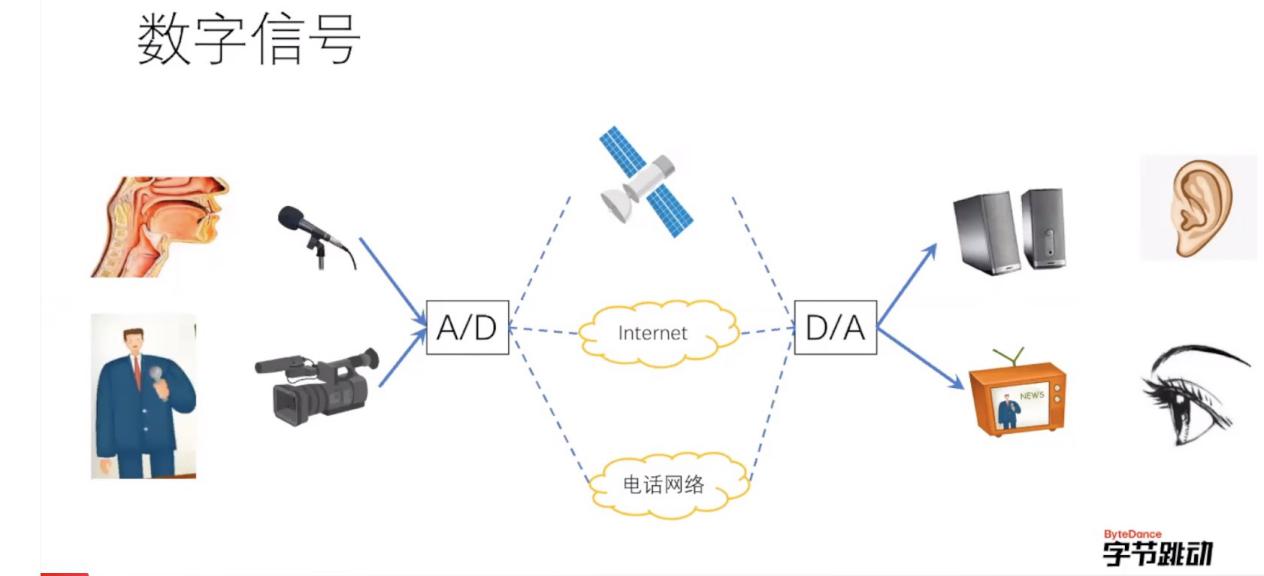
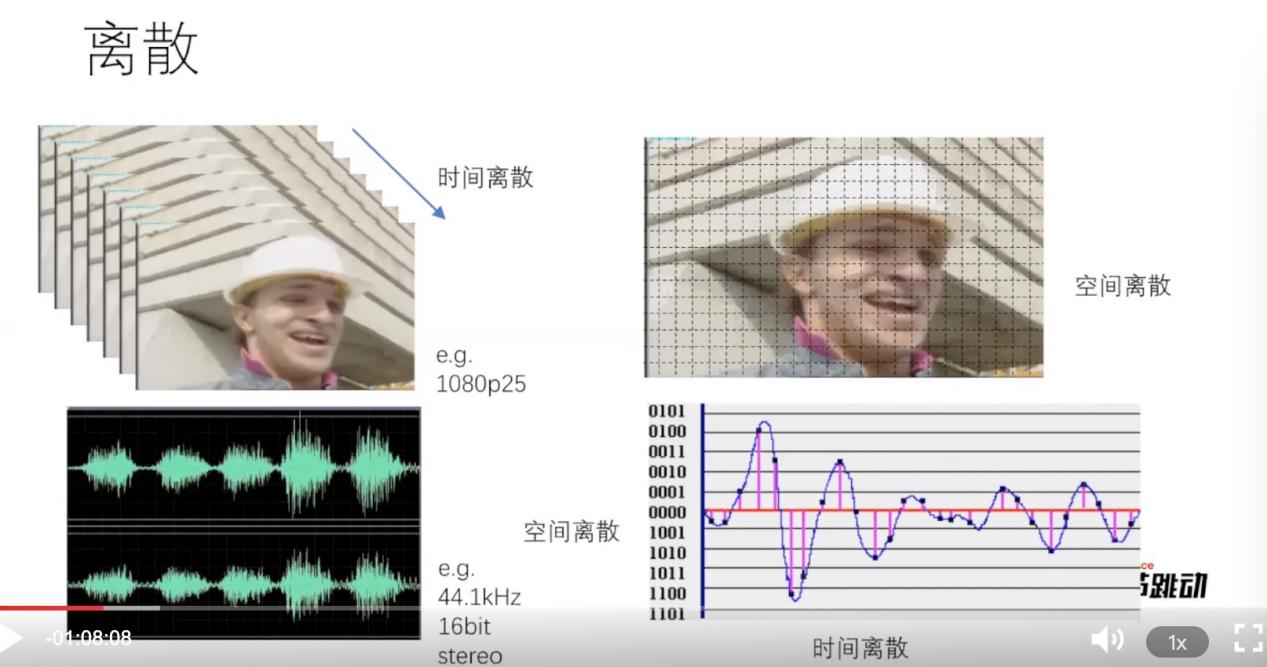
## 流媒体



音视频这种东西都是需要数模转换的，设备只能支持模拟的东西，中途传递的时候肯定是数字信号。

中途传递的信道有很多。

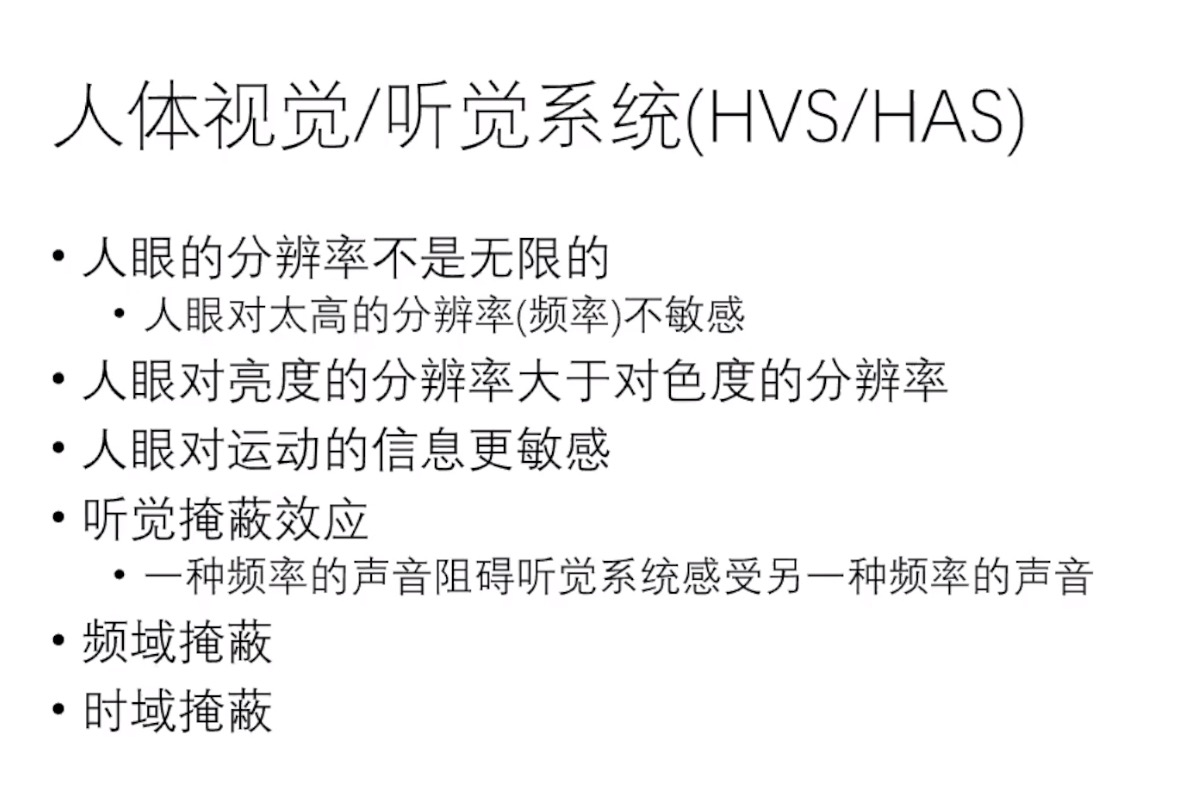


时间离散就是按照频率刷，导致人眼分辨不出来是不连续的。

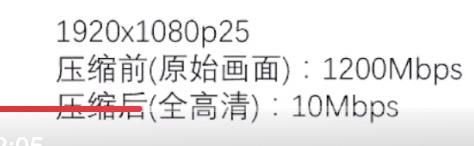
空间离散就是把每一个图在分成小块。

这东西就叫做 帧 视频中的帧。

其实比如一秒25帧，每一个像素还要记录下红黄蓝三原色的比例拼出色彩来，数据会很大，带宽扛不住，所以会根据人眼的特效去做数据压缩。



上面就是压缩的人体特效的依据。

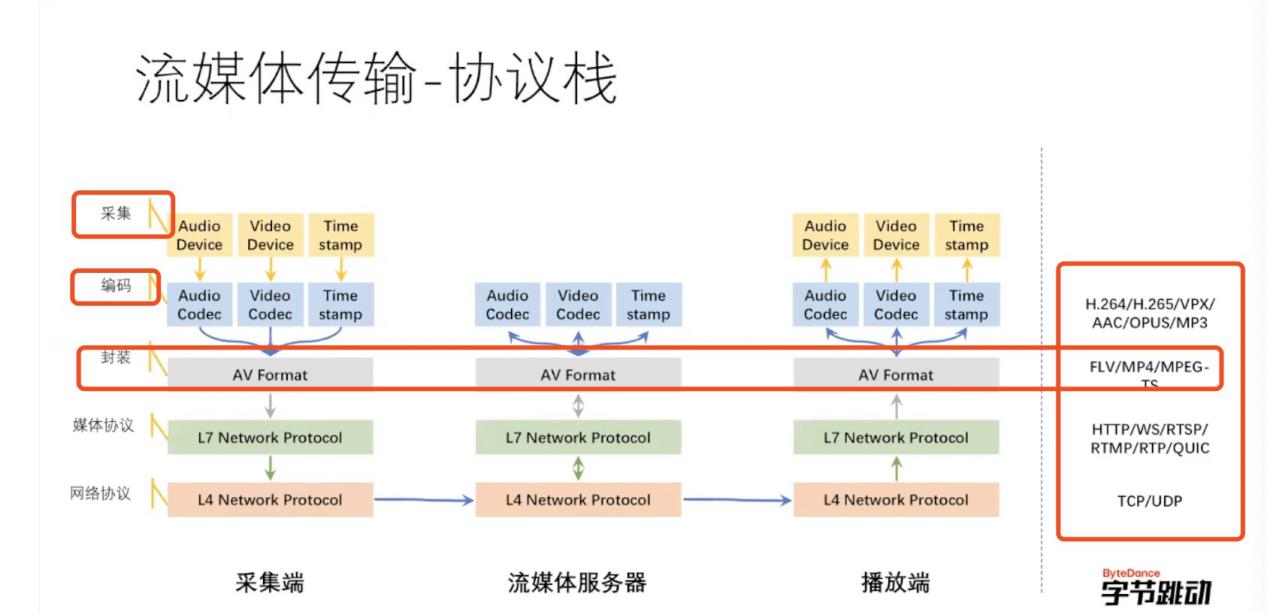


可以看出，如果不压缩的话，1080p的视频要1g每秒的带宽，压缩后就是10M了。

视频压缩 主要是帧间冗余，其实就是前后两帧内容差不多，就不用都存了，存一个就可以，还有就是一个帧的图像内，也会有很多重复的，存一个就可以，这个就有一个算法，找到和你差不多内容的东西，

帧间参考，帧内参考。

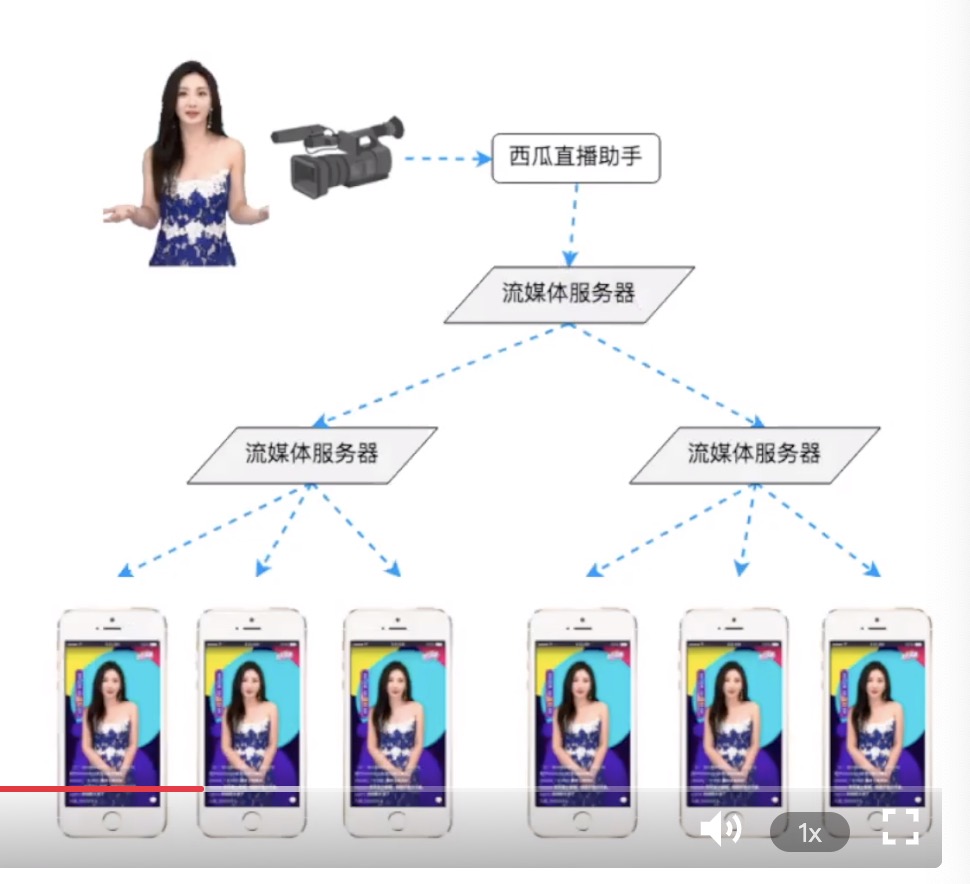
还有就是去掉一些人眼不那么敏感的东西，比如人眼对低频信息敏感，高频不敏感，我们就可以把高频的地方给干掉。



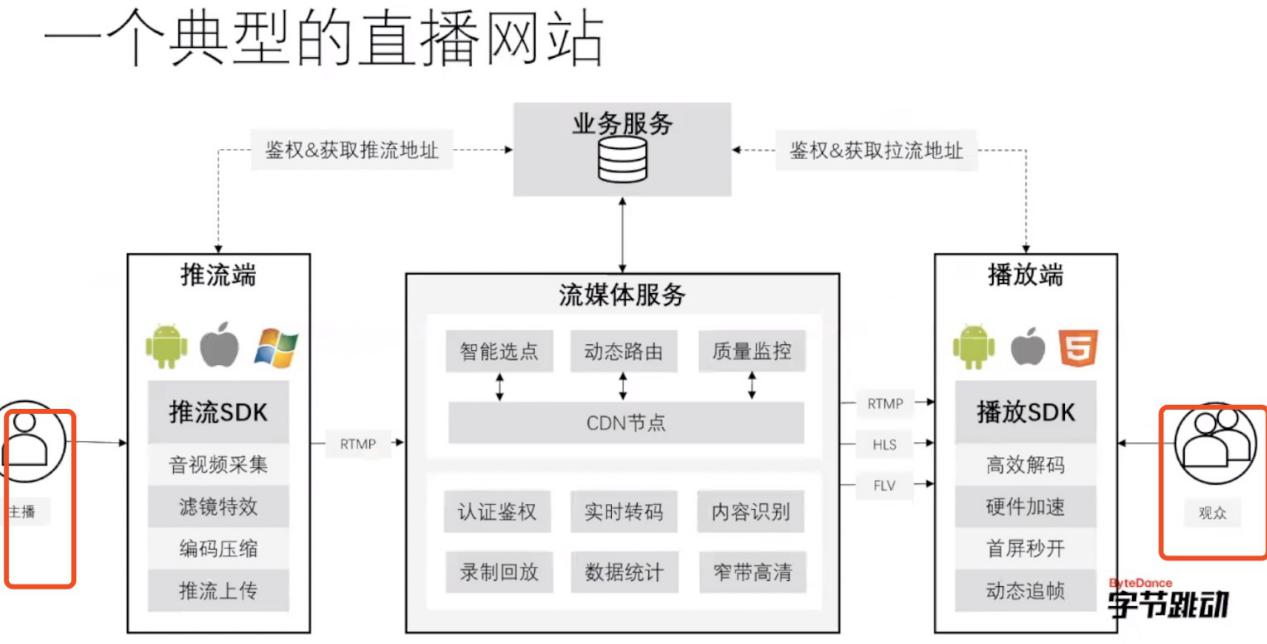
协议栈，可以看到http，rtp，rtmp，rtsp都是可以传输流媒体的应用层协议。

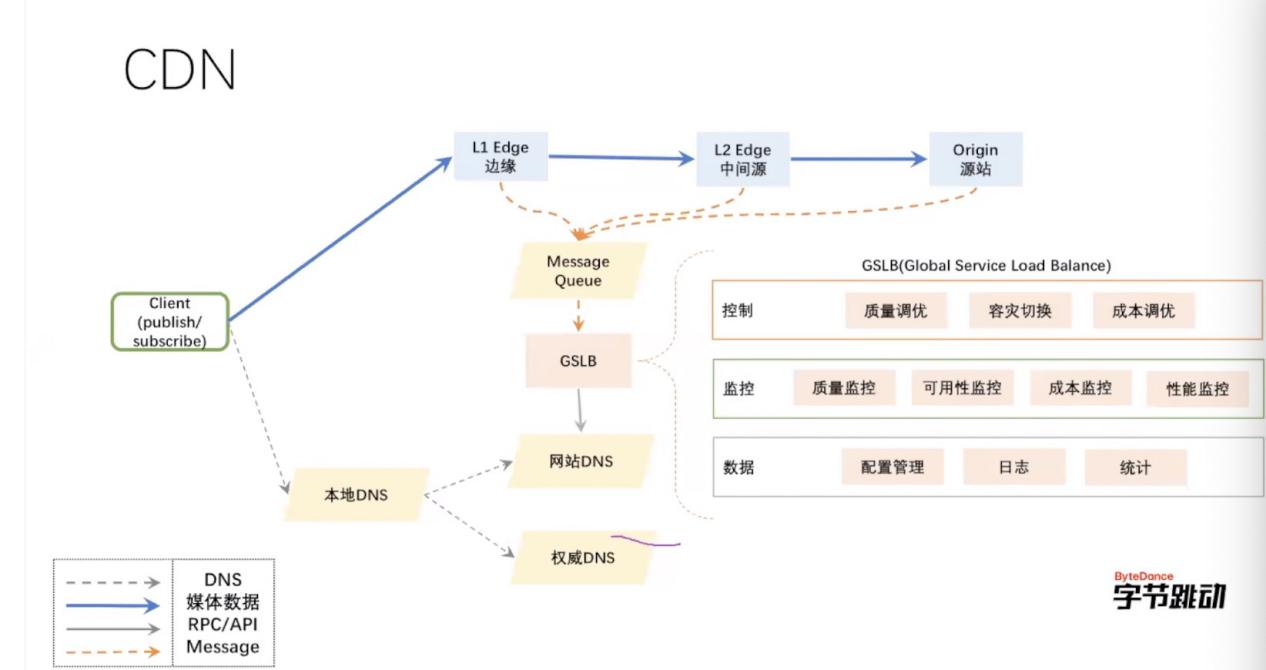
其实传输层udp和tcp都可以，tcp高保真，udp会丢到但是影响也不大，丢几帧也可以。

H264 h265是视频编码格式。



教师端每采集一帧就会发到流媒体服务器，流媒体服务器是上图那样分层级的，全国各地都有还要做cdn，使用户连到最近的流媒体服务器，流媒体服务器和用户端直接tcp连接不断开。





cdn咋做呢，其实cdn都是dns的时候做的，dns的时候可以获取你要访问的域名，和你的信息，通过这两个东西不就可以选节点了么。

本地dns是指运营商在某个镇或者区设置的dns服务器，会同步权威dns的信息。

上面说的流媒体服务器会分层，最下层就是边缘节点，最上层就是源站，也就是和用户连接的是边缘节点咯。

## RTP实时传输协议

RTP协议详细说明了在[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91" \t "/Users/yangwenshuo/Documents\\x/_blank)上传递音频和视频的标准数据包格式。

RTP协议和RTP控制协议[RTCP](https://baike.baidu.com/item/RTCP" \t "/Users/yangwenshuo/Documents\\x/_blank)一起使用，而且它是创建在[UDP协议](https://baike.baidu.com/item/UDP%E5%8D%8F%E8%AE%AE" \t "/Users/yangwenshuo/Documents\\x/_blank)上的。

rtcp是控制rtp的。

数据传输协议RTP，用于实时传输数据。该协议提供的信息包括：时间戳（用于同步）、序列号（用于丢包和重排序检测）、以及负载格式（用于说明数据的编码格式）。

控制协议RTCP，用于QoS反馈和同步媒体流。相对于RTP来说，RTCP所占的带宽非常小，通常只有5%。

RTP提供抖动补偿和数据无序到达检测的机制。由于IP网络的传输特性，数据的无序到达是很常见的。 RTP允许数据通过IP组播的方式传送到多个目的地。RTP被认为是在IP网络中传输音频和视频的基本标准。RTP通常配合模板和负载格式使用。

对于实时多媒体流应用，及时传送信息是首要目标，为达到目标可以忍受部分丢包。例如，在音频应用中的一个丢包，可能导致损失音频数据中的一秒内容，这个很容易通过合适的隐藏算法掩盖过去，从而不被人注意。由于TCP更注重可靠性而不是及时性，在RTP应用中很少使用。取而代之，大部分RTP实施是基于UDP的。

每一个多媒体流会建立一个RTP会话。一个会话包含带有RTP和RTCP端口号的IP地址。例如，音频和视频流使用分开的RTP会话，这样用户可以选择其中一个媒体流。形成会话的端口由其他协议（例如RTSP和SIP）来协商。RTP和RTCP使用UDP端口1024 - 65535。

## http flv

这两种也是可以播放视频的,只不过必须要整个视频都下载完才能播放.

## RTP RTCP

Real-time transport protocol

是用于internet针对多媒体数据的一种传输层协议,RTP和RTCP一起使用,后面的是控制协议.

他是建立在UDP协议之上的,基于udp的话那就是数据会丢失咯

服务端是以固定的速度在网上发送数据,客户端也是以一种固定的速率看视频,不能快进啥的,观看完了还得重新下载.

属于仅仅实现了视频流基本功能的东西.

## RTSP

Real-time stream protocol

rtsp就是在rtp那套上又加了一层功能, 可以使客户端和服务端有交互,进而也就是实现了快进和快退的功能.

而且可以支持udp,也可以支持tcp,(在rtp基础上增加了支持tcp的功能)

## RTMP

Real-time message protocol

视频直播协议 是基于tcp的,数据不会丢失

这个协议就是为了直播而出的, 所以那种多人直播,万人直播都是基于rtmp的.

RTMP 协议也要客户端和服务器通过"握手"来建立 RTMP Connection，然后在Connection上传输控制信息。RTMP 协议传输时会对数据格式化，而实际传输的时候为了更好地实现多路复用、分包和信息的公平性，发送端会把Message划分为带有 Message ID的Chunk，每个Chunk可能是一个单独的Message，也可能是Message的一部分，在接受端会根据Chunk中包含的data的长度，message id和message的长度把chunk还原成完整的Message，从而实现信息的收发.

而且rtmp的cdn做的比较好,一个主播向一个cdn服务器发流消息,cdn服务再带给其他的就可以支持万人直播了.

## HLS

http living stream 可以从名字上看出来是基于http的.

上面的那几个都是基于传输层的,而这个是基于http的,苹果公司开发的基于http的流媒体协议

所以也就是ios用的比较多,专门为苹果手机和平板这种移动端准备的.

HLS 点播，基本上就是常见的分段HTTP点播，不同在于，它的分段非常小。  
相对于常见的流媒体直播协议，例如RTMP协议、RTSP 协议、MMS 协议等，HLS 直播最大的不同在于，直播客户端获取到的，并不是一个完  
整的数据流。  
HLS 协议在服务器端将直播数据流存储为连续的、很短时长的媒体文件(MPEG-TS格式)，而客户端则不断的下载并播放这些小文件，  
因为服务器端总是会将最新的直播数据生成新的小文件，这样客户端只要不停的按顺序播放从服务器获取到的文件，就实现了直播。由此可见，基本上可以认为，HLS 是以>>点播的技术方式来实现直播<<。由于数据通过 HTTP 协议传输，所以完全不用考虑防火墙或者代理的问题,而且分段文件的时长很短，客户端可以很快的选择和切换码率，以适应不同带宽条件下的播放。不过HLS的这种技术特点，决定了它的延迟一般总是会高于普通的流媒体直播协议。