

组时延、时延抖动和丢包，而事实是，无论何时网络变得拥塞，都将出现较大的分组时延、时延抖动和丢包。改善多媒体应用质量的第一种方法就是“在问题上砸钱”，因此直接避免资源竞争即可。这种方法常用于解决有关资源受限的任何问题。在网络多媒体的场合，这意味着在整个网络中提供充足的链路带宽，使网络拥塞及其导致的分组时延和丢失决不会（或仅非常少地）出现。具有充足的链路带宽，分组将很快地通过今天的因特网而没有排队时延或丢失。从许多方面看，这是一种理想的情况：完美地执行多媒体应用，用户是幸运的，所有要求都能够满足而不改变因特网尽力而为的体系结构。

当然，问题是为实现这种极乐世界提供多大容量才是“充足的”，以及提供“充足的”带宽的成本从 ISP 的商业角度来说是否实际。在一个给定拓扑中为网络链路提供多大容量以取得给定水平的性能的问题常被称为带宽供给（bandwidth provisioning）。如何设计一个网络拓扑（其中放置一些路由器，如何用链路互联这些路由器，并为链路分配容量）以取得给定水平的端到端性能这个更为复杂的问题常被称为网络定制（network dimensioning）。带宽供给和网络定制都是复杂的专题，它们远超过了本教科书的范围。然而，我们这里注意到，为了预测两个网络端点之间的应用级性能，必须处理下列问题，并因此提供充足的容量来满足应用的性能要求。

- 网络端点之间的流量需求模型。这些模型可能需要定义在呼叫层次（例如，用户“到达”网络并启动端到端应用）和分组层次（例如，由进行中的应用所产生的分组）。注意负载可能随着时间而变化。
- 定义良好的性能要求。例如，为支持诸如会话式多媒体应用等时延敏感流量，其性能要求可能是：分组的端到端时延大于最大可容忍时延的概率要小于某个很小的值 [Fraleigh 2003]。
- 对给定的负载模型预测端到端性能的模型，以及求出最小成本带宽分配（该带宽分配将导致满足所有用户的需求）的技术。这里，研究人员正忙于研发能够量化给定负载下的性能的性能模型，以及能求出满足性能要求的最小成本带宽分配的优化技术。

假定今天尽力而为的因特网能够（从技术角度讲）以适当的性能水平支持多媒体流量（如果它被定制成这样的话），自然的问题是为什么今天的因特网满足不了这样的要求。答案基本上在经济上和组织上的原因。从经济角度看，用户将愿意向其 ISP 支付足够多的费用，使 ISP 安装充足的带宽经尽力而为的因特网来支持多媒体应用吗？组织问题也许更为令人气馁。注意到在两个多媒体端点之间的端到端路径将通过多个 ISP 的网络。从组织角度看，这些 ISP 将愿意合作（也许以收入共享方式）以确保端到端路径被适当地定制来支持多媒体应用吗？对这些经济和组织问题的展望，参见 [Davies 2005]。对于供给第一层主干网络以支持时延敏感流量的展望，参见 [Fraleigh 2003]。

### 9.5.2 提供多种类型的服务

也许对今天因特网中的以不变应万变的尽力而为服务而言，一种最简单的强化是将流量划分为多种类型，并为这些不同类型的流量提供不同等级的服务。例如，某 ISP 可能要为时延敏感的 VoIP 或电信会议流量比电子邮件或 HTTP 等弹性流量提供更高的服务类型（并对该服务收取更高的费用）。另一种做法是，ISP 可能直接向愿意对这种改进服务支付更多费用的顾客提供更高质量的服务。一些住宅有线接入 ISP 和蜂窝无线接入 ISP 已经采用了这样的梯次等级服务，即铂金卡服务用户比金卡服务用户或银卡服务用户享有更

好的服务性能。

我们都从日常生活中熟悉了不同类型的服务，如航班上头等舱乘客比公务舱乘客得到更好的服务，公务舱乘客又比经济舱乘客得到更好的服务；VIP 在活动中能够立即进入，而所有其他人都必须排队等待；在某些国家中老年人被尊重，提供了荣誉座位和最精细的食物。重要的是注意到在聚合流量中（即在多种流量类型中而不是单个连接中）提供了这种有差别的服务。例如，所有头等舱乘客被一视同仁（没有哪个头等舱乘客得到了比其他头等舱乘客更好的服务），就像所有的 VoIP 分组在网络中得到了相同的对待，与它们所属的特定的端到端连接无关。如我们将见到的那样，通过处理少量的流量聚合，而不是大量的单个连接，提供好于尽力而为服务所需要的新型网络机制能够保持相对简单。

早期因特网设计者的心中清晰地具有这种多种类型服务的概念。回想第4章中 IPv4 首部中的服务类型（ToS）字段。IEN123 [ISI 1979] 描述也呈现在 IPv4 数据报的原型中的 ToS 字段时说：“服务类型 [字段] 提供了所希望的服务质量的抽象参数的指示。当传输一个数据报通过某特定网络时，这些参数被用于引导实际服务参数的选择。几种网络提供了服务优先权，该优先权以某种方式把高优先权流量看得比其他流量更为重要。”在 40 多年前，向不同等级的流量提供不同等级的服务的设想就是清晰的！然而，我们又花费了等长的时间去实现这种愿景。

### 1. 促进思考的场景

下面用几种促进思考的场景来开始我们的提供多种类型服务的网络机制的讨论。

图 9-11 表示了一种简单的网络场景，两个应用分组流产生于位于一个局域网的主机 H1 和 H2，它们的目的地是另一个局域网的主机 H3 和 H4。在这两个局域网上的两台路由器通过一条 1.5Mbps 的链路连接起来。我们假设局域网的速度远远高于 1.5Mbps，并且关注路由器 R1 的输出队列；注意到如果 H1 和 H2 的总计发送速率超过了 1.5Mbps，分组时延和丢包将会出现。

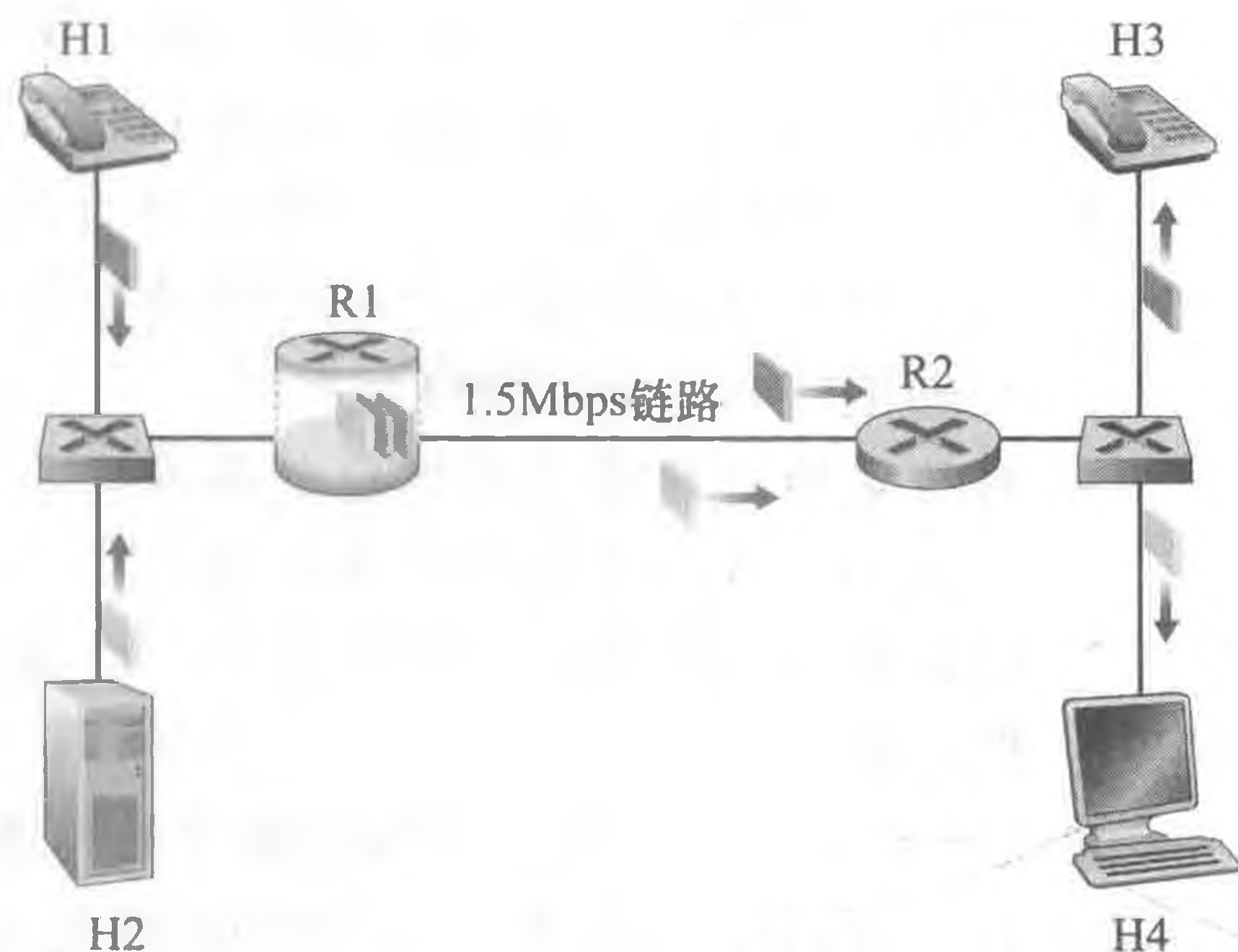


图 9-11 音频应用与 FTP 应用的竞争

我们进一步假设 1Mbps 的音频应用（例如一个 CD 质量的音频呼叫）共享 R1 和 R2 之间 1.5Mbps 的链路，同时从 H2 到 H4 有一个 HTTP Web 浏览应用正在下载一个 Web 网页。

在尽力而为服务的因特网中，该音频和 HTTP 分组在 R1 的输出队列中混合，并且（通常）以先进先出（FIFO）的次序传输。在这种情况下，来自 Web 服务器的突发分组可能潜在地填满这个队列，引起 IP 音频分组过度延迟或者由于 R1 的缓存溢出而丢失。我们应该如何解决这个潜在的问题呢？假定该 HTTP Web 浏览应用没有时间限制，我们的直觉也许是在 R1 为音频分组分配严格的优先级。在一个严格的优先级调度规则下，在 R1 输出缓存的音频分组总是在 R1 输出缓存中的任何 HTTP 分组之前传输。对音频流量而言，从 R1 到 R2 的链路看起来像一条 1.5Mbps 专用链路，而对于 HTTP 流量仅当没有音频流量排队时，才使用 R1 到 R2 的链路。为了让 R1 在它的队列中区分音频和 FTP 分组，每个分组必须被标记为属于这两类流量中的哪一类。这是 IPv4 中服务类型（ToS）字段的最初目的。显而易见，这则是我们对需要提供多种类型流量机制的第 1 个见解。



见解 1：标记分组（packet marking）使得路由器区分属于不同类型流量的分组。

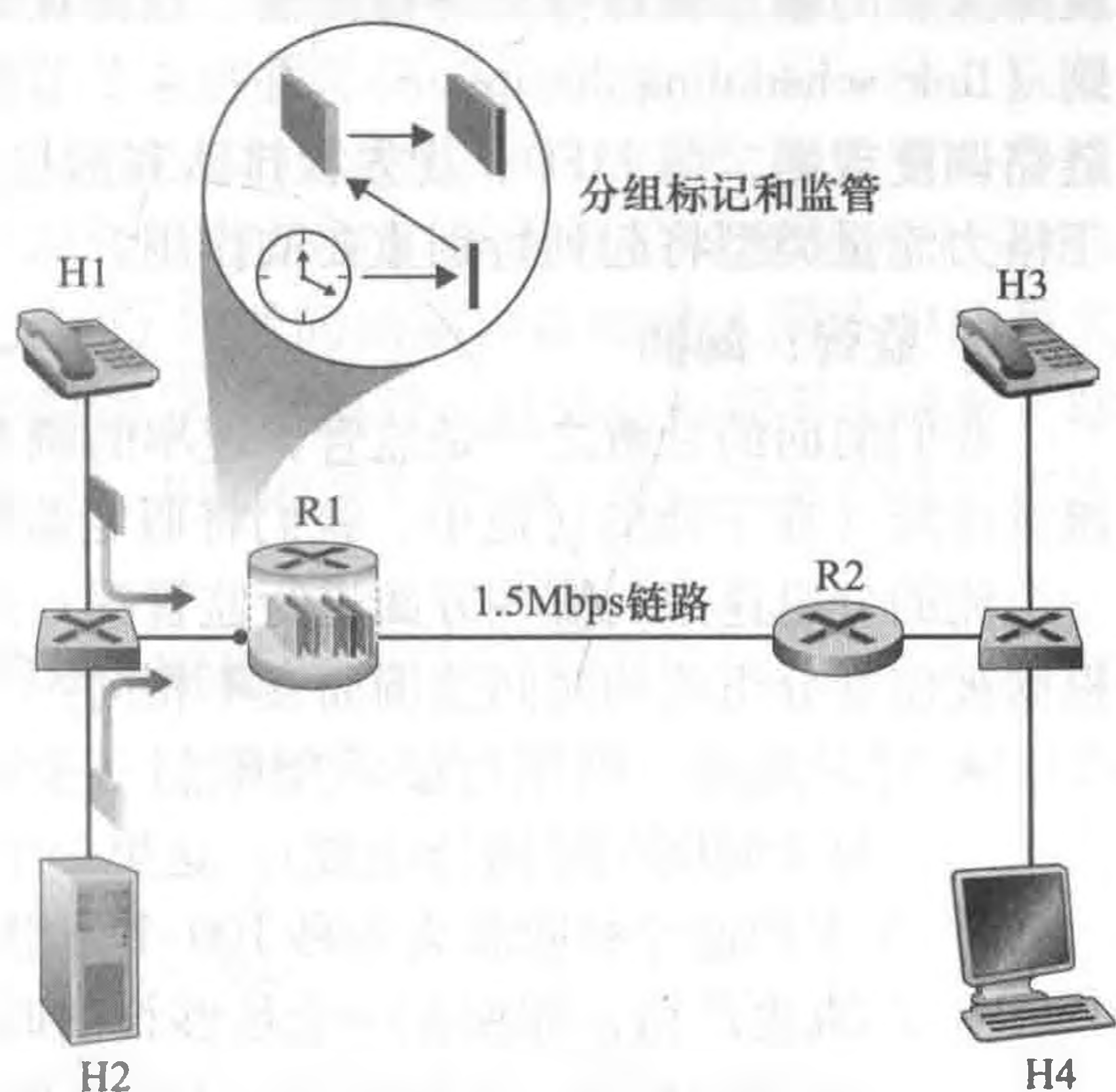
注意到尽管我们的例子考虑到竞争性的多媒体和弹性流，但是相同的见解应用于实现了铂金卡、金卡和银卡服务类型的情况，即仍需要标记分组机制以指示分组属于哪个类型的服务。

现在假设路由器被配置为给标记为属于 1Mbps 音频应用的分组赋予高优先级。因为输出链路速度是 1.5Mbps，即使 HTTP 分组得到较低的优先级，它们仍然可以收到平均 0.5Mbps 的传输服务。但是如果音频应用开始以 1.5Mbps 或者更高的速率（或者恶意的，或者由于应用的差错）发送分组，那会出现什么样的情况呢？在这种情况下，HTTP 分组将挨饿，也就是在 R1 到 R2 的链路上得不到任何服务。如果多个应用（例如，多个音频呼叫）都具有同等的服务类型，共享一段链路带宽，那么也会出现类似问题，即它们也可能共同饿死该 HTTP 会话。理想情况下，一种服务要与各类流量有隔离度，以保护一种流量类型免受其他流量类型干扰。这种保护能够在网络中的不同地方实现，在每台路由器中，在进入网络的首个入口，或在网络边界域间。这则是我们的第 2 个见解。

见解 2：希望在流量类型之间提供流量隔离（traffic isolation）的度，以便一类流量不会受到另一类异常流量的负面影响。

我们将考察在流量类型之间提供这种隔离的特定机制。这里我们注意到，有两大类方法可以使用。首先，可以执行如图 9-12 所示的流量监管（traffic policing）方法。如果流量类型或流必须满足一定的准则（例如，音频流不超过 1Mbps 的峰值速率），那么可以设置一个监管机制来确保这些准则的确被遵守。如果被监管的应用行为异常，这个监管机制将采取某种行动（例如，丢弃或者延时那些违反这些准则的分组），以便实际进入网络的流量符合这些准则。我们很快将考察的漏桶机制也许是使用最广泛的监管机制。在图 9-12 中，分组分类和标记机制（见解 1）以及监管机制（见解 2）都一起在网络的边缘实现，或在端系统中实现，或在边界路由器中实现。

为流量类型之间提供隔离的一种互补的方法是，链路级的分组调度机制为每种类型明确地分配固定量的链路带宽。例如，在 R1 能够给音频类型分配 1Mbps，能够给 HTTP 流分配 0.5Mbps。在这种情况下，音频和 HTTP 流分别看到了容量为 1.0Mbps 和 0.5Mbps 的逻辑链路，如图 9-13 所示。通过严格执行链路级的带宽分配，一种类型



图例：



测定和监管

|| 标记

图 9-12 监管（与标记）音频与 HTTP 流量类型

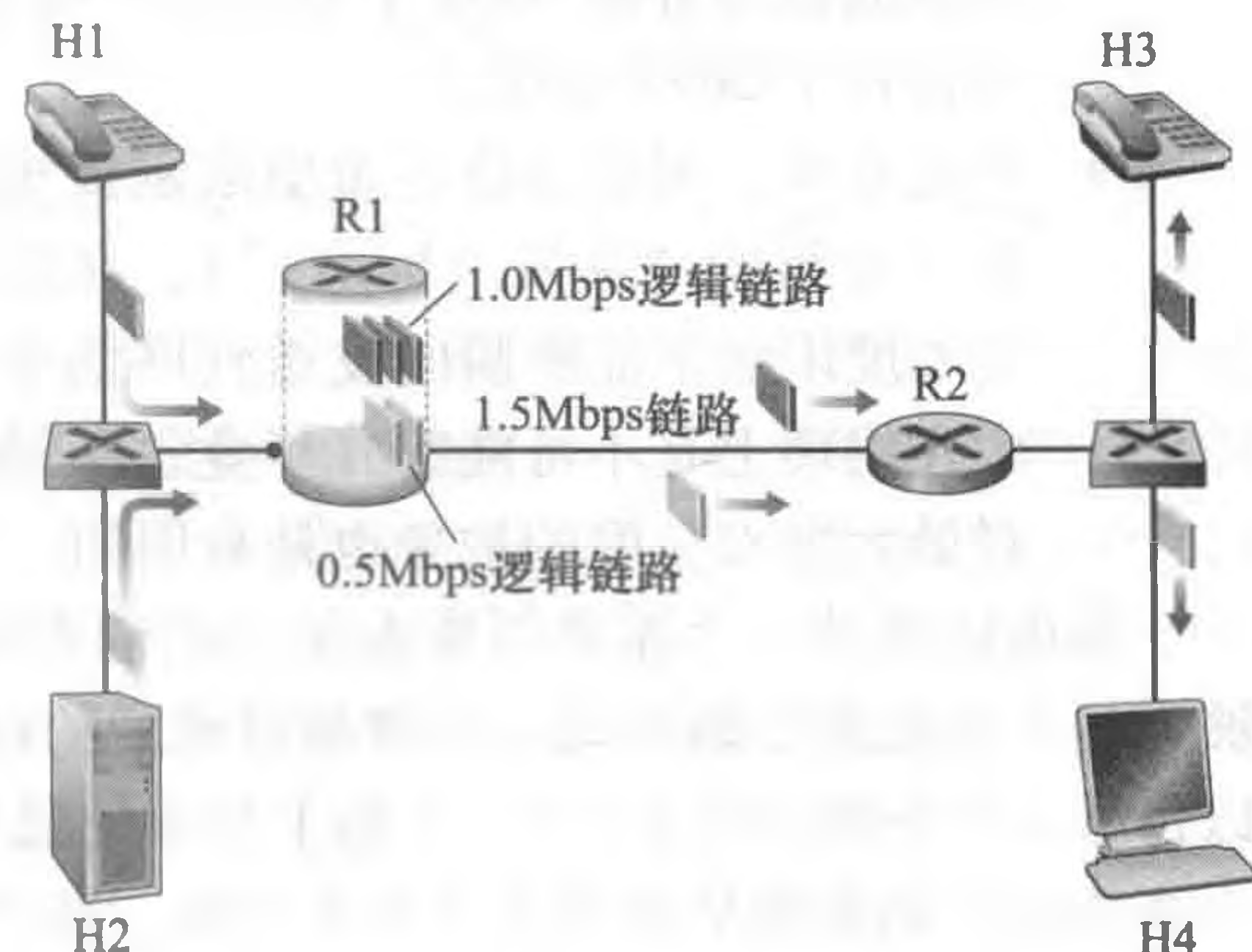


图 9-13 音频与 HTTP 流量类型的逻辑隔离

仅能够使用已经分配的带宽量；特别是，它不能利用其他应用现在未使用的带宽。例如，如果音频流静默了（例如，如果谈话者停顿，不产生音频分组），HTTP 流在 R1 到 R2 的链路上的传输带宽仍然不能够超过 0.5Mbps，即使音频流分配的 1Mbps 带宽在那个瞬间没有使用。由于带宽是一种“使用它或丢失它（use-it-or-lose-it）”的资源，没有理由妨碍 HTTP 流量使用没有由音频流使用的带宽。我们将希望尽可能高效地使用带宽，当能够以别的方法使用它时决不浪费带宽。这引发我们的第 3 个见解。

**见解 3：**当为流量类型或流之间提供隔离时，希望尽可能有效地使用资源（例如链路带宽和缓存）。

## 2. 调度机制

在 1.3 节和 4.2 节我们曾经讨论过，属于各种网络流的分组被复用在一起，并且在与链路关联的输出缓存排队等待传输。选择在链路上传输的排队分组的方式称为链路调度规则（link-scheduling discipline），并在 4.2 节进行了详细讨论。回想在 4.2 节中讨论过三种链路调度规则，即 FIFO、优先权排队和加权公平队列（WFQ）。我们马上将看到 WFQ 对于格力流量类型将起到特别重要的作用。

## 3. 监管：漏桶

我们前面的见解之一是监管，速率的调节是一种重要的 QoS 机制，以该速率允许一类流量或流（在下面的讨论中，我们将假定监管的单位是流）向网络注入分组。但是应该对一个流的分组速率的哪些方面进行监管呢？我们能够指出 3 个重要的监管准则，每个准则根据被监管分组流的时间范围而互不相同：

- **平均速率。**网络可能希望限制一个流的分组能够发送到网络中的长期平均速率（每个时间间隔的分组数）。这里一个关键的问题是监管平均速率的时间间隔。一个平均速率被限制为每秒 100 个分组的流要比一个每分钟 6000 个分组的源受到的约束更严格，即使在一个足够长的时间间隔上它们有相同的平均速率。例如，后者的限制允许一个流在给定 1 秒长的时间间隔内发送 1000 个分组，而前者的限制不允许这种发送行为。
- **峰值速率。**平均速率约束限制了一个相对长的时间内能够发送到网络中的流量，而峰值约束限制了一个较短时间内能够发送的最大分组数。使用我们上面的例子，网络能以每分钟 6000 个分组的平均速率来监管一个流，但是限制该流的峰值速率为每秒 1500 个分组。
- **突发长度。**网络也许还希望限制极短的时间间隔内能够发送到网络中的最大分组数（分组的“突发（burst）”）。在这个限制中，因为时间间隔长度趋近于 0，该突发长度限制了能够瞬间发送到网络中的分组数量。尽管瞬间发送多个分组到网络中在物理上是不可能的（毕竟每条链路都有一个无法超越的物理传输速率！），但对最大突发长度的抽象也是有用的。

漏桶机制是一个能够用来表征这些监管限制的抽象。如图 9-14 所示，漏桶由一个能够容纳  $b$  个令牌的桶组成。令牌加进该桶的过程如下。可能潜在地加入桶中的新令牌总是以每秒  $r$  个令牌的速率产生。（为了简单化起见，我们这里假设时间单元是秒。）当产生一个令牌时，如果桶填充得少于  $b$  个令牌，新产生的令牌加入到该桶中；否则忽略该新产生的令牌，令牌桶保持具有  $b$  个令牌的满状态。



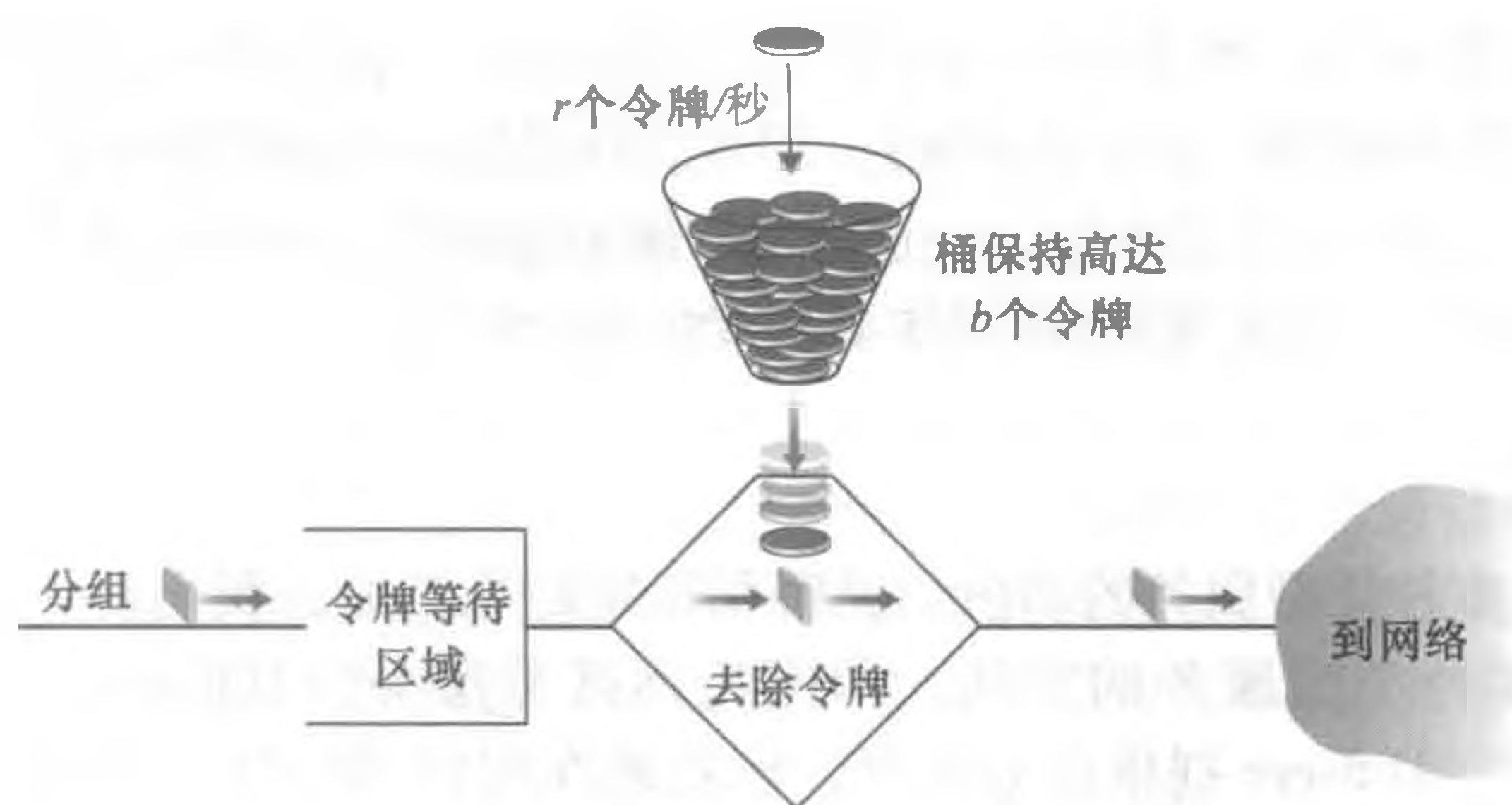
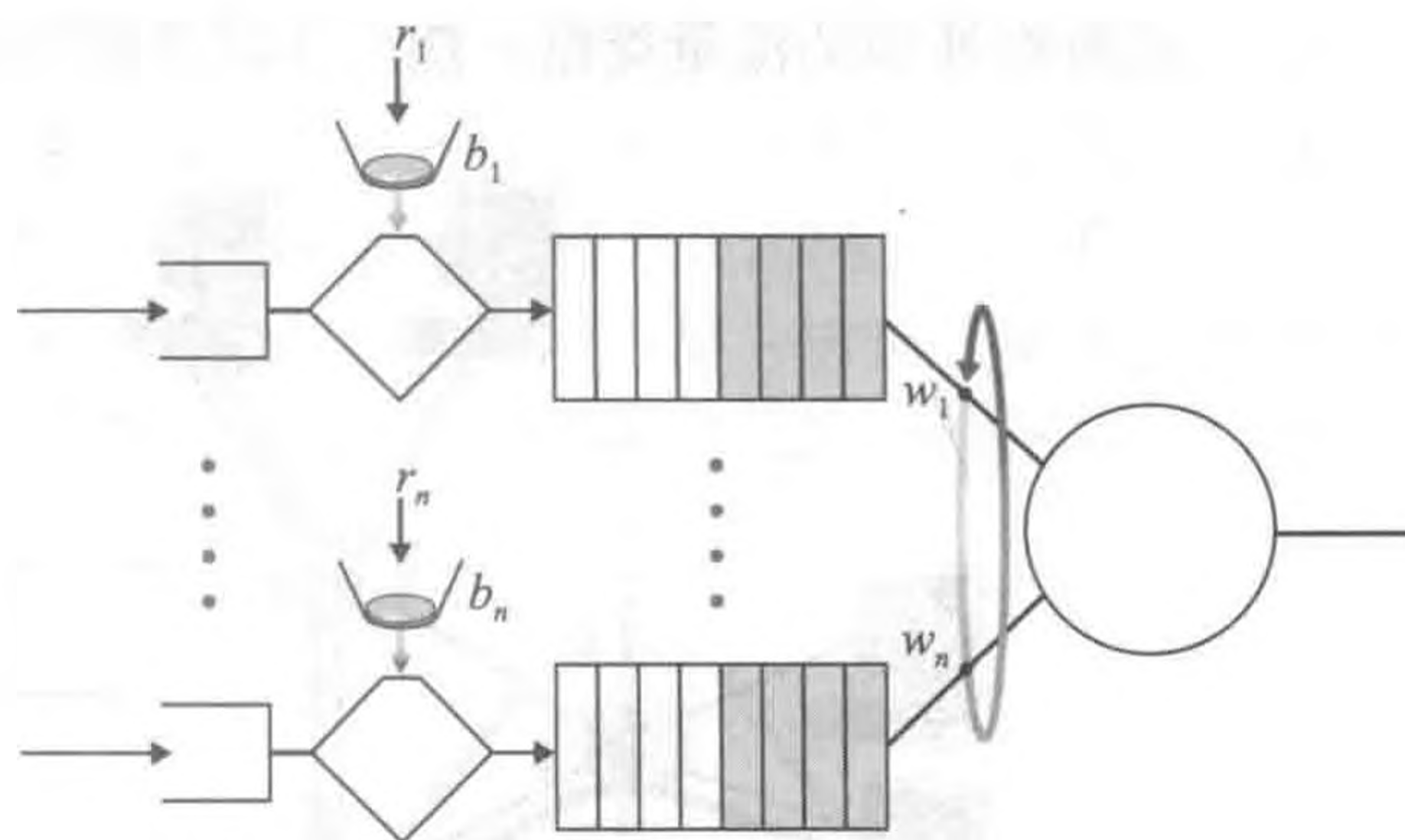


图 9-14 漏桶监管器

现在我们考虑如何用漏桶来监管分组流。假设在一个分组向网络传输之前，必须首先从令牌桶中去除一个令牌。如果令牌桶是空的，分组必须等待一个令牌。（另一种方法是丢弃该分组，尽管我们这里不讨论这种选择。）现在我们考虑这种行为是如何监管一个流量流的。因为在桶中最多能有  $b$  个令牌，所以漏桶监管的流的最大突发长度是  $b$  个分组。此外，因为令牌产生速率是  $r$ ，所以在任何长度为  $t$  的时间间隔内能够进入网络中的最大分组数目为  $rt + b$ 。因此，令牌产生速率  $r$  用于限制分组能够进入网络的长期平均速率。除了监管长期平均速率之外，使用漏桶（具体来说，串联的两个漏桶）来监管流的峰值速率也是有可能的；见本章末的习题。

#### 4. 漏桶 + 加权公平排队 = 队列中可证明的最大时延

我们关于策略的最后讨论是说明如何将调度和监管两种机制结合起来，以提供通过路由器队列的时延界限（鼓励已经忘记 WFQ 的读者复习一下 4.2 节）。我们考虑一台使用了 WFQ 调度的路由器，它的输出链路多路复用了  $n$  条流，其中每条流被一个参数为  $b_i$  和  $r_i$  的漏桶监管， $i = 1, \dots, n$ 。这里我们使用的术语“流”大致是指不能被调度器相互区别的分组集合。实践中，一条流可能是由单个端到端连接上的流量或者许多这种连接的集合流量组成的，参见图 9-15。

图 9-15 采用 WFQ 调度机制的  $n$  路复用的漏桶流

我们前面讨论 WFQ 时讲过，每个流  $i$  保证收到至少等于  $R \cdot w_i / (\sum w_j)$  的共享链路带宽，其中  $R$  是以分组/秒为单位的链路传输速率。当以 WFQ 方式等待服务时（也就是通过漏桶传递之后），则分组经受的最大时延是什么？我们来关注流 1。假设流 1 的令牌桶最初是满的。然后  $b_1$  个分组的突发到达流 1 的漏桶监管器。这些分组去除了漏桶中所有的令牌（没有等待），然后加入了流 1 的 WFQ 等待区域。因为这  $b_1$  个分组以至少  $R \cdot w_1 / (\sum w_j)$  分组/秒的速度得到服务，直到这些分组的最后一个传输完成，所以将有最大时延  $d_{\max}$ ，其中

$$d_{\max} = \frac{b_1}{R \cdot \frac{w_1}{\sum w_j}}$$

该公式的基本原理是，如果在队列中有  $b_1$  个分组并且分组以至少每秒  $R \cdot w_1 / (\sum w_j)$  个分组的速率从队列中接受服务（被去除），那么直到最后一个分组的最后一个比特被传送，时间量不会超过  $b_1 / (R \cdot w_1 / (\sum w_j))$ 。一道课后习题请你证明，只要  $r_1 < R \cdot w_1 / (\sum w_j)$ ，则  $d_{\max}$  确实是流 1 中任何分组在 WFQ 队列中要经受的最大时延。

### 9.5.3 区分服务

在学习了提供多种类型服务的动机、见解和特定的机制后，我们将注意力集中于以举例方式学习提供多种类型服务的方法，即因特网区分服务（Diffserv）体系结构 [RFC 2475; Kilkki 1999]。Diffserv 提供服务区分，也就是在因特网中以一种可扩展性方式用不同的方法处理不同类型流量的能力。可扩展性的需求来源于这样的事实：在因特网的一个主干路由器上同时存在几十万个源到目的地的并行流。我们很快将看到，仅仅通过在网络核心放置简单的功能，而在网络的“边缘”实现更复杂的控制操作，就可以满足该需要。

我们从图 9-16 中显示的简单网络开始讨论。这里我们将描述 Diffserv 的一种可能用法。如 RFC 2475 中所描述，其他的变化是可能的。区分服务体系结构由两个功能元素的集合所组成：

- **边界功能：**分组分类和流量调节。在网络的人边缘（也就是，或者在产生流量的 Diffserv 使能的主机，或者在流量经过的第一个 Diffserv 使能的路由器），到达的分组被标记。更具体地说，在 IPv4 或者 IPv6 分组首部中的区分服务（DS）字段被设置为某个值 [RFC 3260]。该 DS 字段的定义意欲取代我们在第 4 章中讨论的 IPv4 服务类型字段和 IPv6 流量类型字段。例如，在图 9-16 中，从 H1 发送到 H3 的分组可能在 R1 被标记，而从 H2 发送到 H4 的分组可能在 R2 被标记。分组得到的标记标识了该分组所属的流量类型。然后不同类型的流量将在核心网络得到不同的服务。

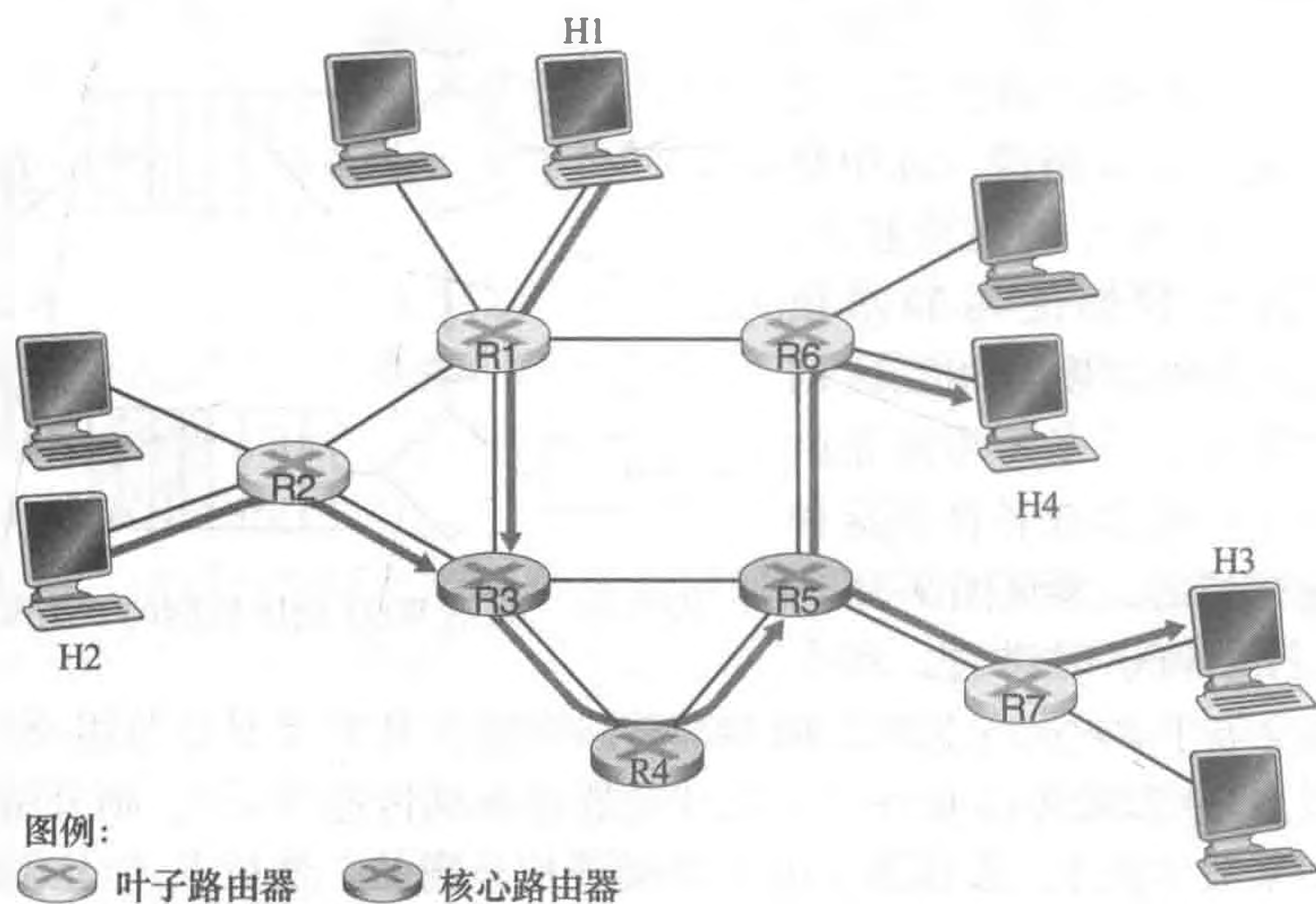


图 9-16 一个简单的 Diffserv 网络例子

- **核心功能：**转发。当一个 DS 标记的分组到达一个 Diffserv 使能的路由器时，根据与分组类型相关的所谓的每跳行为（Per-Hop Behavior, PHB），该分组被转发到它的下一跳。每跳行为影响在竞争的流量类型之间共享路由器缓存和链路带宽的方式。Diffserv 体系结构的一个关键原则是路由器的每跳行为只基于分组标记，即分



组所属的流量类型。因此，如果图 9-16 中从 H1 发送到 H3 的分组和从 H2 发送到 H4 的分组收到同样的标记，网络路由器则将这些分组处理为一个聚合体，而不区别这些分组是源于 H1 还是源于 H2。例如，当这些分组向 R4 转发时，R3 不区分从 H1 和 H2 产生的分组。因此，Diffserv 体系结构消除了为各个“源到目的地”对而保留路由器状态的要求，这是使 Diffserv 具有可扩展性的一个重要考虑。

这里有一个类比可能有助于理解相关概念。在很多大规模的社交活动中（例如，一场大型的公众招待会，一场大型的跳舞俱乐部活动或者迪斯科舞会，一场音乐会或者一场足球赛），参加活动的人收到某种类型的入场券。对于非常重要的人物有 VIP 入场券；对于 21 岁或者年龄更大的人有 21 岁以上的人场券（例如是否可以享受酒类饮料）；对于演唱人员有后台入场券；对于记者有新闻入场券；对于普通人有普通的人场券。这些入场券通常在该活动的入口分发，也就是在活动的边缘进行分发。正是在边缘，进行着计算密集型操作，例如交入场费、检查适合的邀请类型以及对照检查邀请与证件。此外，对于允许进入一个活动的特定类型的人数可能有限制。如果有这种限制，人们可能在进入活动之前必须等待。一旦进入了活动，一个人的入场券使他在活动中的很多场所接受有区别的服务，如为 VIP 提供免费的饮料、较好的桌子、免费食物、单独的房间和殷勤的服务。与之相对照的是，一个普通人禁止进入某些区域，要为饮料支付费用，并且只受到基本服务。在这两种情况下，活动中得到的服务只依赖于入场券的类型。而且在同一类型的所有人得到相同的对待。

图 9-17 提供了在边缘路由器中分类和标记功能的逻辑视图。到达边缘路由器的分组首先被分类。分类器根据一个或多个分组首部字段的值（例如源地址、目的地址、源端口、目的端口和协议 ID）来选择分组，并引导该分组去做合适的标记功能。如前面所指出的，分组的标记携带在分组首部的 DS 字段中。

在某些情况下，端用户可能已经同意限制其分组发送速率以符合某个申报的流量配置文件（traffic profile）。该流量配置文件可能包含对峰值速率和分组流的突发度的限制，如我们前面在漏桶机制中所见。只要用户以符合协商的流量配置文件的方式向网络中发送分组，这些分组就会得到它们的优先级标记，并沿着到目的地的路径转发。另一方面，如果违反了该流量配置文件，那些超出流量配置文件的分组就可能被打上不同的标记，或被整形（例如，为了能够遵守最大速率限制而延时），或可能在网络边缘被丢弃。图 9-17 中所示的测定功能（metering function）的作用是比较进入的分组流和协商的流量配置文件，并确定某分组是否在协商的流量配置文件之内。有关是否立即重新标记、转发、延时或者丢弃一个分组的实际决定是由网络管理员决定的策略问题，而不是由 Diffserv 体系结构规定的。

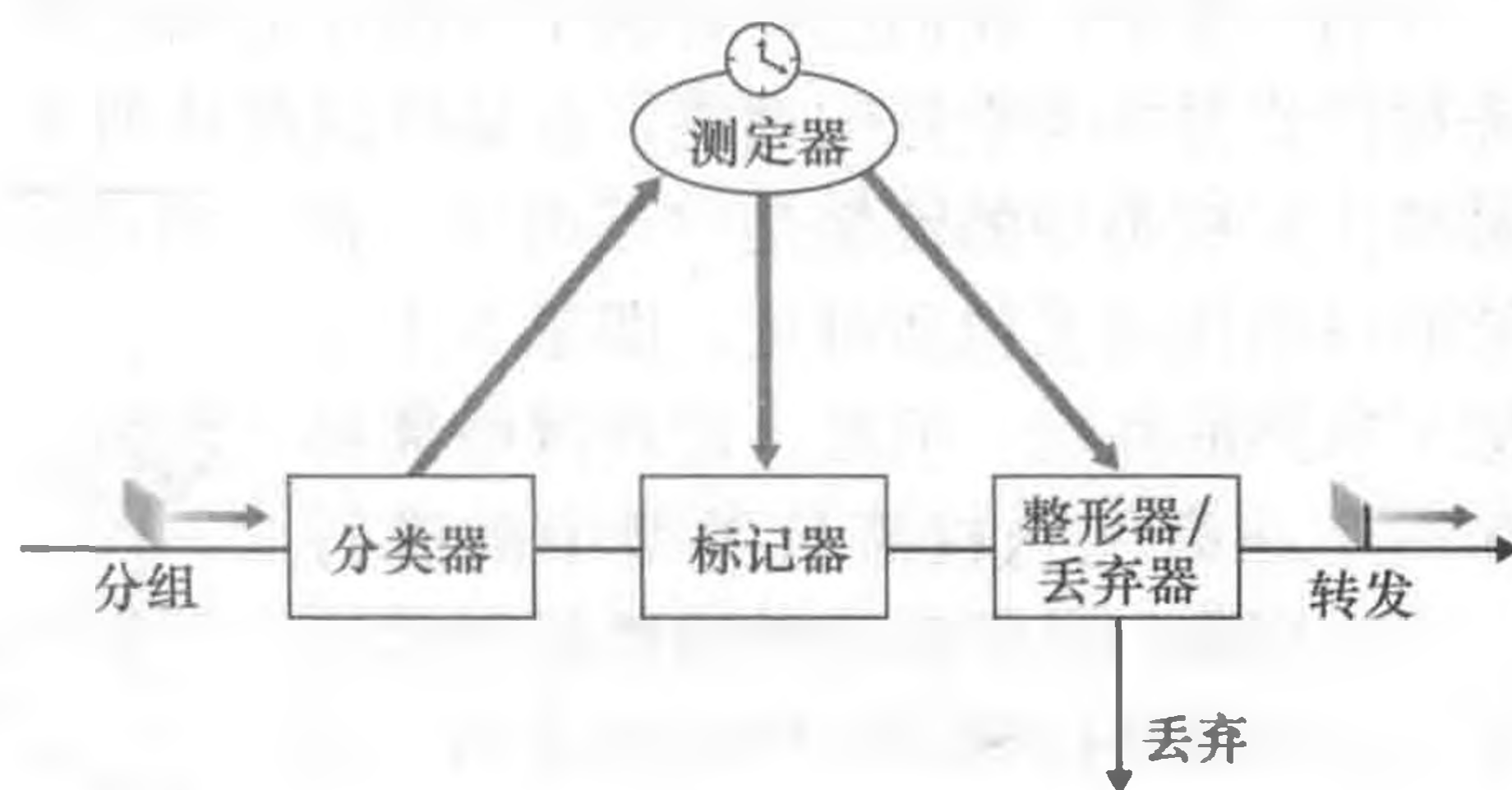


图 9-17 在端路由器上的分组分类与流量调节的逻辑视图

到现在为止，我们一直关注着 Diffserv 体系结构中的标记和监管功能。Diffserv 体系结构的第二个关键组件涉及由 Diffserv 使能路由器所执行的每跳行为（PHB）。该每跳行为有点神秘，但被仔细地定义为“Diffserv 节点的外部可观察的转发行为的描述，该节点应用了一个特定的 Diffserv 行为聚合”[RFC 2475]。稍微深入地钻研一下这个定义，我们可以

看到它里面包含的几个重要的考虑：

- PHB 能够导致不同服务类型的流量接收到不同性能（即不同的外部可观察的转发行为）。
- 虽然 PHB 定义了在各种类型之间的性能（行为）差别，但它不强求为了获得这些行为的任何特定机制。只要外部可观察的性能准则得到满足，任何实现机制和任何缓存/带宽分配策略都可以使用。例如，一个 PHB 将不要求使用一个特定的分组排队规则（例如优先级队列、WFQ 队列或 FCFS 队列）以取得特定的行为。PHB 是最后的结果，资源分配和实现机制是达到 PHB 的手段。
- 性能的差别必须是可观察的，因而是可测量的。

已经定义了两种 PHB：一种是加速转发 PHB [RFC 3246]，另一种是确保转发 PHB [RFC 2597]。加速转发（Expedited Forwarding, EF）PHB 规定了一类流量离开路由器的速率必须等于或者大于某个已配置的速率。确保转发（Assured Forwarding, AF）PHB 将流量分为 4 类，其中每个 AF 类都确保提供某种最小数量的带宽和缓存。

我们通过对 Diffserv 的服务模型进行一些观察来结束对 Diffserv 的讨论。首先，我们隐含地假设 Diffserv 部署在单个管理域中，但典型情况是端到端的服务必须在跨越通信端系统之间的多个 ISP 来形成。为了提供端到端的 Diffserv 服务，端系统之间的所有 ISP 不仅必须提供这种服务，而且为了向端用户提供真正的端到端服务，多数 ISP 之间也要协作并做出安排。如果没有这种协作，直接向客户出售 Diffserv 的 ISP 会发现他们将重复地说：“是的，我们知道你支付了额外费用，但是我们与丢弃和延迟你流量的 ISP 之间没有服务约定。我很抱歉在你的 VoIP 电话中有很多间隙！”第二，如果 Diffserv 实际存在并且该网络运行的负载不大，大多数时间尽力而为服务和 Diffserv 服务之间将没有可觉察的差异。的确，端到端时延通常是由接入速率和路由器跳数造成的，而不是由路由器中的排队时延造成的。想象一个支付了增值服务费用的不幸的 Diffserv 客户，发现为其他人提供的尽力而为服务几乎总是具有与增值服务相同的性能！

#### 9.5.4 每连接服务质量保证：资源预约和呼叫准入

在前一节中，我们已经看到了分组标记和监管、流量隔离以及链路级调度能够为一类服务提供比另一类更好的性能。在某些调度规则下，如优先权调度，较低类型的流量基本对最高优先权类型的流量是“不可见”的。借助于适当的网络定制，最高类型的服务的确能够取得极低的丢包和时延，即基本上是类似于电路的性能。但是，这种网络能够确保一个在高优先权流量类型中的进行流，仅使用我们已经描述的这些机制就能在整个流期间持续得到这样的服务吗？答案是否定的。在本节中，我们将看到当对各个连接提供硬服务保证时，为什么还需要另外的网络机制和协议。

我们返回 9.5.2 节的场景中，考虑两个 1Mbps 的音频应用经 1.5Mbps 的链路传输它们的分组，如图 9-18 所示。这两

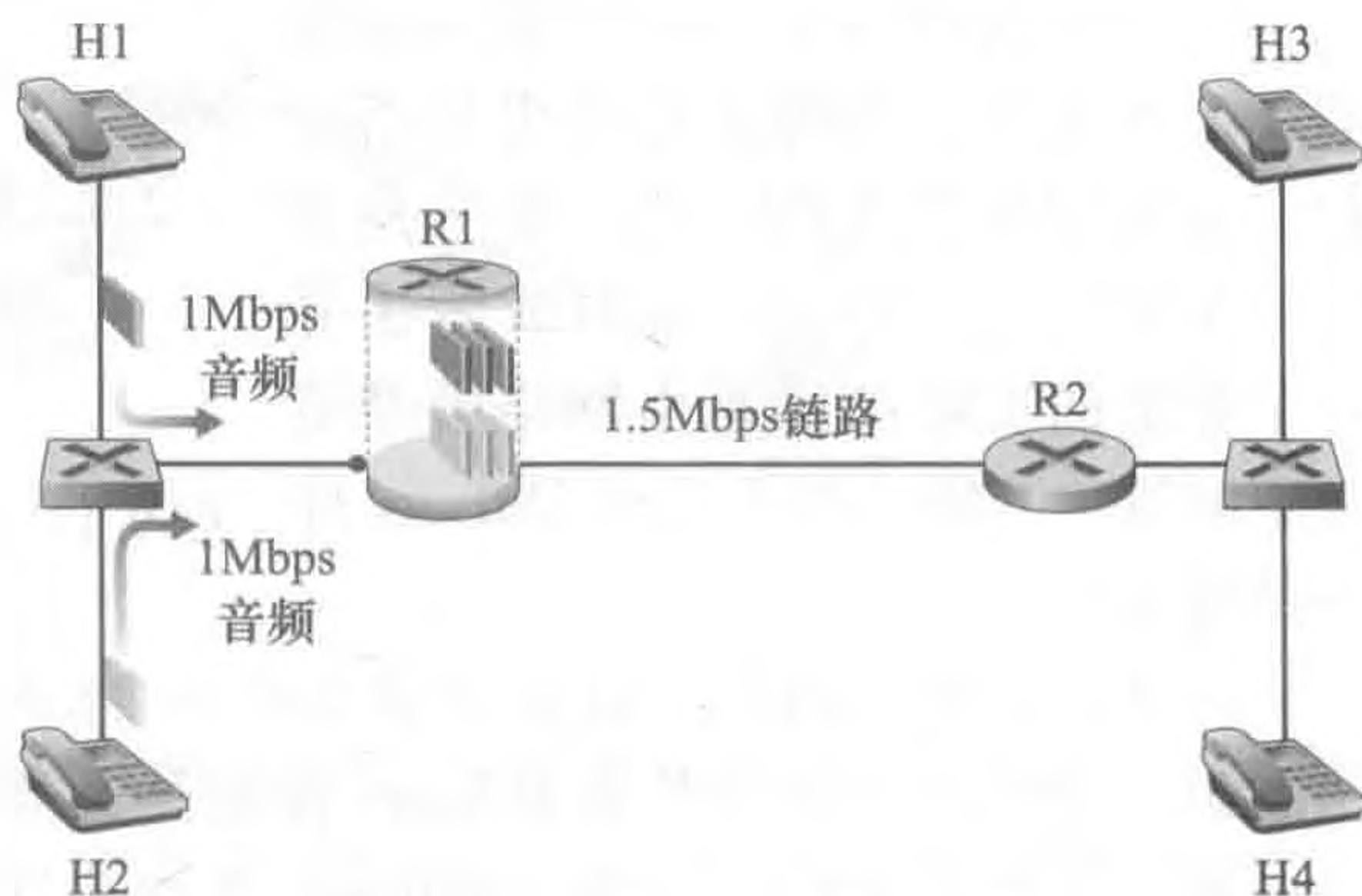


图 9-18 两个竞争的音频应用过载 R1 到 R2 的链路



条流的结合数据率 (2Mbps) 超过了这条链路的容量。即使使用分类和标记、流量隔离以及共享未使用的带宽 (这里并没有未使用的带宽), 这很明显还是一个与丢包有关的命题。只是因为没有足够的带宽同时满足两个应用的需求。如果这两个应用平等地共享该带宽, 每个应用将会丢失其 25% 的传输分组。这是一个无法接受的低 QoS, 以至于两个音频应用完全不可用; 甚至从一开始就没有必要传输任何音频分组。

在图 9-18 中两个应用不能同时满足的情况下, 这个网络应当做些什么呢? 如果允许两者以一种不可用的 QoS 继续在应用流上浪费网络资源, 最终将无法向端用户提供任何效用。希望该答案是清晰的, 即应用流中的一个应当被阻塞 (即拒绝接入网络), 而另一个应当允许继续进行, 使该应用使用所需的全部 1Mbps。电话网是一个执行这种呼叫阻塞的网络例子, 即如果要求的资源 (在电话网的情况下是一个端到端的电路) 不能分配给该呼叫, 该呼叫就被阻塞了 (阻止进入网络), 并且返回给用户一个忙信号。在我们的例子中, 如果一个流没有分配到足够的 QoS 来使自己可用, 允许它进入网络没有任何好处。的确, 接纳一个不能得到它需要的 QoS 的流是要付出代价的, 因为网络资源被用于支持一个对端用户没有用的流。

通过基于流的资源要求以及已经准入流的资源要求明确地准入或阻塞流, 网络能够保证准入流可得到它们所请求的 QoS。这隐含的意思是: 需要为流提供有保证的 QoS 就需要流来申告它的 QoS 需求。让流申告它的 QoS 需求, 然后让网络接受该流 (以所要求的 QoS) 或者阻塞该流的过程称为呼叫准入 (call admission) 过程。这则是我们的第 4 个见解 (除了前面 9.5.2 节中的 3 个见解外), 该见解增加了提供 QoS 所需的机制。

见解 4: 如果不总是能够提供充分的资源, 并且要确保 QoS, 则需要一个呼叫准入过程, 在这个过程中流申告其 QoS 需求, 然后要么被网络准入 (以所要求的 QoS), 要么被网络阻塞 (如果网络不能提供所要求的 QoS)。

如果一个呼叫 (一个端到端流) 一旦开始就要被确保给定的服务质量, 则我们在图 9-18 中的促进思考的例子强调了对几种新的网络机制和协议的需求:

- 资源预留。为了满足一个呼叫所希望的 QoS, 确保它具有所需的资源 (链路带宽、缓存) 的唯一方法是显式地为该呼叫分配这些资源, 用网络的行话来讲, 该过程称为资源预留 (resource reservation)。一旦呼叫预约了资源, 它在其整个过程中按需访问这些资源, 而不管所有其他呼叫的需求。如果一个呼叫预约并得到链路带宽的  $x$ Mbps 的保证, 而且传输速率决不会大于  $x$  Mbps, 则该呼叫将具有无丢包和无时延的性能。
- 呼叫准入。如果预留了资源, 则该网络必须具有一种用于呼叫请求和预留资源的机制。由于资源不是无限的, 如果请求的资源不可用, 则进行呼叫准入请求的呼叫将被拒绝准入, 即被阻塞。电话网执行的是这种呼叫准入机制, 即当我们拨一个号码时, 就请求了资源。如果完成该呼叫所需的电路 (TDMA 时隙) 是可用的, 则分配电路并且完成了呼叫。如果电路不可用, 则该呼叫被阻塞, 我们得到了忙信号。为了得到网络的准入, 被阻塞的呼叫能够一再尝试, 但是直到它成功地完成呼叫准入过程, 才被允许向网络发送流量。当然, 分配链路带宽的路由器, 其分配的带宽不应当超过链路可用的带宽。通常, 一个呼叫仅能预约一条链路的带宽的一部分, 因此一台路由器可以为多于一条呼叫分配链路带宽。然而, 如果要提供硬服务质量保证的话, 为所有呼叫分配的带宽总和应当小于该链路的容量。
- 呼叫建立信令。以上描述的呼叫准入过程要求一条呼叫沿着它的源到目的地路径

在每台网络路由器中能够预约充足的资源，以确保满足它的端到端 QoS 请求。每台路由器必须决定会话所请求的本地资源，考虑已经承诺给其他进行中的会话的资源量，并决定它是否有足够的资源来满足在这台路由器上的该会话的每跳 QoS 要求，而不违反对所有已经准入的会话做出的本地 QoS 保证。需要一个信令协议来协调这些各种各样的活动，即本地资源的每跳分配，以及该呼叫是否能够在沿着端到端路径上的每台路由器中预约充分的资源。这是如图 9-19 所示的呼叫建立协议（call setup protocol）的任务。资源预留协议（resource ReSerVation Protocol, RSVP）[Zhang 1993, RFC 2210] 就是在因特网体系结构中为此目的所提出的，用于提供服务质量保证。在 ATM 网络中，Q2931b 协议 [Black 1995] 在 ATM 网络的交换机和端点之间承载了这种信息。

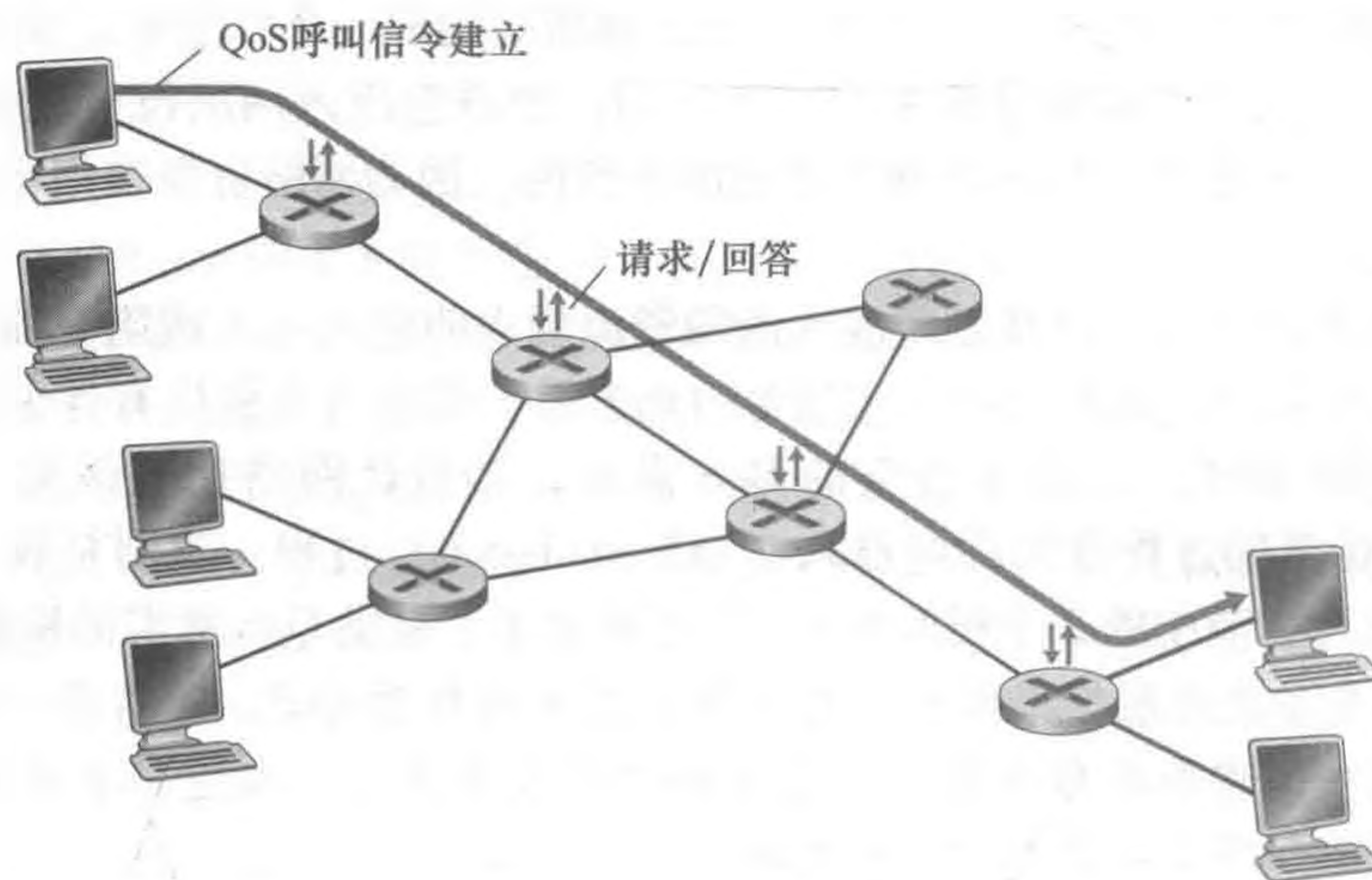


图 9-19 呼叫建立过程

尽管投入了巨大的研究和开发力量，并且甚至开发出了提供每连接服务质量保证的产品，但是几乎没有这种服务的扩展部署。有许多可能的原因。首当其冲的是，情况很可能是我们在 9.2 ~ 9.4 节学习的简单的应用级机制与适当的网络定制相结合（9.5.1 节），为多媒体应用提供了“足够好的”尽力而为的网络服务。此外，ISP 判断：提供每连接服务质量保证所需的附加复杂性以及部署和管理网络的成本，相对于该服务可预测的用户收入，前者可能过高。

## 9.6 小结

多媒体网络是当今因特网中最为激动人心的发展之一。全世界的人们在收音机和电视机面前花费的时间越来越少，而转向使用他们的智能手机和设备接收音频和视频传输，它们包含了实况转播和预先录制的节目。此外，借助于如 YouTube 这样的网站，用户除了是多媒体因特网内容的消费者，也成为内容的生产者。除了视频分发，因特网也正在用于传输电话。事实上，经过下一个 10 年，因特网连同无线因特网接入，可能致使传统的电路交换电话系统成为明日黄花。VoIP 不仅提供了廉价的电话服务，而且将提供大量的增值服务，例如视频会议、在线目录服务、语音消息及融入如脸书和微信等社交网络。

在 9.1 节中，我们描述了视频和语音的内在特点，将多媒体应用分为 3 种类型：(i) 流式存储音频/视频；(ii) 会话式 IP 语音/视频；(iii) 流式实况音频/视频。



在 9.2 节中, 我们更为深入地学习了流式存储视频。对于流式视频应用, 预先录制的视频放置在服务器上, 用户向这些服务器发送请求以按需观看这些视频。我们看到流式视频系统能够被分为两种类型: UDP 流和 HTTP 流。我们观察到对于流式视频的最为重要的性能测量是平均吞吐量。

在 9.3 节中, 我们研究了诸如 VoIP 等会话式多媒体应用能够设计运行在尽力而为网络上的方法。对于会话式多媒体, 定时考虑是重要的, 因为会话式应用是高度时延敏感的。在另一方面, 会话式多媒体应用是容忍丢包的, 即偶尔的丢包仅会在音频/视频播放中引起偶尔的干扰信号, 并且这些丢包常常能被部分地或者全部地掩盖。我们看到将客户缓存、分组序号和时间戳结合起来的方法, 极大地缓解了网络引入的时延抖动的影响。我们也展望了 Skype 所依赖的技术, Skype 是 IP 语音和 IP 视频界的领头公司之一。在 9.4 节中, 我们学习了两种最为重要的用于 VoIP 的标准化协议, 即 RTP 和 SIP。

在 9.5 节中, 我们介绍了几种网络机制 (链路级调度规则和流量监管) 是如何用于在几类流量之间提供区分服务的。

## 课后习题和问题



### 复习题

#### 9.1 节

- R1. 针对 Victor Video 正在观看 4Mbps 视频, Facebook Frank 每 20 秒观看一幅新的 100KB 图片, 以及 Martha Music 正在听 200kbps 音频流的情况, 重新构造 9-1 表。
- R2. 在视频中有两种类型的冗余。描述它们, 并讨论如何能够利用它们进行有效压缩。
- R3. 假定一个模拟音频信号每秒抽样 16 000 次, 并且每个样本量化为 1024 级之一, 该 PCM 数字音频信号的比特率将是多少?
- R4. 多媒体应用能够分为三种类型。阐述它们的名称并对每种类型进行描述。

#### 9.2 节

- R5. 流式多媒体系统能够分为三种类型。阐述它们的名称并对每种类型进行描述。
- R6. 列举 UDP 流的三种缺点。
- R7. 对于 HTTP 流, TCP 接收缓存和客户应用缓存是相同的東西吗? 如果不是, 它们是怎样交互的呢?
- R8. 考虑对于 HTTP 流的简单模型。假设服务器以 2Mbps 的恒定速率发送比特, 并且当已经接收到 800 万比特时开始播放。初始缓存时延  $t_p$  是多少?

#### 9.3 节

- R9. 端到端时延和分组时延抖动的区别是什么? 分组时延抖动的原因是什么?
- R10. 为什么在预定的播放时间之后收到的分组被认为是丢失了?
- R11. 9.3 节描述了两种 FEC 方案, 简要地总结它们。这两种方案通过增加开销而增加了流的传输速率。交织技术也会增加传输速率吗?

#### 9.4 节

- R12. 接收方如何识别在不同会话中的不同 RTP 流? 同一个会话中的不同流是怎样识别的?
- R13. SIP 注册器的作用是什么? 一个 SIP 注册器的作用与移动 IP 中的归属代理的作用有怎样的差异?

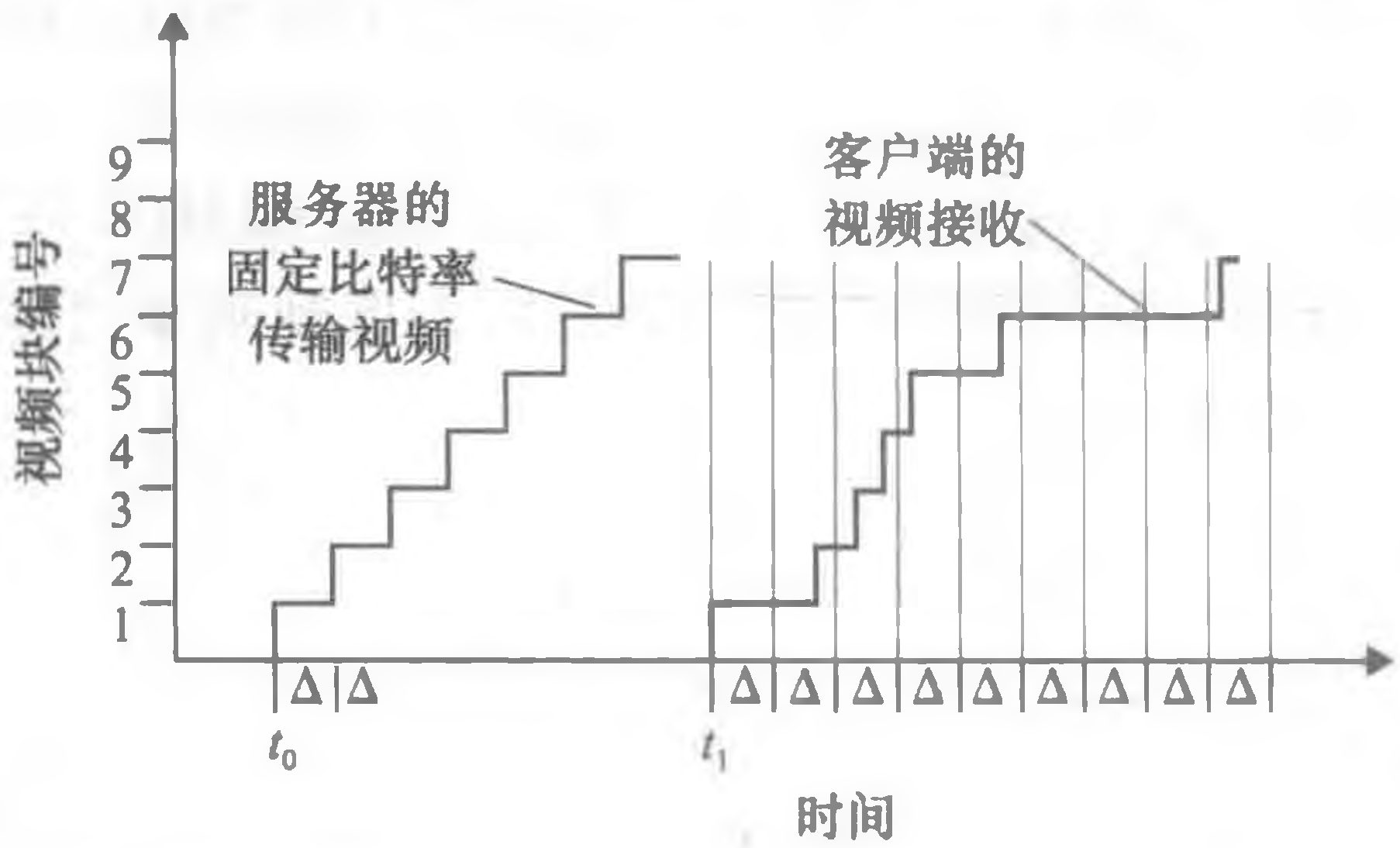


### 习题

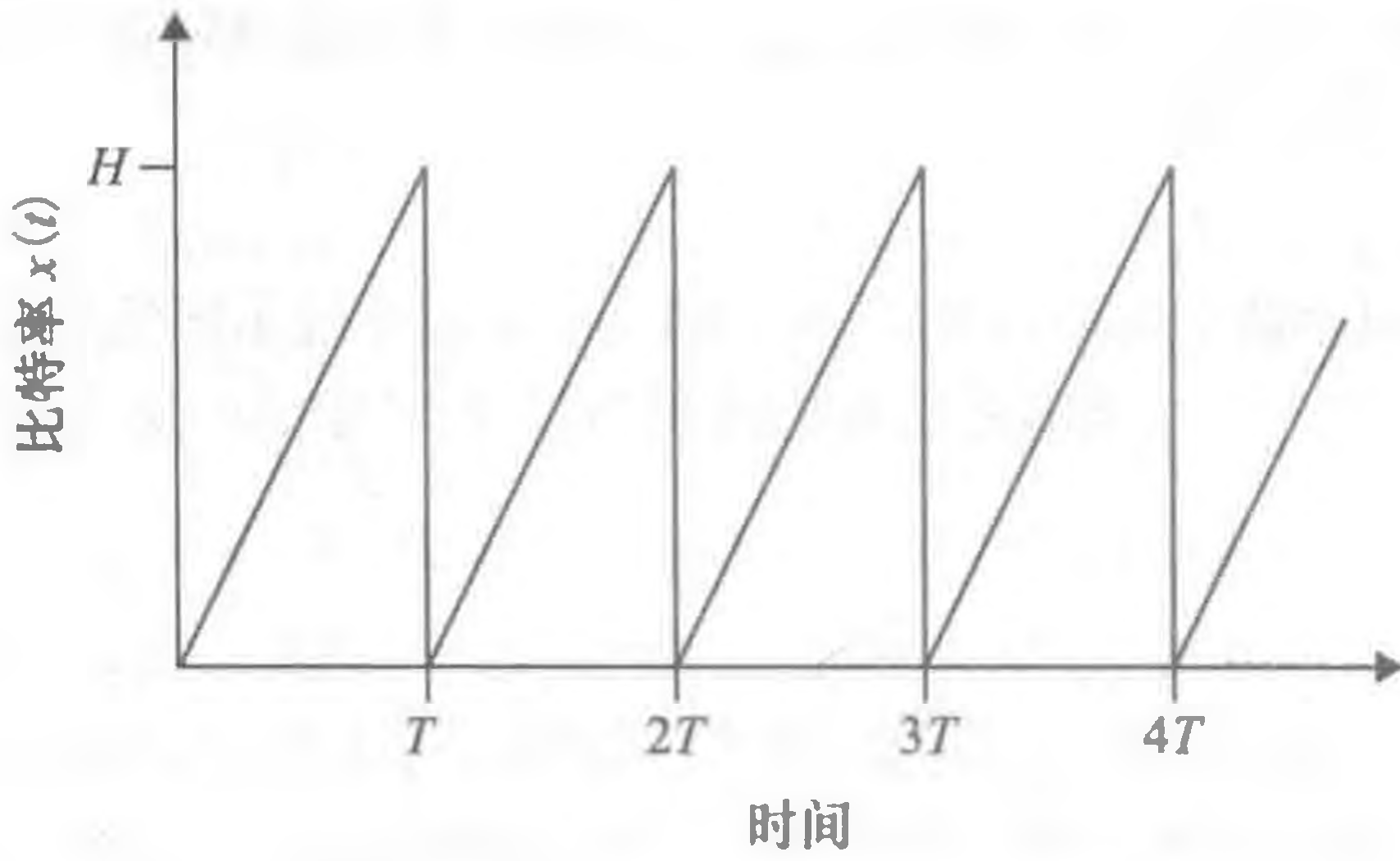
- P1. 考虑下图。类似于我们对图 9-1 的讨论, 假设视频以固定比特率进行编码, 因此每个视频块包含了在相同的固定时间量  $\Delta$  播放的视频帧。服务器在  $t_0$  时刻传输第一个视频块, 在  $t_0 + \Delta$  时刻传输第二

块，在  $t_0 + 2\Delta$  时刻传输第三块，等等。一旦客户开始播放，每个块应当在前一块后播放  $\Delta$  时间单元。

- 假设第一块在  $t_1$  时刻一到达，客户就开始播放。在下图中，多少视频块（包括第一块）将及时到达客户供其播放？解释你怎样得到该答案的。
- 假设该客户现在在  $t_1 + \Delta$  时刻开始播放。多少视频块（包括第一块）将及时到达客户供其播放？解释你怎样得到该答案的。
- 在如上面（b）的相同场景中，存储在客户缓存中等待播放的块的最大数量有多少？解释你怎样得到该答案的。
- 在客户中为使每个视频块及时到达供其播放，最小的播放时延是多少？解释你怎样得到该答案的。

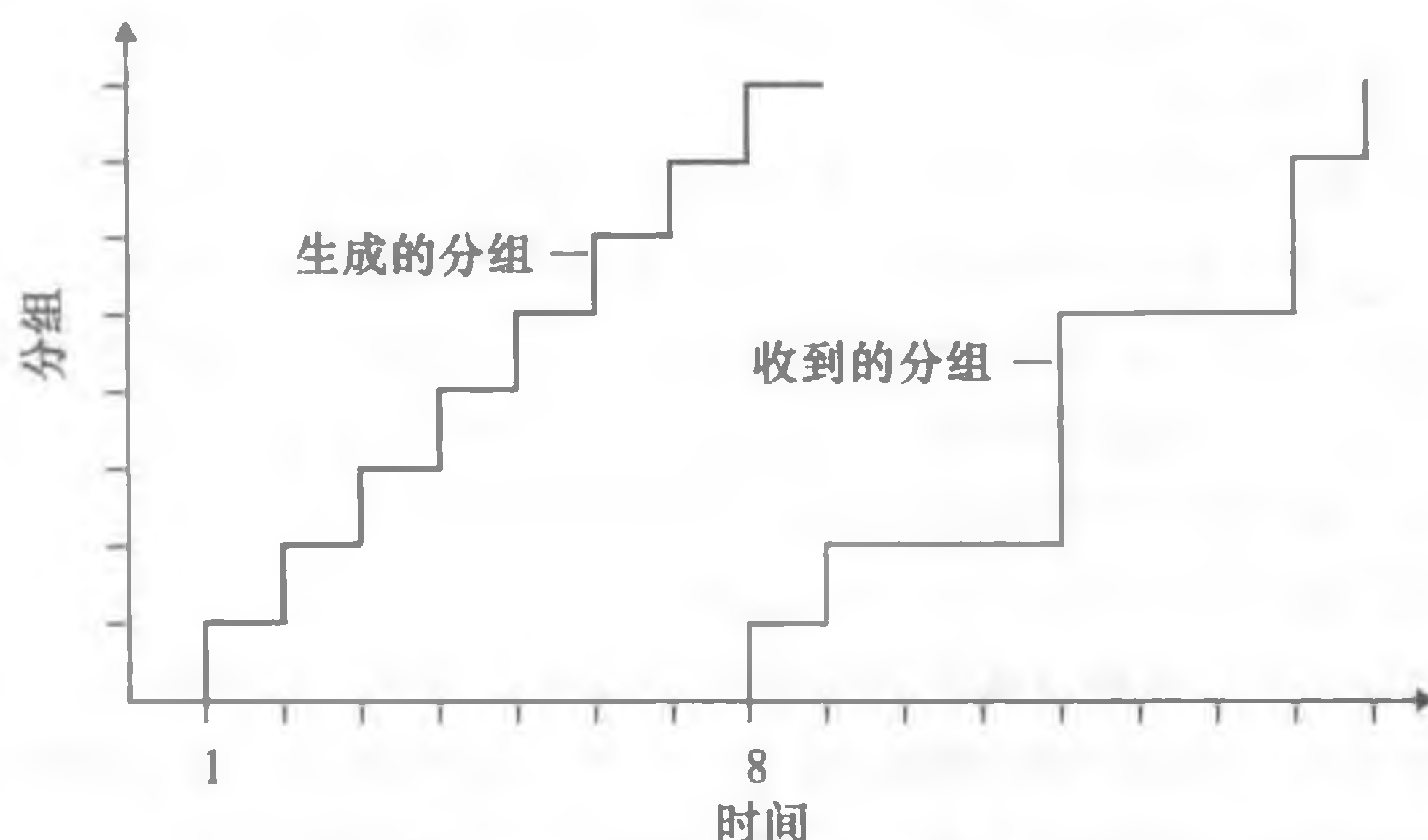


- P2. 回想在图 9-3 中显示的 HTTP 流的简单模型。前面讲过  $B$  表示客户应用缓存的长度， $Q$  表示在客户应用缓存开始播放之前必须被缓存的比特数量。还有  $r$  表示视频消耗速率。假设无论何时客户缓存不满，服务器都以恒定速率  $x$  发送比特。
- 假设  $x < r$ 。如在本书所讨论，在此场合播放将在连续播放期和停滞期之间变动。作为  $Q$ 、 $r$  和  $x$  的函数，确定每个连续播放和停滞期的长度。
  - 现在假设  $x > r$ 。求客户应用缓存变满的时刻  $t = t_f$ ？
- P3. 回想在图 9-3 中显示的 HTTP 流的简单模型。假设缓存长度是有限的，并服务器以可变的速率  $x(t)$  发送比特。具体而言，假设  $x(t)$  具有下列锯齿形状。其速率在时刻  $t = 0$  初始为零，在时刻  $t = T$  线性爬升到  $H$ 。然后反复重复这种模式，如下图中显示的那样。
- 服务器的平均发送速率是什么？
  - 假设  $Q = 0$ ，客户一接收到一个视频块就开始播放。将发生什么现象？
  - 现在假设  $Q > 0$ 。作为  $Q$ 、 $H$  和  $T$  的函数，确定首先开始播放的时间。
  - 假设  $H > 2r$  和  $Q = HT/2$ 。证明在初始播放时延之后将不会有停滞。
  - 假设  $H > 2r$ 。求出最小的  $Q$  值，使在初始播放时延之后将不会有停滞。
  - 现在假定缓存长度  $B$  是有限的。假定  $H > 2r$ 。作为  $Q$ 、 $B$ 、 $T$  和  $H$  的函数，确定当客户应用缓存首次变满的时刻  $t = t_f$ 。





- P4. 回想在图 9-3 中显示的 HTTP 流的简单模型。假设客户应用缓存长度是有限的，服务器以恒定的速率  $x$  发送，并且视频的压缩率是  $r$ ，其中  $r < x$ 。还假设播放立即开始。假设用户早在时刻  $t = E$  中止视频。在中止的时刻，服务器停止发送比特（如果它还没有发送视频中所有比特的话）。
- 假设视频为有限长。问浪费了多少比特（即发送但未观看）？
  - 假设视频为  $T$  秒长，其中  $T > E$ 。问浪费了多少比特（即发送但未观看）？
- P5. 考虑一个 DASH 系统，其中有  $N$  个视频版本（有  $N$  种不同的速率和质量）和  $N$  种音频版本（有  $N$  种不同的速率和质量）。假设我们要允许播放者在任何时间对任何  $N$  种视频版本和任何  $N$  种音频版本进行选择。
- 如果我们生成文件，使音频与视频混合起来，这样服务器在给定的时间仅发送一种媒体流，该服务器将需要存储多少文件（每个文件有一种不同的 URL）？
  - 如果服务器单独发送音频流和视频流并且让客户同步这些流的话，该服务器将需要存储多少文件？
- P6. 在 9.3 节的 VoIP 例子中，令  $h$  为加到每个块中的首部字节总数，包括 UDP 和 IP 的首部。
- 假设每 20 秒发射一个 IP 数据报，求出由该应用一侧产生的数据报的每秒传输速率（以比特计）。
  - 当使用 RTP 时， $h$  的典型值是什么？
- P7. 考虑在 9.3 节中描述估计平均时延  $d_i$  的过程。假设  $u = 0.1$ 。令  $r_1 - t_1$  是最近的采样时延，令  $r_2 - t_2$  是下一个最近的采样时延，等等。
- 对于一个特定的音频应用，假设四个分组到达接收方的采样时延为  $r_4 - t_4$ 、 $r_3 - t_3$ 、 $r_2 - t_2$  和  $r_1 - t_1$ 。根据这 4 个采样来表示时延  $d$  的估计值。
  - 对于  $n$  个采样时延归纳出公式。
  - 对于 b 小题的公式，令  $n$  趋于无穷，给出最后的公式。评论一下为什么这个平均过程被称为一个指数移动平均数。
- P8. 重复习题 P7 中的 (a) 和 (b)，求出平均时延偏差的估计值。
- P9. 对于 9.3 节中的 VoIP 例子，我们引入了一个估计时延的在线过程（指数移动平均数）。在本习题中我们将研究另一种过程。设  $t_i$  是接收到的第  $i$  个分组的时间戳；设  $r_i$  是收到第  $i$  个分组的时间。令  $d_n$  表示在收到第  $n$  个分组后我们对平均时延的估计。在收到第一个分组后，我们设置时延估计值等于  $d_1 = r_1 - t_1$ 。
- 假设我们希望对于所有的  $n$ ，有  $d_n = (r_1 - t_1 + r_2 - t_2 + \cdots + r_n - t_n) / n$ 。根据  $d_{n-1}$ 、 $r_n$  和  $t_n$ ，给出  $d_n$  的递归公式。
  - 描述一下对于 VoIP，为什么在 9.3 节中描述的时延估计比 (a) 部分叙述的时延估计更合适。
- P10. 比较 9.3 节描述的估计平均时延的过程和 3.5 节估计往返时间的过程。这些过程有什么相同？有什么不同？
- P11. 考虑下图（它类似于图 9-3）。某发送方在  $t = 1$  时开始周期性地发送分组化的音频。在  $t = 8$ ，第一个分组到达接收方。



- 分组 2 到分组 8 的时延（从发送方到接收方，忽略任何播放时延）是什么？注意到在该图上每个

垂直和水平线段具有 1、2 或 3 个时间单元的长度。

- 如果第一个分组在  $t=8$  时刻一到达接收方，音频就开始播放，发送的前 8 个分组中的哪些将不能按时到达进行播放？
- 如果在  $t=9$  时刻音频开始播放，发送的前 8 个分组中的哪些将不能按时到达进行播放？
- 在接收方，导致所有前 8 个分组按时到达进行播放的最小播放时延是什么？

P12. 再次考虑习题 P11 中的图，显示分组音频传输和接收时间。

- 计算分组 2 到分组 8 的估计时延，使用 9.3.2 节中的对于  $d_i$  的公式。使用  $u=0.1$  的值。
- 使用 9.3.2 节中对于  $v_i$  的公式，从对分组 2 到分组 8 的估计平均时延计算其估计的偏差。

P13. 在 9.3 节中讲过 VoIP 的两种 FEC 方案。假设第一种方案为每 4 个初始块产生一个冗余块。假设第二种方案使用传输速率为标称流传输速率的 25% 的低比特率编码。

- 每种方案需要多少额外带宽？每种方案增加多少播放时延？
- 如果在每组的 5 个分组中第一个分组丢失了，这两种方案如何执行？哪一种方案有更好的音频质量？
- 如果在每组的 2 个分组中第一个分组丢失了，这两种方案如何执行？哪一种方案有更好的音频质量？

P14. a. 考虑 Skype 中的一个音频会议具有  $N>2$  个与会者。假定每个与会者生成一个速率  $r$  bps 的恒定流。呼叫的发起方将需要每秒发送多少比特？其他  $N-1$  个与会者每个都将需要每秒发送多少比特？聚合所有与会者，总的发送速率是多少？

- 对于使用一台中心服务器的一个 Skype 视频会议，重复 (a)。
- 重复 (b)，只是现在情况为每个对等方发送其视频流副本到其他  $N-1$  个对等方。

P15. a. 假设我们向因特网发送两个 IP 数据报，每个数据报携带不同的 UDP 段。第一个数据报的源 IP 地址为 A1，目的 IP 地址为 B，源端口为 P1，目的端口为 T。第二个数据报的源 IP 地址为 A2，目的 IP 地址为 B，源端口为 P2，目的端口为 T。假设 A1 和 A2 不同，P1 和 P2 不同。假设这两个数据报都到达它们的目的地址，这两个 UDP 数据报会被同一个套接字接收吗？为什么？

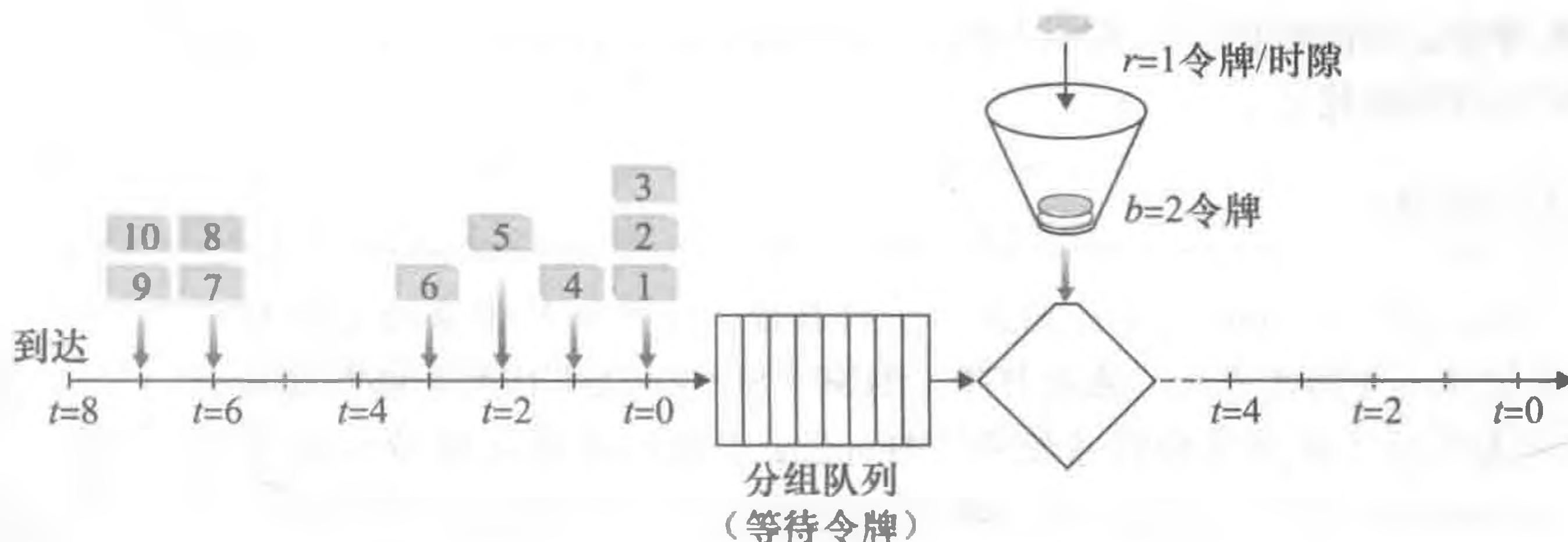
- 假设 Alice、Bob 和 Claire 要使用 SIP 和 RTP 来进行音频会议呼叫。Alice 与 Bob 和 Claire 之间发送和接收 RTP 分组，只有一个 UDP 套接字足够吗（SIP 报文所需的套接字除外）？如果够，那么 Alice 的 SIP 客户如何区分 RTP 分组是来自 Bob 还是来自 Claire？

P16. 是非判断题：

- 如果存储视频直接从 Web 服务器流式传输到媒体播放器，这个应用则正在使用 TCP 作为底层的传输协议。
- 当使用 RTP 时，发送方有可能在会话的中间改变编码。
- 所有使用 RTP 的应用必须使用端口 87。
- 假设一个 RTP 会话对每个发送方有独立的音频和视频流，则这些音频和视频流使用同样的 SSRC。
- 在区分服务中，尽管每跳行为定义了各类型之间的性能差别，但它没有强制要求为了获得这些性能而使用任何特定机制。
- 假设 Alice 要和 Bob 建立一个 SIP 会话。在她的 INVITE 报文中包括了这样的行：`m = audio 48753 RTP/AVP 3`（AVP 3 指示 GSM 音频）。因此 Alice 在该报文中指示她要发送 GSM 音频。
- 参考前一句说法，Alice 在她的 INVITE 报文中指示了她将把音频发送到端口 48753。
- SIP 报文在 SIP 实体之间通常使用一个默认的 SIP 端口号发送。
- 为了维护注册，SIP 客户必须周期地发送 REGISTER 报文。
- SIP 强制所有的 SIP 客户支持 G.711 音频编码。

P17. 考虑下图，图中显示了一个由分组流提供输入的漏桶监管器。令牌桶能够保持最多两个令牌，并最初在  $t=0$  时刻是满的。新的令牌以每时隙 1 个令牌的速率到达。输出链路速率采用下列规则：如果两个分组在一个时隙的开始获得令牌，它们能够在相同的时隙中进入输出链路。该系统的定时细节如下：





1. 分组在时隙的开始到达（如果有的话）。因此在该图中，分组 1、2 和 3 在时隙 0 到达。如果在队列中已经有分组，则到达的分组加入该队列的尾部。分组以 FIFO 方式朝着该队列的前方行进。
2. 当到达分组增加进队列之后，如果有任何排队分组，那些分组中的一个或两个（取决于可用令牌的数量）将每个从令牌桶中去除一个令牌，并在那个时隙去往输出链路。因此，分组 1 和 2 从桶中去除一个令牌（因为最初有两个令牌）并在时隙 0 期间去往输出链路。
3. 如果令牌桶未满载则新令牌加入，因为令牌产生速率是  $r=1$  令牌/时隙。
4. 然后时间前进到下一个时隙，并且重复这些步骤。

回答下列问题：

- a. 对每个时隙，在到达分组处理后（上述步骤 1）但在任何分组通过队列传输并去除一个令牌之前的那个时刻，指出位于队列中的分组和位于桶中的令牌数量。因此，对于在上述例子中的  $t=0$  时刻的时隙，分组 1、2 和 3 位于队列中，桶中有两个令牌。
- b. 对每个时隙，指出在令牌从队列中去除后哪个分组出现在输出链路。这样，对于上述例子中的  $t=0$  时隙，在时隙 0 期间，分组 1 和 2 从漏桶出现在输出链路上。

P18. 重复习题 P17，但假设  $r=2$ 。再次假设初始时桶是满的。

P19. 考虑习题 P18 并假设现在  $r=3$  且  $b=2$  与以前一样。你对上述问题的回答有变化吗？

P20. 考虑一下监管分组流的平均速率和突发长度的漏桶监管器。我们现在也要监管峰值速率  $p$ 。说明这个漏桶监管器的输出如何能够提供给第二个漏桶监管器，使得这两个串行的漏桶能够监管平均速率、峰值速率和突发长度。要给出第二个监管器的桶长度和令牌产生速率。

P21. 如果对于任何  $t$  来说，在每个时间间隔长度  $t$  内到达漏桶的分组数小于  $rt+b$  个，那么就说分组流符合一个突发长度  $b$  和平均速率  $r$  的漏桶规范  $(r, b)$ 。一个符合漏桶规范  $(r, b)$  的分组流必须在参数  $r$  和  $b$  的漏桶监管器那里等待吗？评估你的答案。

P22. 说明只要  $r_1 < R w_1 / (\sum w_j)$ ，那么  $d_{\max}$  实际上是流 1 中任何分组在 WFQ 队列中要经受的最大时延。



## 编程作业

在这个实验中，将实现流式视频服务器和客户。该客户使用实时流协议（RTSP）来控制服务器的动作。服务器使用实时协议（RTP）来分组化视频以便在 UDP 上传输。将为你提供在客户和服务端中部分实现了 RTSP 和 RTP 的 Python 代码，你的工作是完成客户和服务端代码。当完成时，你将已经创建了一个进行下面工作的客户 - 服务器应用：

- 客户发送 SETUP、PLAY、PAUSE 和 TEARDOWN 等 RTSP 命令，并且服务器应答这些命令。
- 当服务器处于播放状态，它周期地抓取存储的 JPEG 帧，用 RTP 对该帧分组化，并将该 RTP 分组发送到一个 UDP 套接字中。
- 客户接收该 RTP 分组，去除 JPEG 帧，解压缩该帧，并在客户的监管器上再现该帧。

为你提供的代码在服务器实现 RTSP 协议，在客户实现对 RTP 解分组化。这个代码也考虑传输视频的显示。你需要在客户实现 RTSP 和 RTP 服务器。该编程作业将极大地增强学生对 RTP、RTSP 和流式视频的理解。我们极力推荐它。该作业也建议进行大量可选的练习，包括在客户和服务端实现 RTSP 的