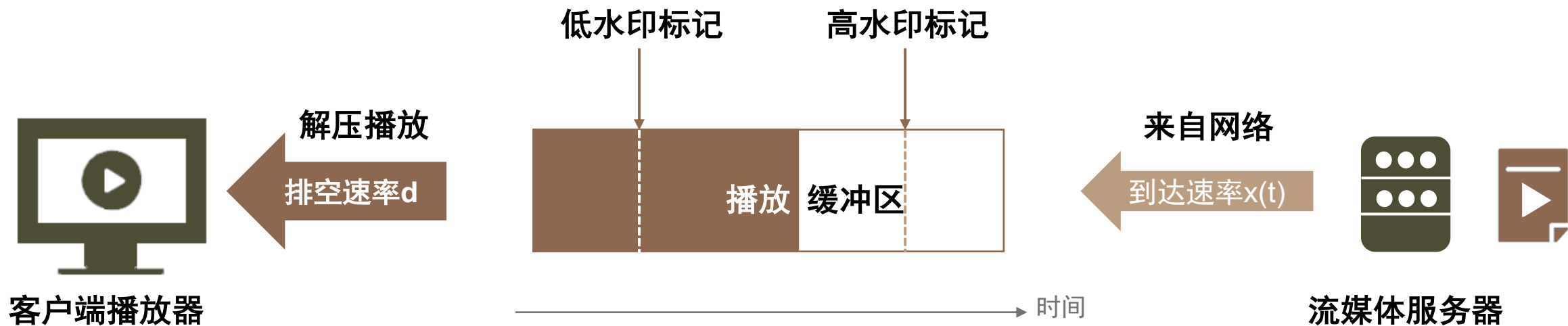


播放器消除抖动

播放器消除抖动

- 在播放缓冲区中设立高低水印标记
- 播放器根据高低水印标记控制服务器端媒体流数据的发送

- 当缓冲区内的媒体流数据量逼近高水印记时，客户端通知服务器暂停发送媒体流数据
- 当缓冲区内的媒体流数据量少于低水印记时，客户端通知服务器恢复发送媒体流数据



高低水印记与网络带宽的关系

?

高水印记和低水印记应该设置成多大才不会影响播放?

- 低水印记标记必须确保缓冲区内流媒体数据量大于2倍的网络时延*播放速率
- 高水印记标记必须确保缓冲区的剩余空间必须大于2倍的网络时延*端-端带宽

播放器以固定的速率 d 读取缓冲区内的媒体流数据



客户端播放器

排空速率 d

低水印标记

高水印标记



媒体流数据以网络带宽对应的速率到达客户机

来自网络
到达速率 $x(t)$



流媒体服务器



北京大学