

我们假设无论何时客户缓存为非空时,服务器以一种恒定速率 $x$ 发送比特。(这是一种显而易见的简化,因为TCP的发送速率由于拥塞控制而变化;在本章后面的习题中我们将考察更为真实的与时间相关的速率 $x(t)$ 。)假设在时刻 $t=0$ ,应用缓存为空,视频开始到达客户应用缓存。我们现在问,在什么时刻 $t=t_p$ 开始播放呢?并且在播放过程中,什么时刻 $t=t_f$ 客户应用缓存变满呢?

首先,我们来确定 $t_p$ ,此时 $Q$ 比特已经进入应用缓存并且开始播放。前面讲过比特以速率 $x$ 到达客户应用缓存,并且在开始播放之前没有比特从其缓存中删除。所以,建立 $Q$ 比特所需的时间(初始缓存时延)是 $t_p = Q/x$ 。

我们现在来决定 $t_f$ ,这是客户应用缓存变满的时刻。我们先观察,如果 $x < r$ (即如果服务器发送速率小于视频消耗速率),则客户缓存将决不会满!的确,时刻 $t_p$ 开始,缓存将以速率 $r$ 排空并且仅以速率 $x < r$ 填充。最终客户缓存将完全排空,此时当客户缓存等待另一个 $t_p$ 秒来建起 $Q$ 比特的视频时,视频将在屏幕上停滞。所以,当网络中可用速率小于视频速率时,播放将在连续播放期和停滞播放期之间进行变动。在课后习题中,将请你决定每个连续播放期和停滞期的长度,它们都作为 $Q$ 、 $r$ 和 $x$ 的函数。当 $x > r$ 时,现在我们来决定 $t_f$ 。在这种情况下,在时刻 $t_p$ 开始,缓存以 $x-r$ 的速率从 $Q$ 增加到 $B$ ,因为比特以速率 $r$ 消耗但以速率 $x$ 到达,如图7-3所示。有了这些提示,在课后习题中将请你决定 $t_f$ ,即客户缓存变满的时刻。注意到当网络中的可用速率大于视频速率时,在初始缓存时延后,用户将享受连续的播放直到视频结束。

#### 4. 视频的早期中止和重定位

HTTP流系统经常利用HTTP GET请求报文中的HTTP字节范围首部(HTTP byte-range header),该首部指示了客户当前要从所希望的视频中获取的字节范围。当用户要在视频中及时重定位(即跳跃)到未来点时,这特别有用。当用户重定位到一个新位置时,客户发送一个新HTTP请求,用字节范围首部指出服务器应当从文件的哪个字节起发送数据。当服务器接收到该新的HTTP请求时,它能够忘记任何较早的请求,而是由字节范围请求中指示的字节开始发送。

在我们讨论重定位主题的时候,我们简要地提及当某用户重定位到视频中的某个未来点或提前终止视频时,某些由服务器发送的已预取但尚未观看的数据将不会被观看,即导致了网络带宽和服务器资源的浪费。例如,假设在视频中的某时刻 $t_0$ 客户缓存充满 $B$ 比特,在此时用户重定位到视频中的某个瞬间 $t > t_0 + B/r$ ,然后从这点起观察视频直到结束。在这种情况下,缓存中的所有 $B$ 比特将未被观看,用于传输这 $B$ 比特的带宽和服务器资源完全被浪费掉了。在因特网中,有大量的带宽因提前终止而浪费,这些成本可能相当大,特别是对于无线链路[Lhm 2011]。由于这个原因,许多流系统仅使用了长度适当的客户应用缓存,或者将限制在HTTP请求中使用字节范围首部预取的视频数量[Rao 2011]。

重定位和提前终止可以与下列做法类比:烹调了很多肉,仅吃了一部分,并将其他都扔掉,因而浪费了食物。因此,下次你因为没有吃完所有晚餐而被父母批评浪费食物时,你能够迅速反驳:当他们在因特网上观看电影并进行带宽重定位时,他们浪费了带宽和服务器资源!但是,别人错了不等于你对了,食物和带宽都不应被浪费!

#### 7.2.3 适应性流和DASH

尽管如前一小节所述,HTTP流在实践中已经得到广泛部署(例如,YouTube自发展