

的时间与客户收到该块的时间之间的水平距离；注意到网络时延随视频块不同而变化。在此例子中，如果客户准备当第一块在 t_1 时刻一到达就开始播放，那么第二块将不能在 $t_1 + \Delta$ 时刻及时到达进行播放。在这种情况下，视频播放或将停止运行（等待第二块的到达）或可能漏掉第二块，即这两种情况都将导致不希望的播放损伤。相反，如果客户将播放延迟到 t_3 时刻开始，这时第一块到第六块都已经到达，所有已经收到的块在它们的播放时间前都能够进行周期性的播放。

7.2.1 UDP 流

我们这里仅简要讨论 UDP 流，在适当时向读者更为深入地介绍这些系统背后隐含的协议。使用 UDP 流，服务器通过 UDP 以一种稳定的速率记录下视频块，用与客户的视频消耗速率相匹配的速率传输视频。例如，如果视频消耗率是 2Mbps，每个 UDP 分组承载 8000 比特视频，则服务器将每隔 $(8000 \text{ 比特}) / (2 \text{ Mbps}) = 4 \text{ ms}$ 向其套接字发送一个 UDP 分组。如我们在第 3 章所知，因为 UDP 未采用某种拥塞控制机制，所以服务器能够以视频的消耗速率将分组推进网络中，而无 TCP 的速率控制的限制。UDP 流通常使用很小的客户端缓存，空间维持小于 1 秒视频就足够了。

在将视频块传递给 UDP 之前，服务器将视频块封装在运输分组中，该运输分组是专门为传输音频和视频而设计的，使用了实时传输协议（Real-Time Transport Protocol, RTP）[RFC 3550] 或某种类似（可能是专用）的方案。我们将在 7.3 节再讨论 RTP，那时我们将在会话式语音和视频系统环境中讨论 RTP。

UDP 流的另一种不同的性质是，除了服务器到客户的视频流外，两者间还并行地维护一个单独的控制连接，通过该连接，客户可发送有关会话状态变化的命令（如暂停、重新开始、重定位等）。这种控制连接在许多方面类似于我们在第 2 章中学习的 FTP 控制连接。在本书配套 Web 网站上更为详细地解释了实时流协议（Real-Time Streaming Protocol, RTSP）[RFC 2326]，它是一种用于这样的控制连接的流行开放协议。

尽管 UDP 流已经在多个开源系统和专用产品中得到应用，但它有三个重大不足。首先，由于服务器和控制之间的可用带宽无法预测并且是变化的，恒定速率 UDP 流不能够提供连续的播放。例如考虑以下场景：视频消耗速率为 1Mbps，服务器到客户可用带宽通常超过 1Mbps，但每过几分钟就有几秒钟时间其可用带宽低于 1Mbps。在这种场景下，以 1Mbps 恒定速率经 RTP/UDP 传输视频的 UDP 流系统很可能将提供不好的用户体验，在可用带宽低于 1Mbps 之后产生停滞或漏帧。UDP 流的第二个缺点是它要求如 RTSP 服务器这样的媒体控制服务器，以对每个进行中的客户会话处理客户到服务器的交互请求和跟踪客户状态（例如在视频中的客户播放点，视频是否被暂停或播放等）。这增加了部署大规模的按需视频系统的总体成本和复杂性。第三个缺点是许多防火墙配置为阻塞 UDP 流量，防止这些防火墙后面的用户接收 UDP 视频。

7.2.2 HTTP 流

在 HTTP 流中，视频直接作为具有一个特定 URL 的普通文件存储在 HTTP 服务器上。当用户要看视频时，客户和服务器之间建立一个 TCP 连接，并且发送一个对该 URL 的 HTTP GET 请求。服务器则尽可能快地在 HTTP 响应报文中发送该视频文件，这就是说，以 TCP 拥塞控制和流控制允许的尽可能快的速率进行处理。在客户端上，字节收集在一个