

任意时刻开始观看一部电影，所以，如果有100个用户全部在晚8点左右开始观看某个新电影，很可能不会有两个用户在完全相同的时刻开始，所以他们无法共享一个数据流。使优化成为可能的修改是，通知所有用户电影只在整点和随后每隔（例如）5分钟开始。因此，如果一个用户想在8:02看一部电影，那么他必须等到8:05。

这样做的收益是，不管存在多少客户，对于一部2小时的电影，只需要24个数据流。如图7-17所示，第一个数据流开始于8:00。在8:05，当第一个数据流处于第9000帧时，第二个数据流开始。在8:10，当第一个数据流处于第18 000帧并且第二个数据流处于第9000帧时，第三个数据流开始，以此类推直到第24个数据流开始于9:55。在10:00，第一个数据流终止并且再一次从第0帧开始。这一方案称为近似视频点播（near video on demand），因为视频并不是完全随着点播而开始，而是在点播之后不久开始。

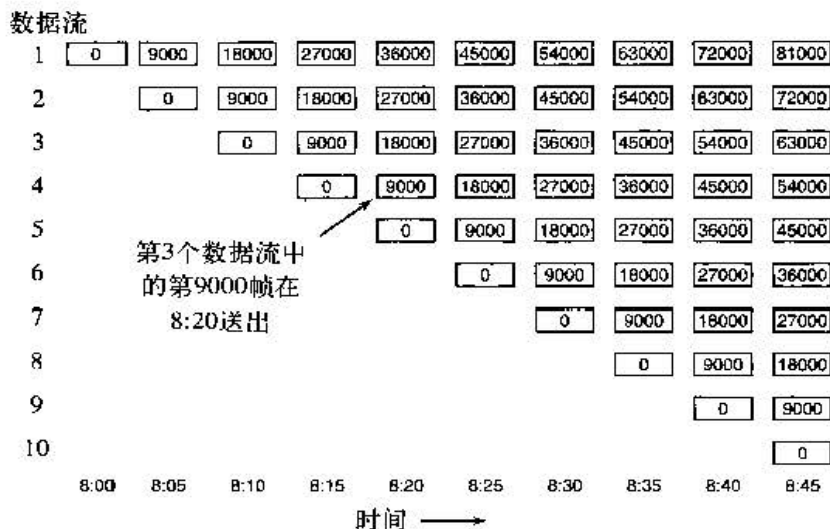


图7-17 近似视频点播以规则的间隔开始一个新的数据流，在本例中时间间隔为5分钟（9000帧）

这里的关键参数是多长时间开始一个数据流。如果每2分钟开始一个数据流，那么对于一部2小时的电影来说就需要60个数据流，但是开始观看的最大等待时间是2分钟。运营商必须判定人们愿意等待多长时间，因为人们愿意等待的时间越长，系统效率就越高，并且同时能够被观看的电影就越多。一个替代的策略是同时提供不用等待的选择权，在这种情况下，新的数据流可以立刻开始，但是需要对系统做更多的修改以支持即时启动。

在某种意义上，视频点播如同使用出租车：一招手它就来。近似视频点播如同使用公共汽车：它有着固定的时刻表，乘客必须等待下一辆。但是大众交通只有在存在大众的时候才有意义。在曼哈顿中心区，每5分钟一辆的公共汽车加起来至少还可以拉上一些乘客，而在怀俄明州乡间公路上旅行的公共汽车，可能在所有的时间几乎都是空空的。类似地，播放史蒂文·斯皮尔伯格的最新大片可能吸引足够多的客户，从而保证每5分钟开始一个新的数据流；但是对于《乱世佳人》这样的经典影片，最好还是简单地在点播的基础上播映。

对于近似视频点播，用户不具有VCR控制能力。没有用户能够暂停一部电影而去一趟厨房。他们所能做的最好的事情不过是当他们从厨房中返回时，向后退到随后开始的一个数据流，从而使漏过的几分钟资料重现。

实际上，近似视频点播还有另外一个模型。在这个模型中，人们可以在他们需要的任意时候预订电影，而不是预先宣布每隔5分钟将开演某部电影。每隔5分钟，系统要查看哪些电影已经被预订并且开始这些电影。采用这一方案时，根据点播的情况，一部电影可能在8:00、8:10、8:15和8:25开始，但不会在中间的时间开始。结果，没有观众的数据流就不会被传输，节约了磁盘带宽、内存和网络容量。另一方面，现在到厨房去制作冰淇淋就有点冒险，因为不能保证在观众正在观看的电影之后5分钟还有另一个数据流正在运行。当然，运营商可以给用户提供一个选项，以便显示所有同时发生的数据流的一个列表，但是大多数人觉得他们的电视机遥控器按钮已经太多，不大可能会热情地欢迎更多的几个按钮。