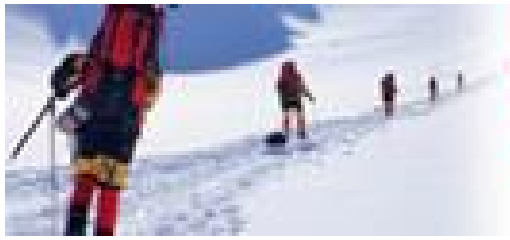


Data Communications and Networking Fourth Edition

Forouzan

第24章

拥塞控制和服务质量

24-1 数据通信量

- 对拥塞控制和服务质量关注的重要方面是数据通信量；
- 在拥塞控制中，要设法避免通信量拥塞；
- 在服务质量中，要设法为通信量创造合理的环境

图24.1 通信量描述符（描述数据流的定性值）

p平均数据速率、峰值数据速率、最大突发长度；

p有效带宽是指网络需要分配给通信流的带宽，它是三个值的函数：平均数据速率、峰值数据速率和最大突发长度（计算过程复杂）。

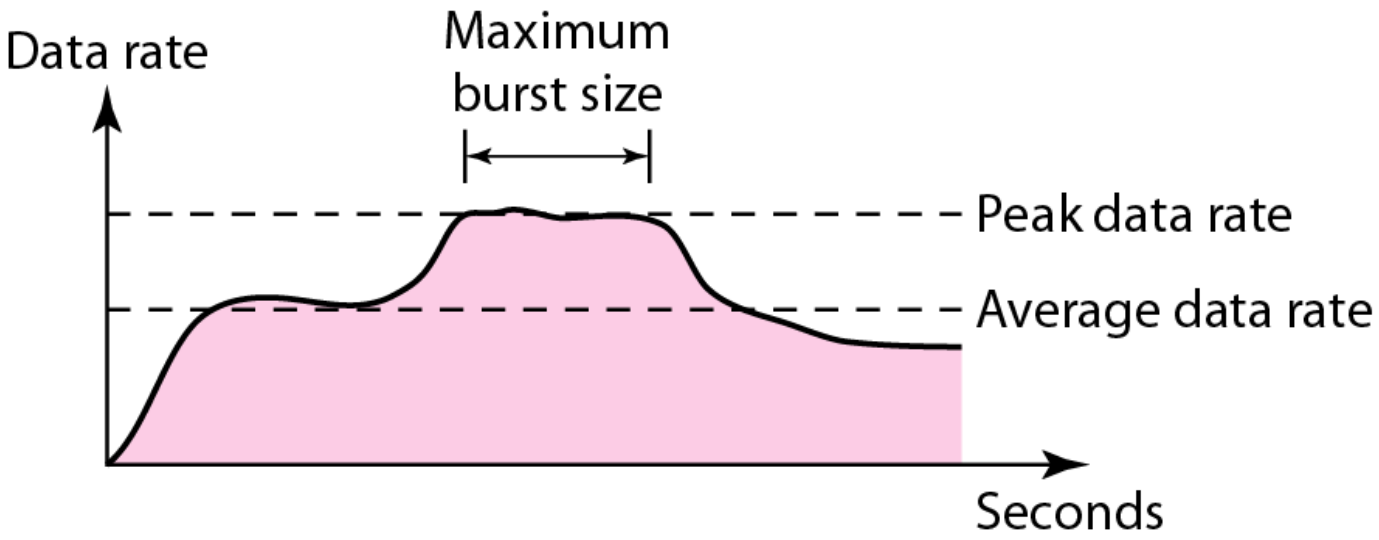
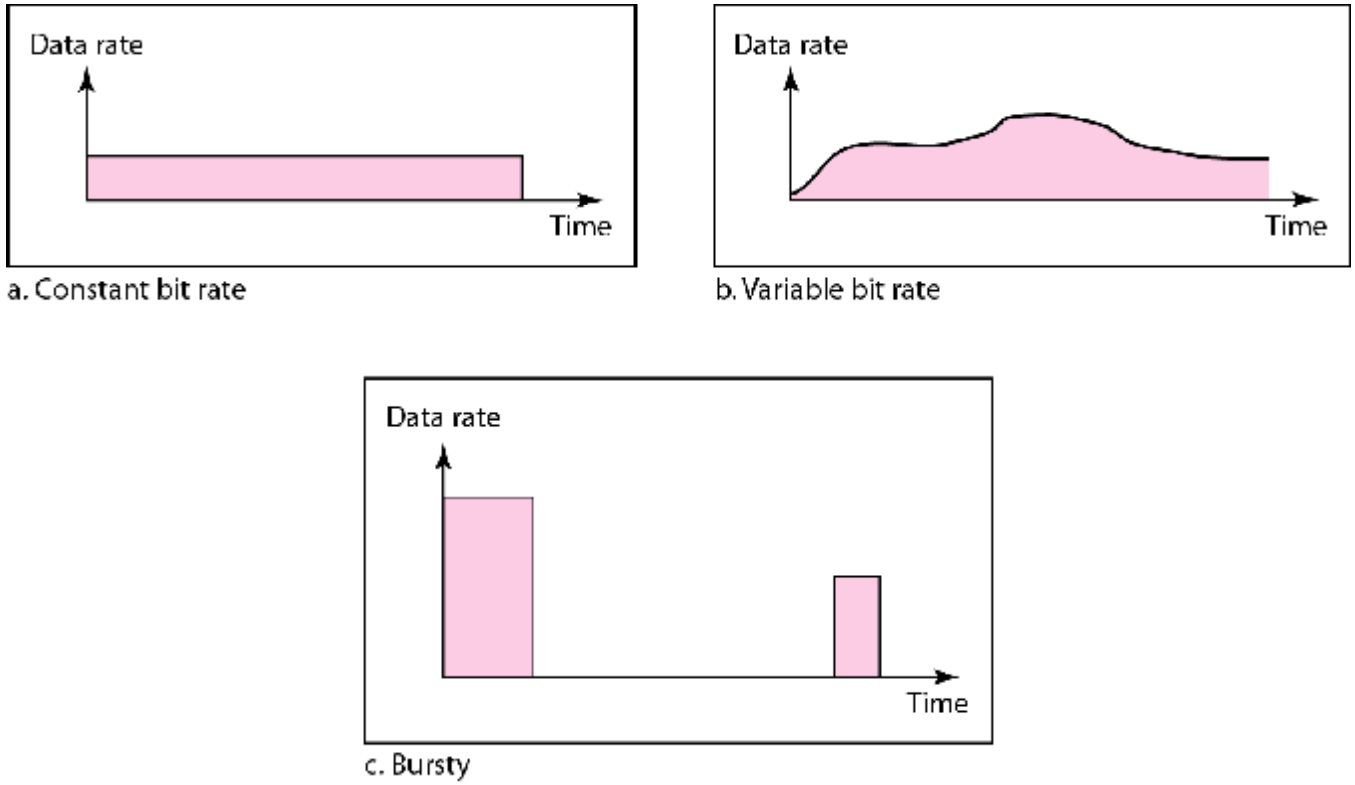


图24.2　三个通信量特征值

p数据流可用以下通信量特征值之一来表述：恒定比特率（容易处理）、可变比特率（较难处理，一般不需要重新整形）或突发性（最难处理，通常需要重新整形）



24-2 拥塞

p 如果网络中的载荷（load），即发送到网络中的分组数量超过了网络的容量（即网络中能处理的分组数量），那么在网络中就可能发生拥塞；

p 拥塞控制（congestion control）指的是控制拥塞和使载荷低于网络容量的机制和技术；

p 拥塞可能发生在任何涉及等待机制的系统中

图24.3 路由器中的队到

p如果分组的到达速率高于分组的处理速率，那么输入队列会变得越来越长；

p如果分组的转发速率低于分组的处理速率，输出队列也会变得越来越长。

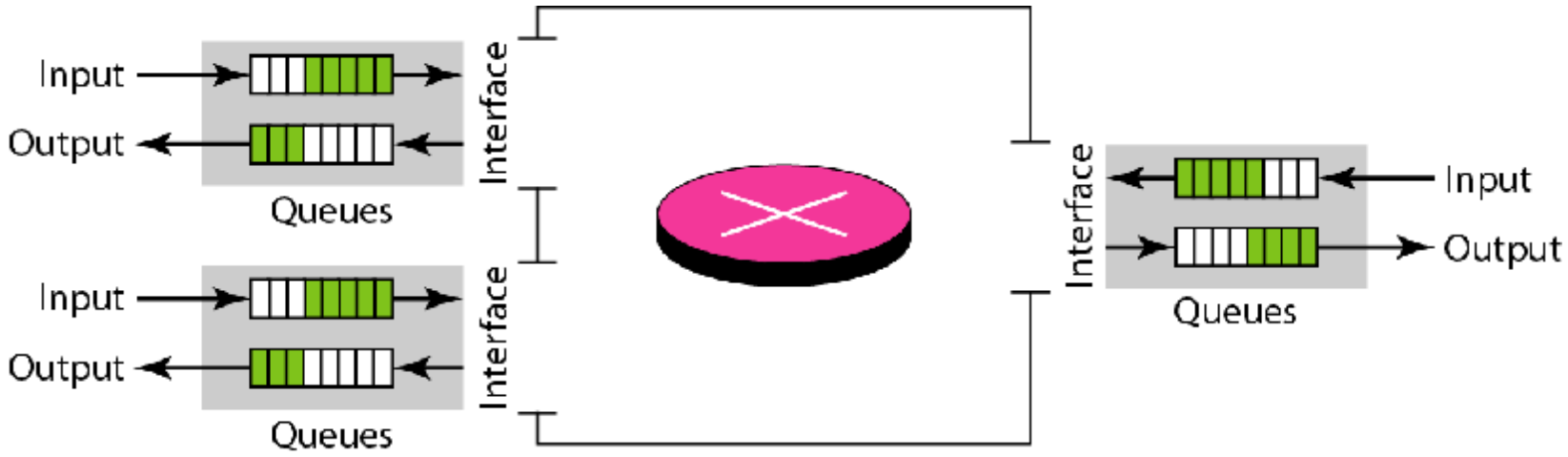
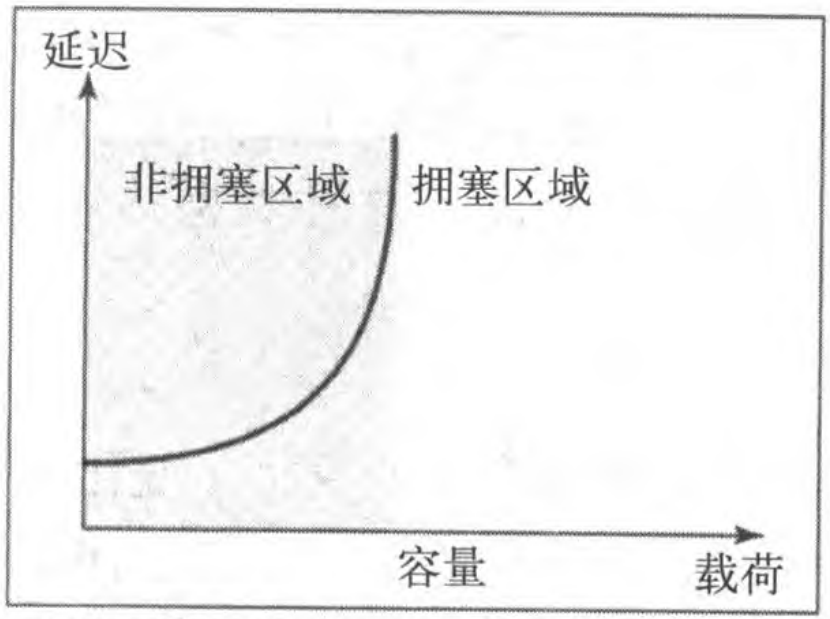
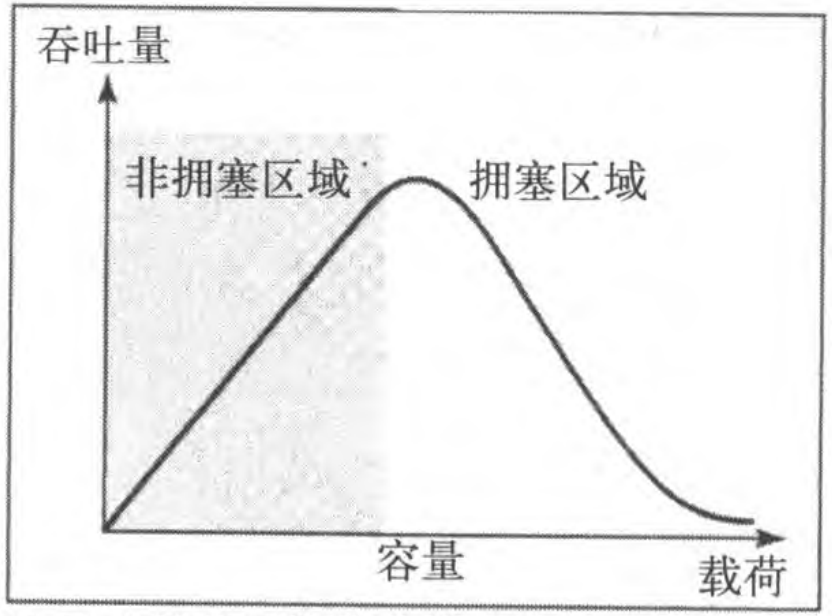


图24.4 分组延迟和吞吐量是网络载荷的函数

- 拥塞控制包括了两个测试网络性能的要素：延迟和吞吐量；
- 将网络的吞吐量（throughput）定义为单位时间内通过网络的分组数量（载荷小于容量时，随载荷成比例增长；载荷超过容量时，吞吐量将降低）



a) 分组延迟是网络载荷的函数

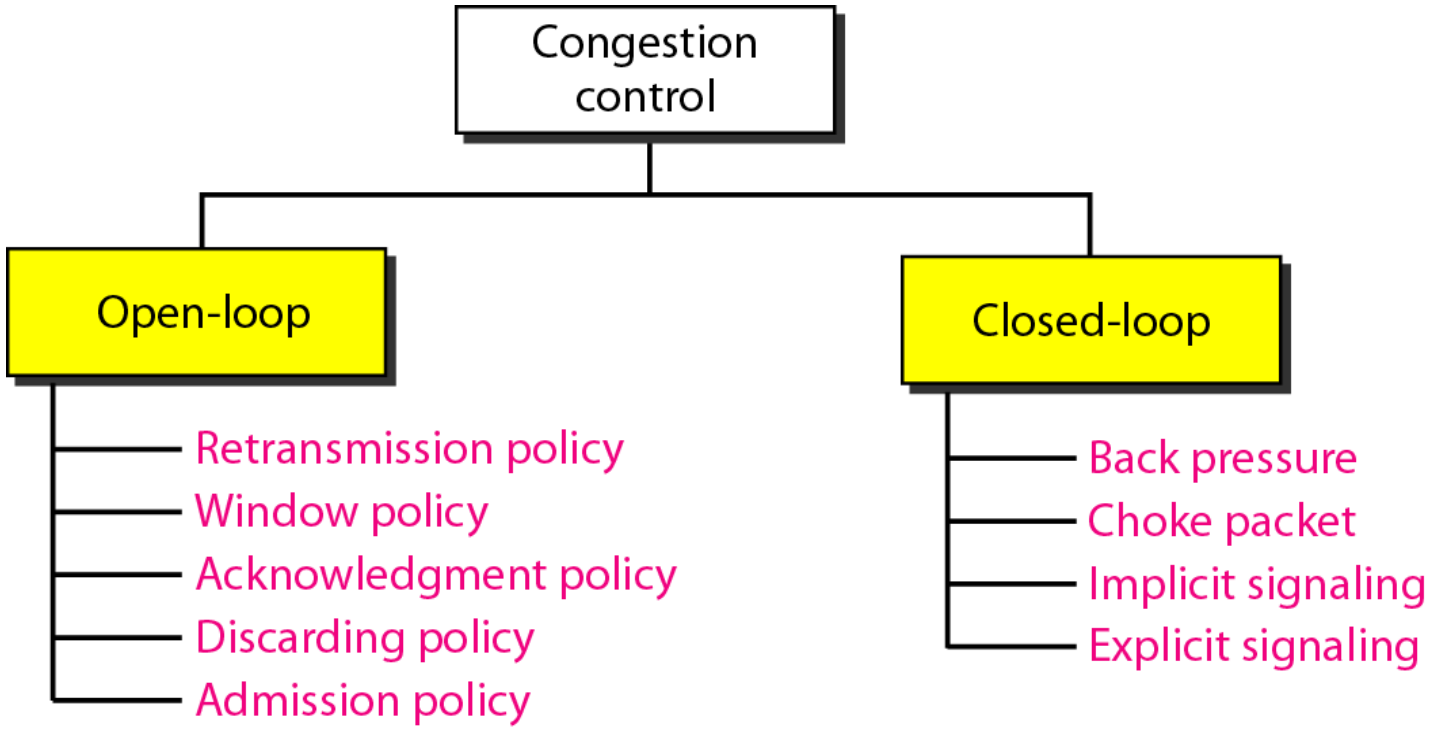


b) 吞吐量是网络载荷的函数

24-3 拥塞控制

p拥塞控制是指在拥塞发生之前预防拥塞、或在拥塞发生之后消除拥塞的技术和机制；

p通常把拥塞控制分为两大类：开环拥塞控制（阻止或预防，不让拥塞发生）和闭环拥塞控制（缓解，发生后减轻）



开环拥塞控制

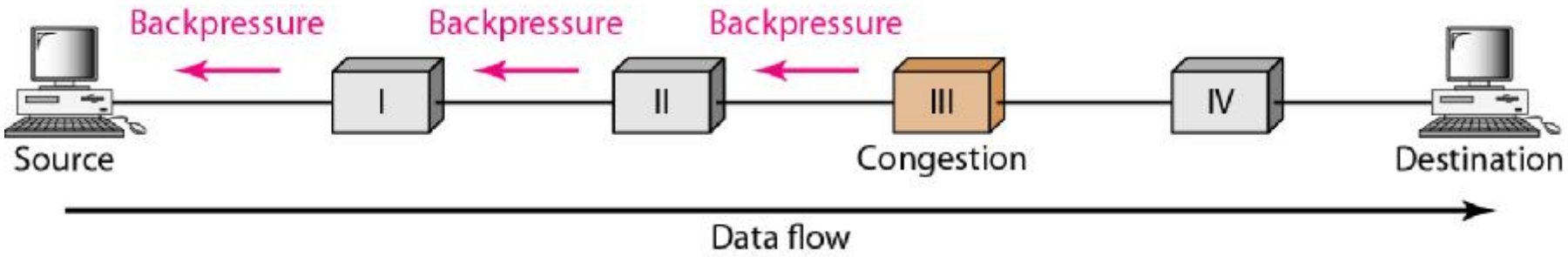
- 重传策略：一个好的重传策略能预防拥塞，必须优化设计重传策略和重传定时器，使之高效，并用来预防拥塞；
- 窗口策略：发送方窗口的类型也会影响拥塞，对拥塞控制而言，选择性重复窗口要优于回退N 帧窗口；
- 确认策略：确认也是网络中负载的一个部分，发送确认越少意味着网络的负载越轻；
- 丢弃策略：路由器使用好的丢弃策略可以预防拥塞，同时不破坏传输的完整性（比如丢弃声音中不敏感的分组）；
- 许可策略：虚电路网络中，允许数据流进入网络之前，与数据流通路相连的交换机首先检查其资源需求，如果网络有拥塞或可能出现拥塞，路由器将拒绝建立虚电路。

闭环拥塞控制-背压

p一个拥塞点停止接收来自直接上行节点或一些近邻节点的数据，这会引起上行节点或一些近邻节点发生拥塞，它们依次拒绝它们的上行节点或一些近邻节点的数据，依次类推；

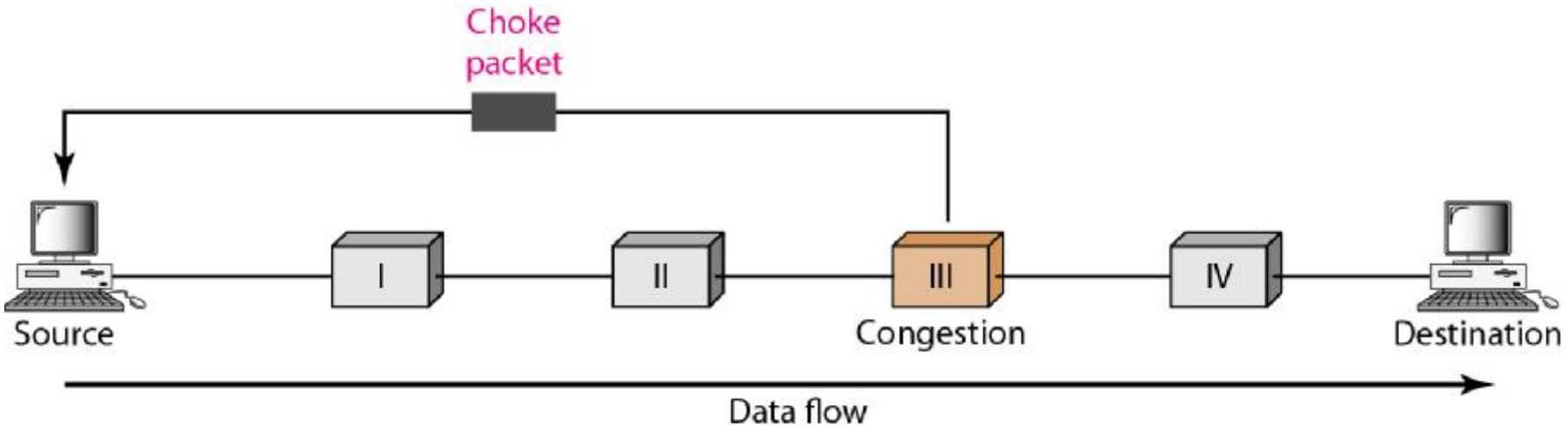
p背压是点到点拥塞控制，它从一点开始，然后传播，数据流反方向到达源端；

p背压技术仅用于虚电路网络，在虚电路网络中，每个节点知道数据的流是来自它的上行节点



闭环拥塞控制-抑制分组

- 该分组由节点发送给源端，通知它发生拥塞的情况；
- 在背压方法中，警告从一个节点到它的上行节点，虽然警告可能最后到达源端；而在抑制分组方法中，警告从已经发生拥塞的路由器直接传到源端，中间经过的节点没被警告；
- 例如，ICMP源站抑制报文，警告报文直接发送给源站点，中间的路由器不进行任何处理



闭环拥塞控制-隐含信令和显示信令

- 隐含信令：与源端之间没有通信，源端从其他征兆中察觉某处有拥塞；例如源端发送多个分组，暂时还没有收到确认时，可能发生了拥塞；在接收确认过程中发生的延迟现象就可以认为网络发生了拥塞：源端应该降低发送速率；
- 显式信令：拥塞节点显式通知源端或目的端发生了拥塞，但显式信令方法与抑制分组方法不同；在抑制分组方法中，有一个单独的分组用于此目的，而在显式信令方法中，信号包含在携带数据的分组中，包含两种类型：
  - Ø后向信令：将一个位设置在分组中，并使之向着与拥塞发生方向相反的方向移动，它提示源端网络发生了拥塞；
  - Ø前向信令：将一个位设置在分组中，并使之向着发生拥塞的方向移动，它提示目的端网络发生了拥塞。

24-4 示例-TCP拥塞控制

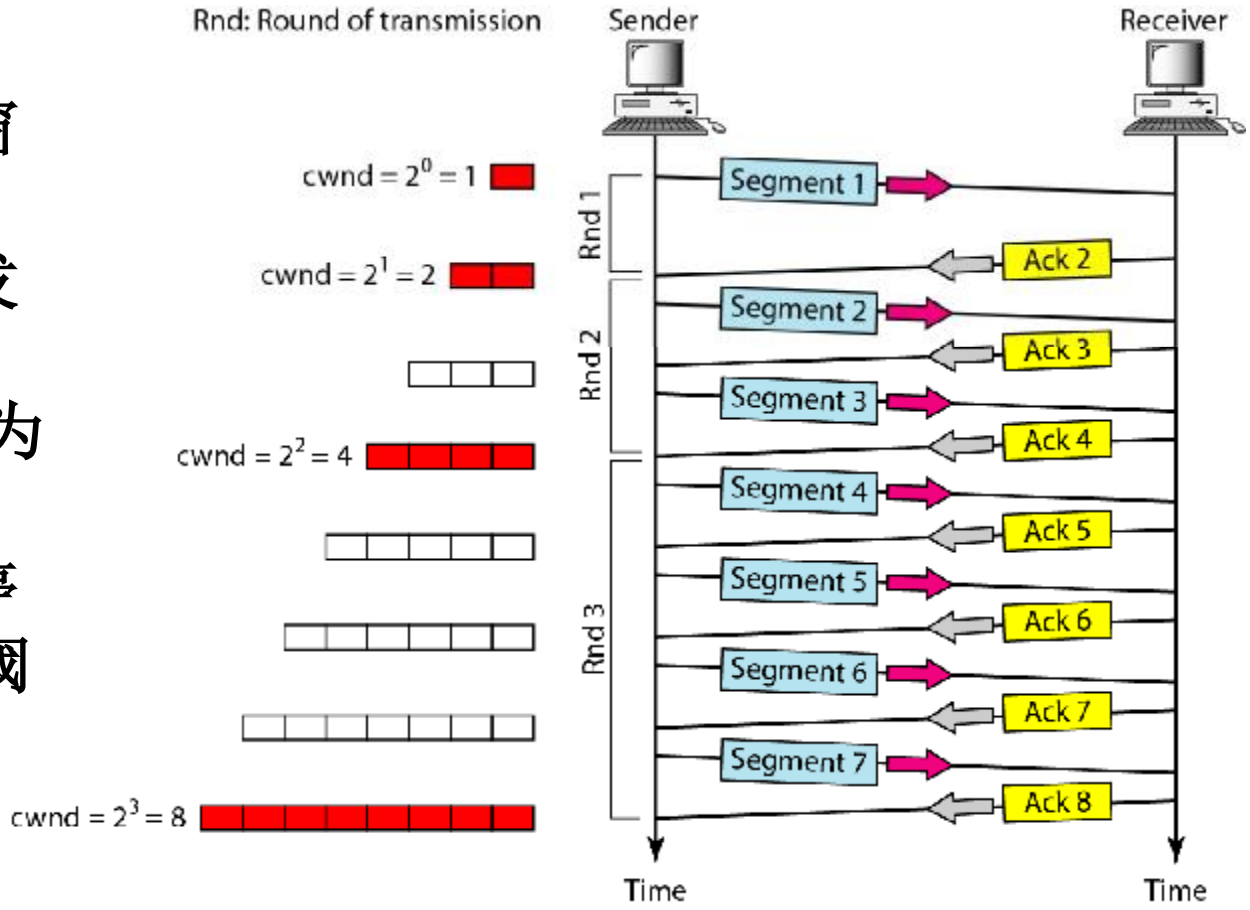
- 接收端窗口rwnd：是接收端根据其目前的接收缓存大小所许诺的最新的窗口值，是来自接收端的流量控制；接收端将此窗口值放在TCP报文首部中的窗口字段，传送给发送端（可以理解为给发送端的信用）；
- 拥塞窗口cwnd：是发送端根据自己估计的网络拥塞程度而设置的窗口值，是来自发送端的流量控制；
- 发送端发送窗口的上限值应当取接收端窗口rwnd和拥塞窗口cwnd两个变量中较小的一个；
- 当rwnd < cwnd时，是接收端的接收能力限制发送窗口的最大值；
- 当cwnd < rwnd时，则是网络的拥塞限制发送窗口的最大值。

TCP拥塞控制策略（即如何确定cwnd？）

- TCP处理拥塞的一般策略基于三个阶段：慢速启动、拥塞避免和拥塞检测；
- 在慢速启动阶段，发送方用很慢的传输速率启动，但迅速地增加到阈值；
- 拥塞避免阶段：在达到阈值时，为了避免拥塞而降低数据速率；
- 最后，如果检测到拥塞（任何时刻），则发送方基于如何检测到拥塞而返回到慢速启动或拥塞避免阶段。

图24.8 慢速启动：指数增长

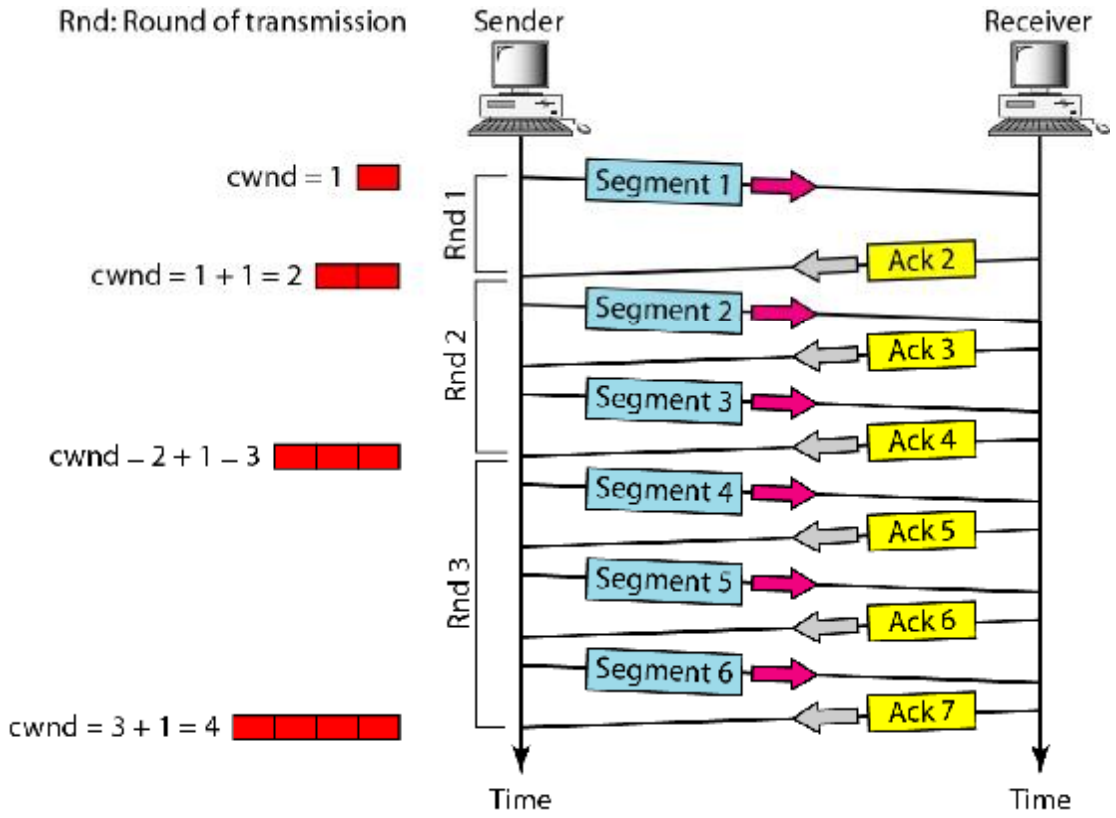
- 开始cwnd=1×MSS，慢速启动；
- 每收到一个确认，窗口大小增加一个MSS；
- 每次传输后（收到发送窗口的所有确认），cwd变为2^n×MSS，n为传输次数，指数增长；
- 慢速启动到达阈值停止该阶段（多数实现阈值为65535字节）



在慢速启动算法中，拥塞窗口大小按指数规律增长直到达到阈值。

图24.9 拥塞避免：加性增加

- 为了在拥塞发生之前避免拥塞，必须降低指数增长的速度；
- 当拥塞窗口大小达到慢速启动的阈值时，慢速启动阶段停止，加性增加阶段开始；
- 在拥塞避免算法中，每次整个窗口所有段都被确认（一次传输）时，拥塞窗口才增加1。



在拥塞避免算法中，拥塞窗口大小是加性增加的直到检测到拥塞。

拥塞检测：乘性减少

- 如果发生拥塞，拥塞窗口的大小必须减小；
- 发送方能推测出发生拥塞现象的唯一方法是通过重传段的要求，重传在两种情况下发生；
- 若计时器到时，存在着非常严重拥塞的可能性：一个段可能已丢失且没有关于该段的消息，此时TCP做出强烈反应：
  - a. 设置阈值为当前（发生拥塞时）窗口大小的一半；
  - b. 设置cwnd为一个段的大小；
  - c. 启动慢速启动阶段。
- 若接收到三个重复ACK，存在轻度拥塞的可能性：一个段可能已丢失但有些段可能已安全到达，此时TCP做出轻度反应：
  - a. 设置阈值为当前（发生拥塞时）窗口大小的一半；
  - b. 设置cwnd为阈值（有些实现是阈值加上三个段）；
  - c. 启动拥塞避免阶段。

图24.10 TCP拥塞策略总结（P510）

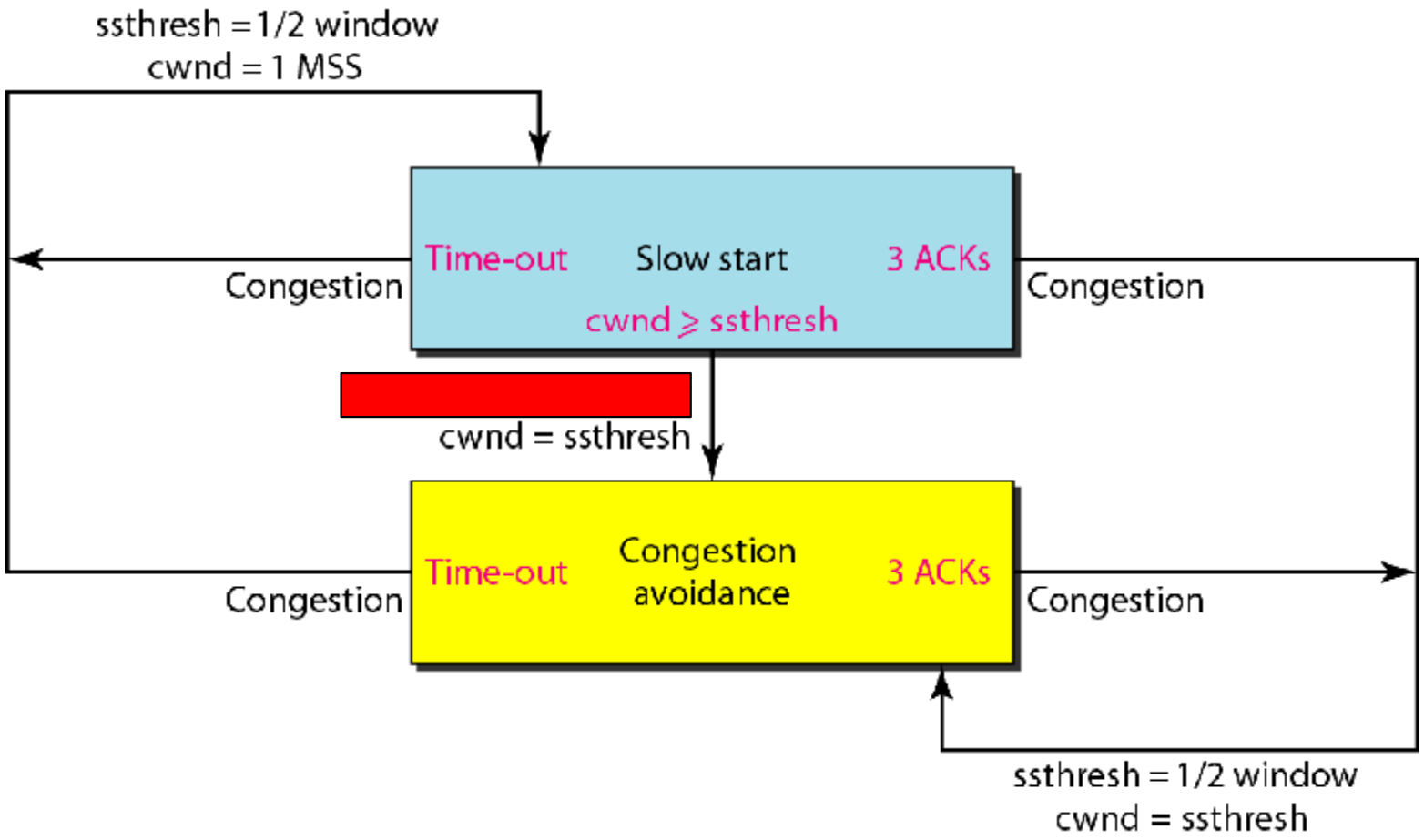
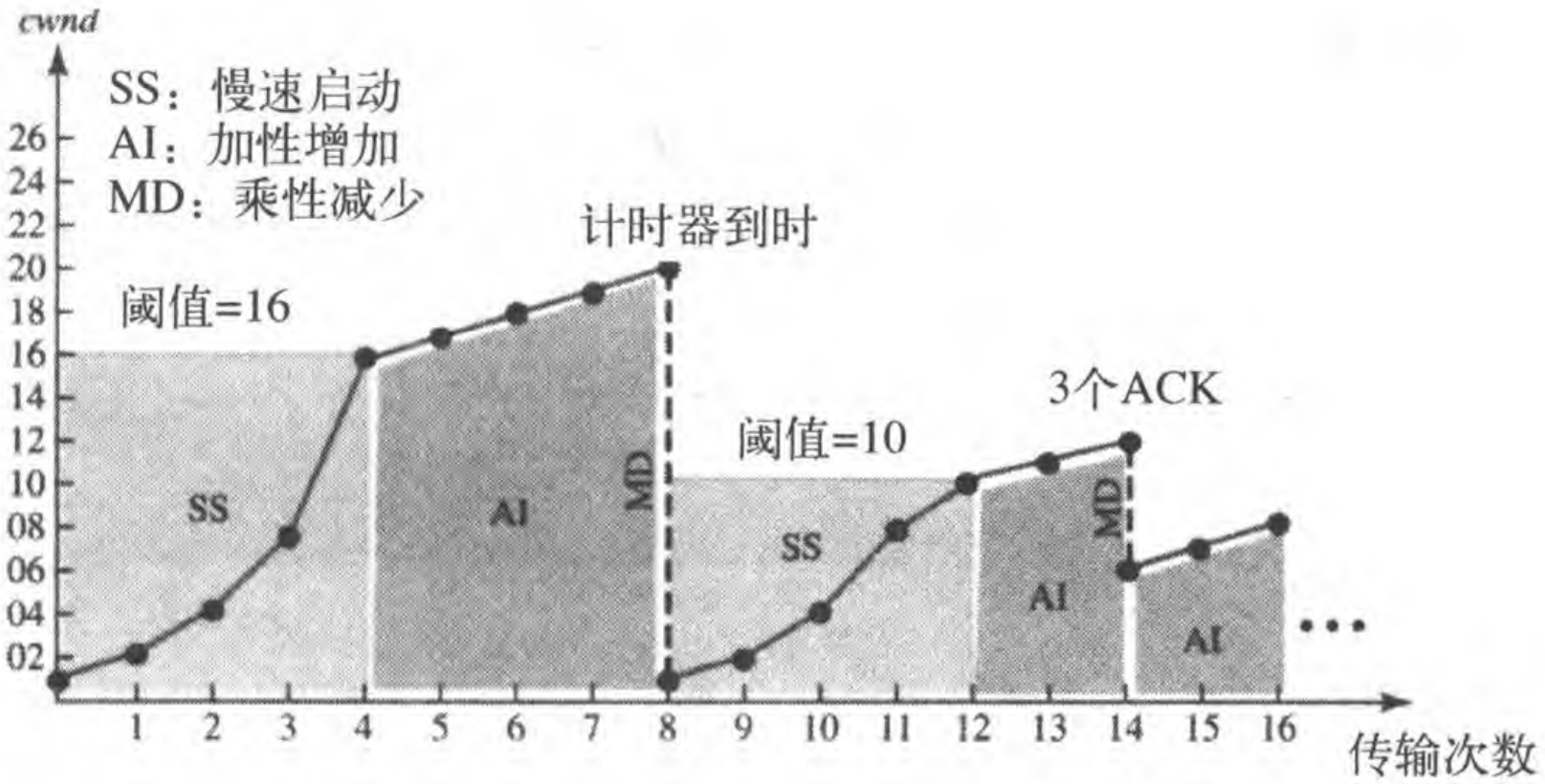


图24.11 拥塞例子（P510）



24-5 服务质量

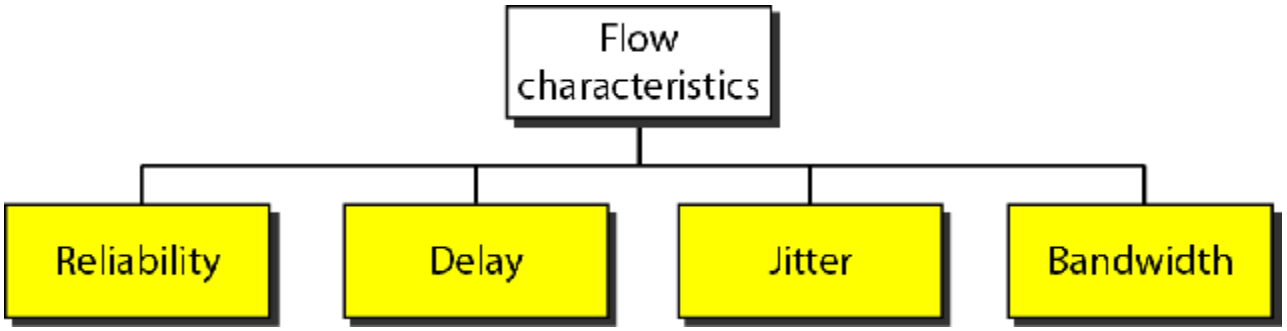
p 服务质量（quality of service，QoS）是一个网络互联问题，对该问题的讨论已经远远超出对它的定义；

p 可以非形式地将服务质量定义为数据流所追求的某种目标。

图24.15 数据流特性

p按传统的观点，数据流具有四种特性：可靠性、延迟、抖动和带宽；

p其中，抖动指属于同一数据流的分组延迟的变化



24-6 改进QoS的技术

- 改进服务质量的四种常用方法：调度、通信量整形、许可控制和资源预留；
- 其中，资源预留提前对数据流需要的缓冲区、带宽、CPU时间等资源进行预留；
- 许可控制指的是由路由器或交换机使用的一种机制，它基于一个称为数据流规范的预定义参数来接收或拒绝数据流

调度

- 来自不同数据流的分组到达交换机或路由器，并由它进行处理；
- 好的调度会以公平合理的方式来对待不同的数据流；
- 已设计了多种调度技术用来改进服务质量，常用的三种技术：FIFO队列、优先权队列和加权公平队列。

图24.16 FIFO队列

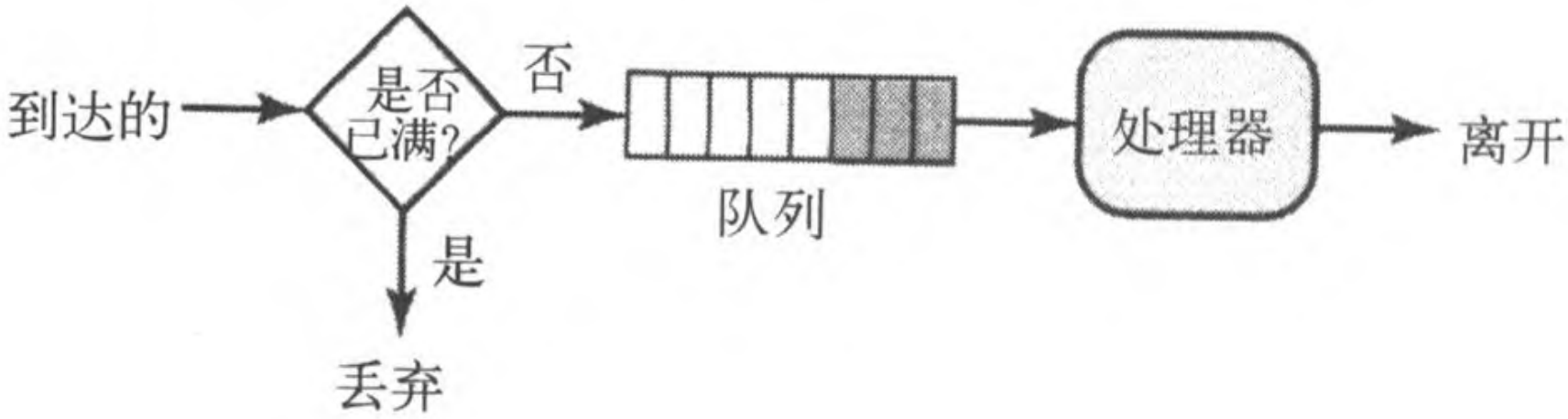


图24.17 优先级队列

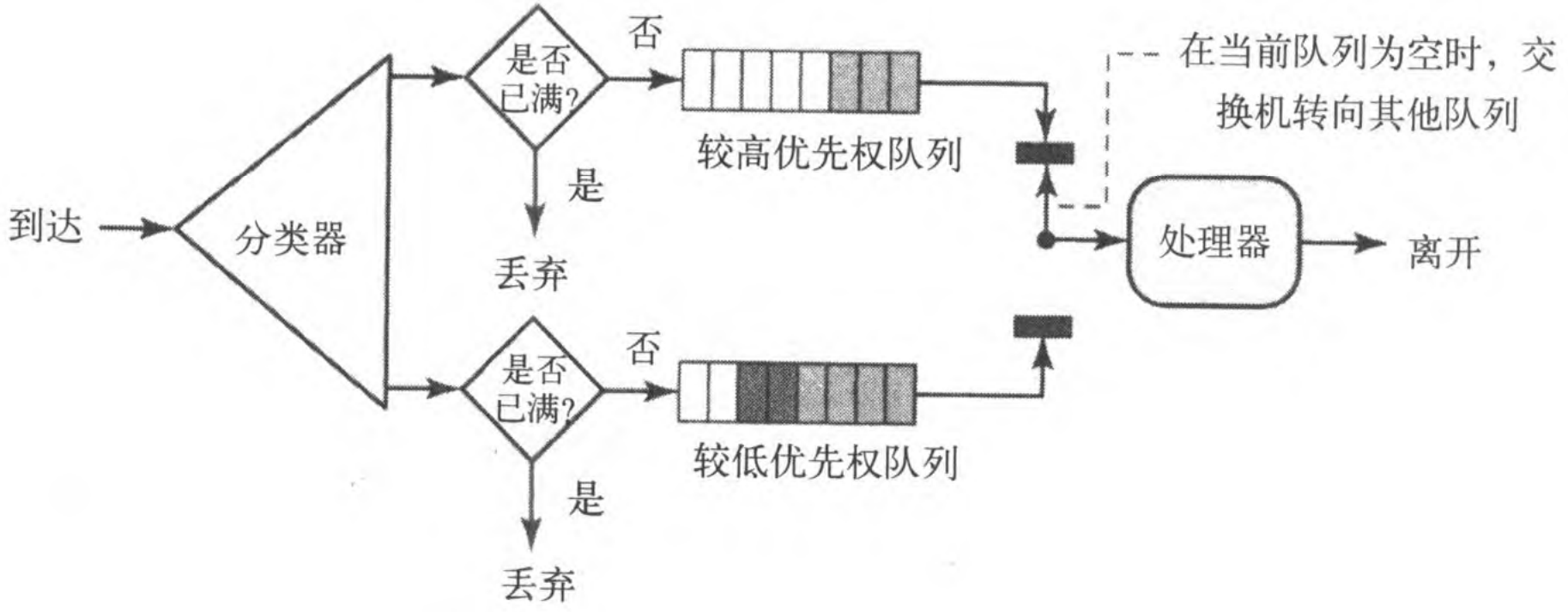
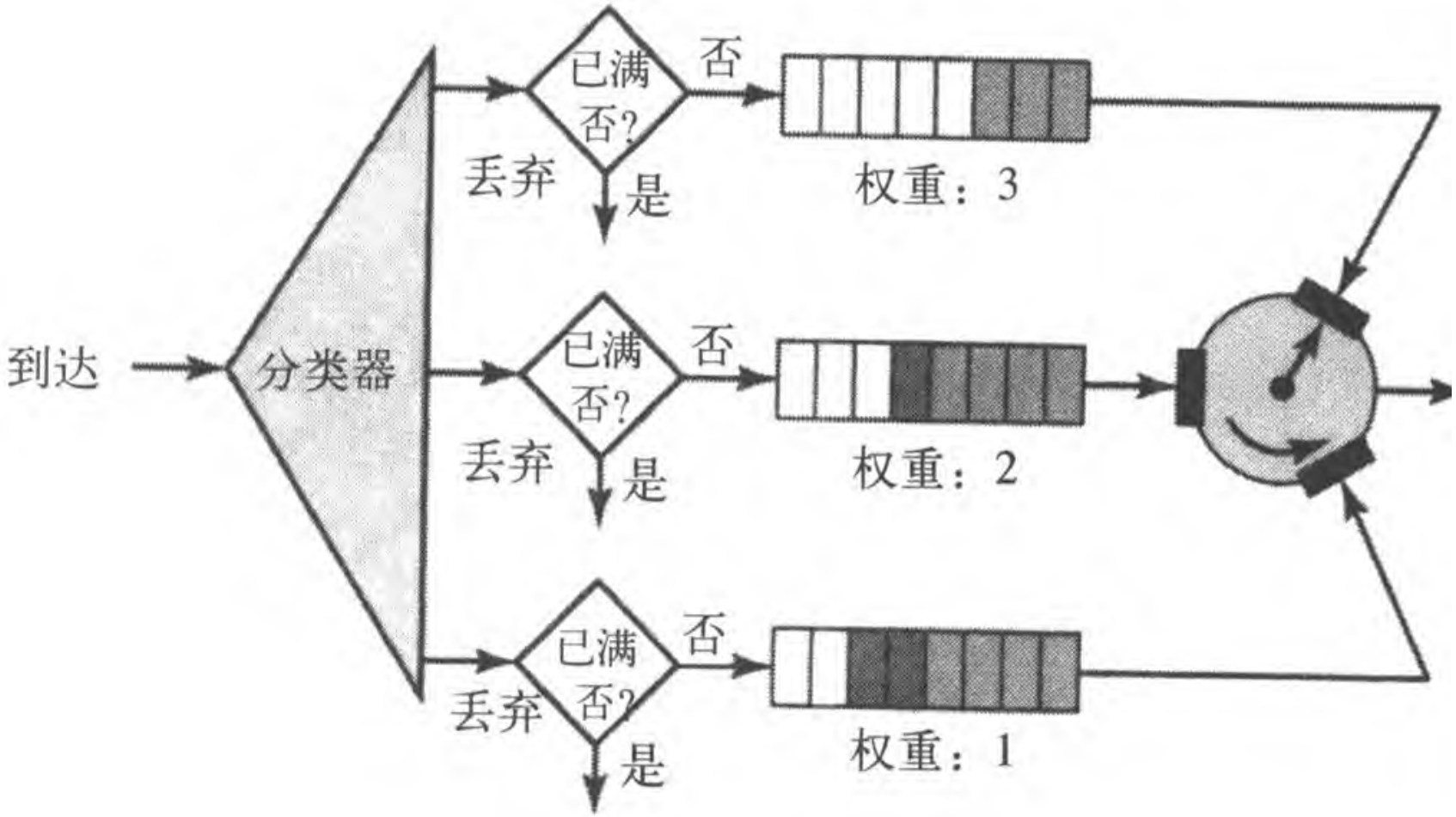

到达
分类器
是否
已满?
否
是
丢弃
较高优先权队列
是否
已满?
否
是
丢弃
较低优先权队列
处理器
离开
在当前队列为空时，交
换机转向其他队列

图24.18 加权公平队列

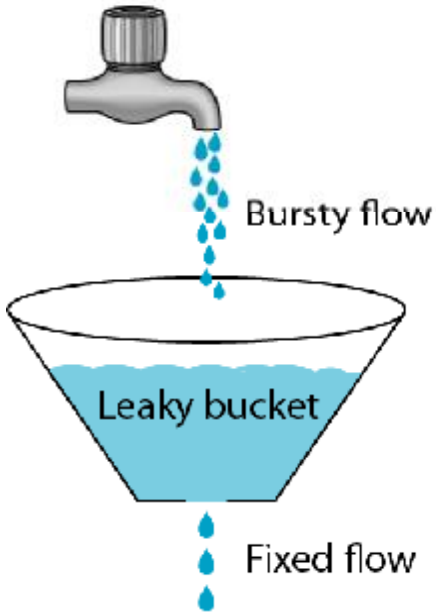


通信量整形

- 通信量整形是一种控制发送到网络中的通信量和速率的机制；
- 通信量整形有两种技术：漏桶和令牌桶。

图24.19 漏桶

- 水漏的速率并不依赖于将水倒入桶中的速率，输入速率可以发生变化，但是输出速率保持恒定；
- 能消除突发性通信量，将突发性大块数据存储到桶中，然后以平均速率发送出去



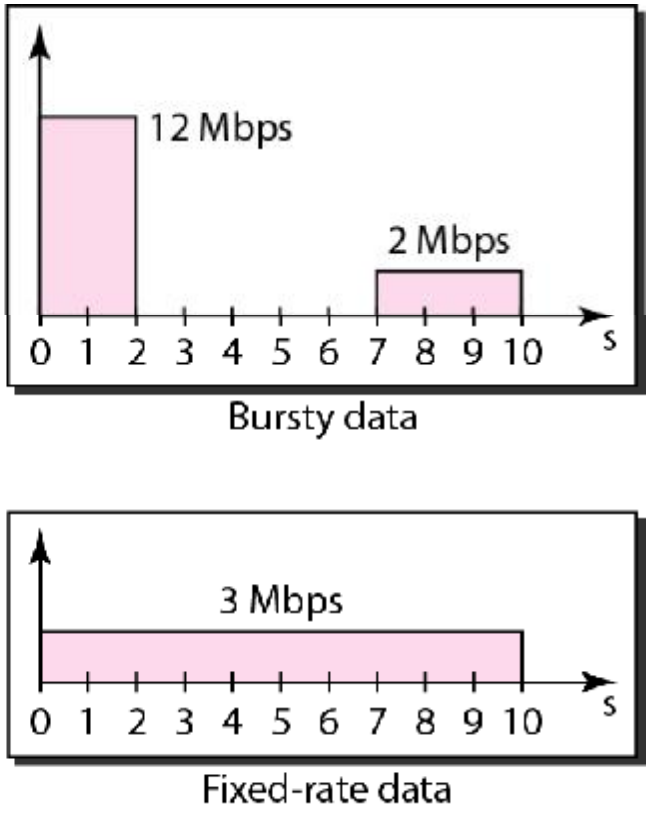
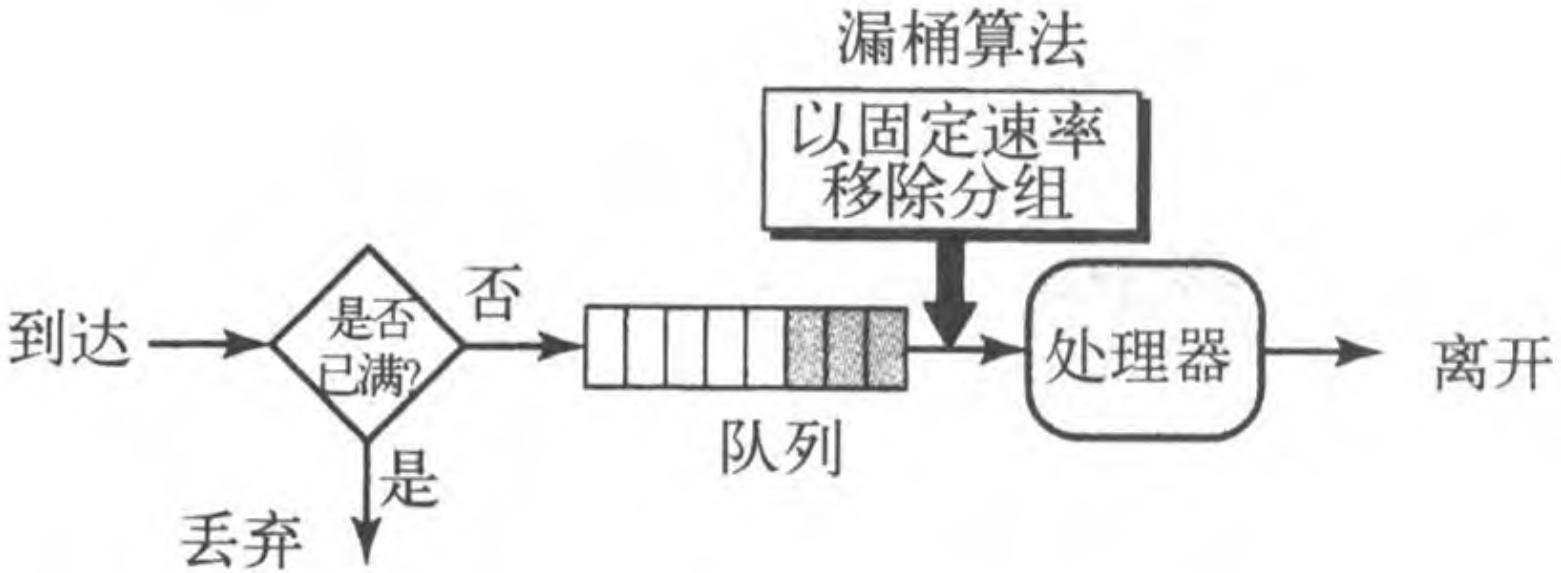


图24.20 漏桶的实现过程

p 如果通信量由固定大小的分组组成，那么在单位时间内，进程从队列中移除固定数量的分组；

p 如果通信量是由变长分组组成的，那么固定输出速率必须是基于字节个数或位个数。

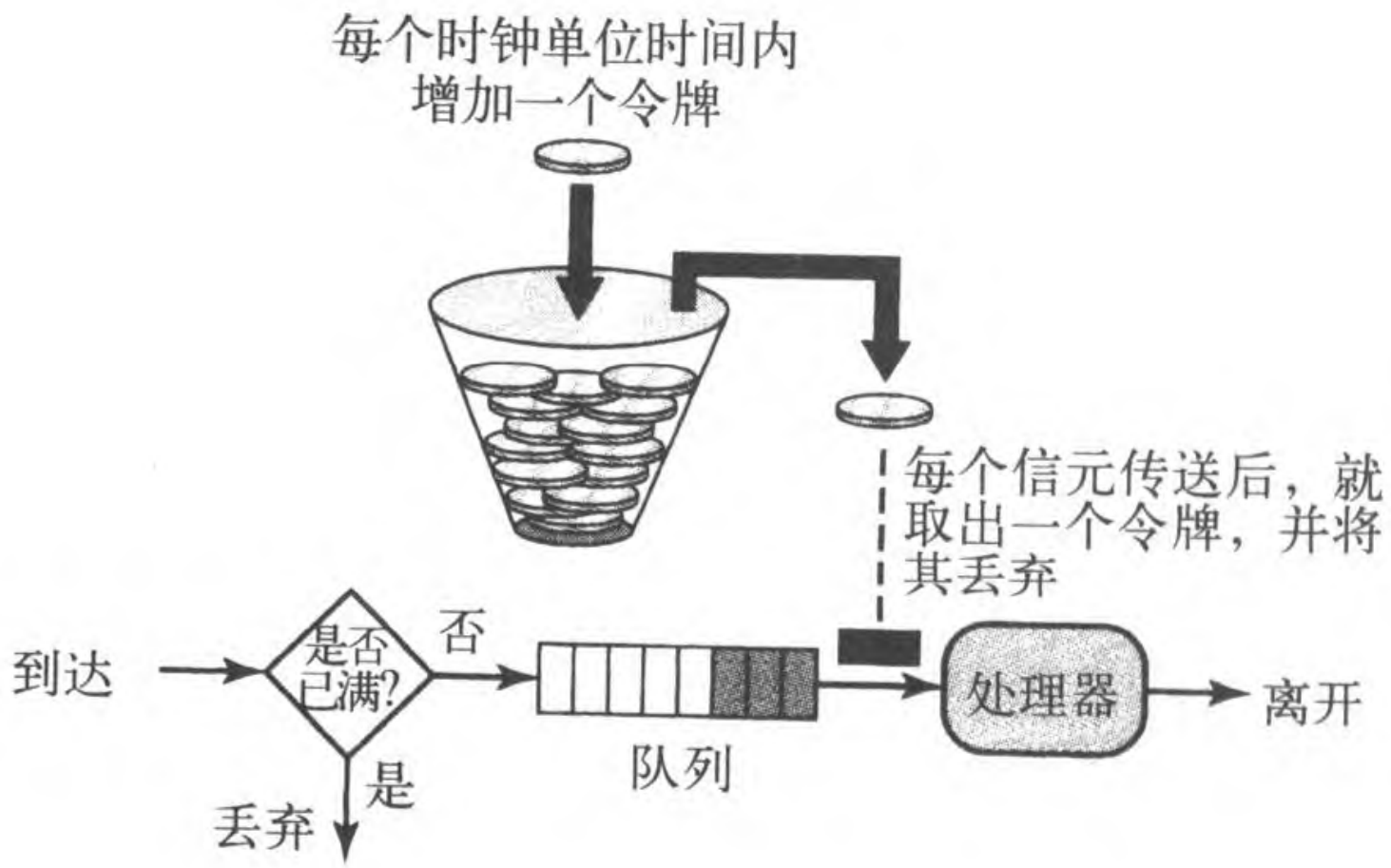


漏桶算法通过均分数据速率将突发性通信量整形为固定速率的通信量。如果桶已满，就可能丢失分组。

令牌桶

- 漏桶算法不能给空闲的主机提供信用，而即使主机中有高突发性数据，漏桶也只能允许平均速率的数据，并没有考虑主机空闲时间；
- 令牌桶（token bucket）算法允许空闲主机以令牌的形式为未来累积信用；
- 在每个时钟单位时间内，系统发送n个令牌到桶中；在每个数据信元（或字节）发送后，系统就从中去掉一个令牌；
- 只要令牌桶不是空的，主机就能发送突发性数据；
- 使用计数器可以很容易地实现令牌桶算法；
- 令牌桶允许以规定的最大速率发送突发性通信量

图24.21 令牌桶算法的原理



练习题（小学生水平）

p在一个6Mbps的网络上，有一台主机通过一个令牌桶进行流量调整。令牌桶的填充速率为1Mbps，初始时候它被填充到8Mb的容量，请问计算机以6Mbps的全速率可以传输多长时间？

解：

使用公式$S=C/(M-P)$，这里的S表示以秒计量的突发时间长度，M表示以每秒字节计量的最大输出速率，C表示以字节计的桶的容量，P表示以每秒字节计量的令牌到达速率。则：

$$S=(8\times10^6/8)/(6\times10^6/8-1\times10^6/8)=1.6\text{秒}$$

因此，计算机可以用完全速率6Mbps发送1.6秒的时间。

24-7 综合业务

p因特网已设计出两种用于提供服务质量的模型：综合业务和差分业务；

p两个模型都强调在网络层使用服务质量，尽管这些模型也可以用于诸如数据链路层的其他层；

pIP最初为尽力传递而设计，意味着每个用户接受同样级别的服务，这种类型的传递不能保证一项业务的最低要求，例如实时音频和视频应用所需的带宽；

p即使这样的应用偶然获得了额外的带宽，也可能对其他应用造成不利影响，并导致拥塞现象发生；

p综合业务（integrated service）有时称为IntServ，它是基于数据流的QoS模型，这意味着用户需要创建一条从源端到目的端的流，即一种虚电路，并将资源需求通知给所有的路由器。

综合业务是设计用于IP的基于数据流的QoS模型。

综合业务IntServ

- 如何在一种无连接的协议上实现一个基于数据流的模型呢？解决办法是在IP上使用信令协议，它为资源预留提供信令机制，这个协议称为资源预留协议（Resource Reservation Protocol，RSVP）；
- 当源端做资源预留时，需要定义数据流规范；数据流规范包括两个部分：Rspec（资源规范）定义数据流需要预留的资源（如缓冲区、带宽等），Tspec（通信量规范）定义数据流的通信量特征；
- 综合业务已经定义了两种类型的业务：保证型业务和受控载荷型业务。

RSVP

- 资源预留协议（RSVP）是一种信令协议，它帮助IP创建一条数据流通路，从而实现资源预留；
- RSVP是一种用于多播的信令系统，但也能用于单播（作为只有一个成员的多播特例）；
- RSVP中由接收方（而不是发送方，包括中间的网络设备）建立预留；
- RSVP有多种类型的报文，主要讨论两种：路径（Path）和预留（Resv）

图24.22 路径报文

- 预留资源需要知道路径，为此RSVP使用路径报文；
- 路径报文从发送方传送出来，到达多播路径中的所有接收方；
- 在整个路径中，路径报文为接收方存储必要的信息；
- 在多播环境中将路径报文发送出去，并且在路径分叉时会产生新的报文

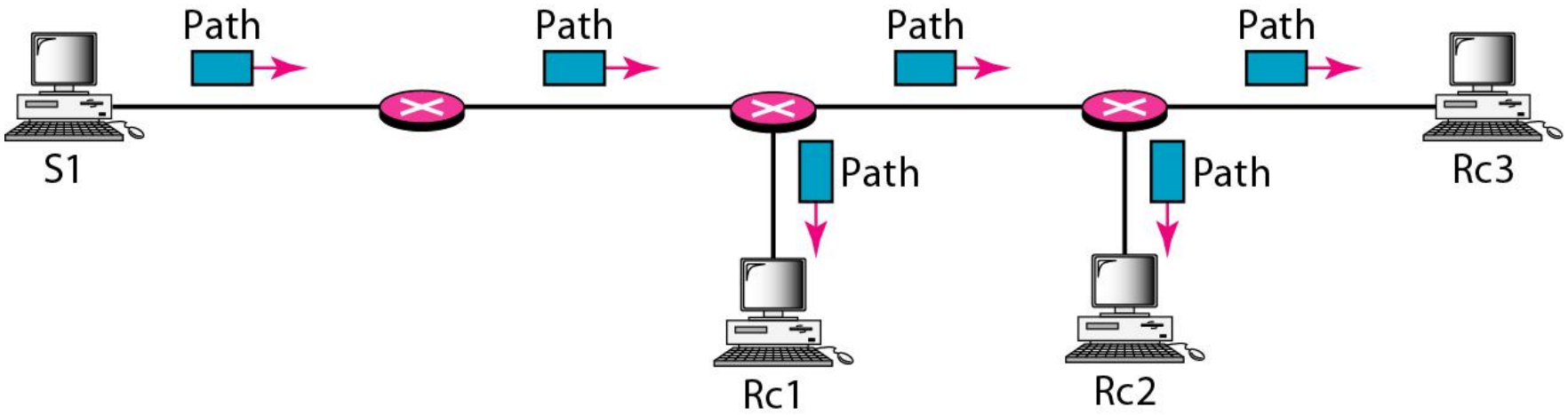


图24.23 预留报文

- 接收方收到路径报文后，发送预留报文；
- 预留报文朝着发送方传送（上行），并且在支持**RSVP**的路由器上预留资源；
- 如果路由器不支持路径中的**RSVP**，它将使用尽力传递方法对分组进行路由

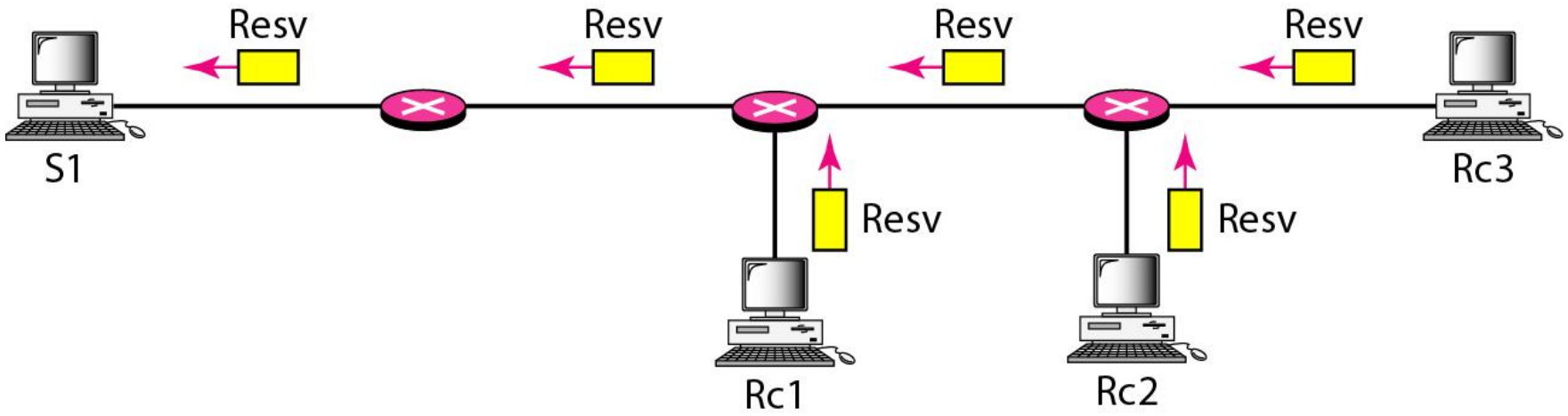


图24.24 预留合并

- RSVP并没有为数据流中的每个接收方预留资源，而是将资源预留进行了合并；
- 图中，R3将两个请求合并，只对2Mbps带宽的那个请求（即两个中较大的那个请求）做预留，因为2Mbps的输入预留就能处理这两个请求。

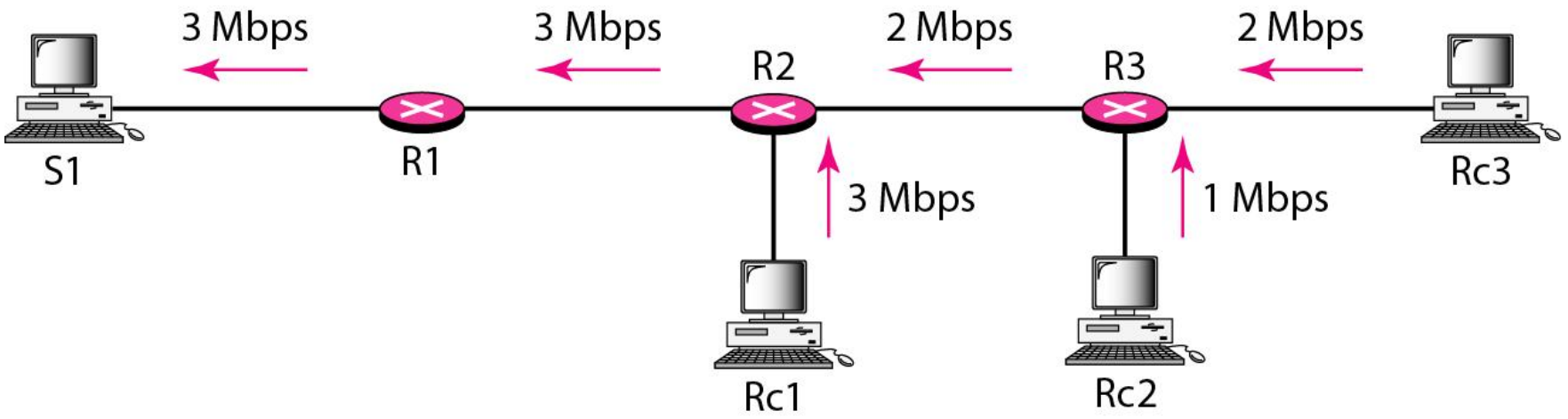
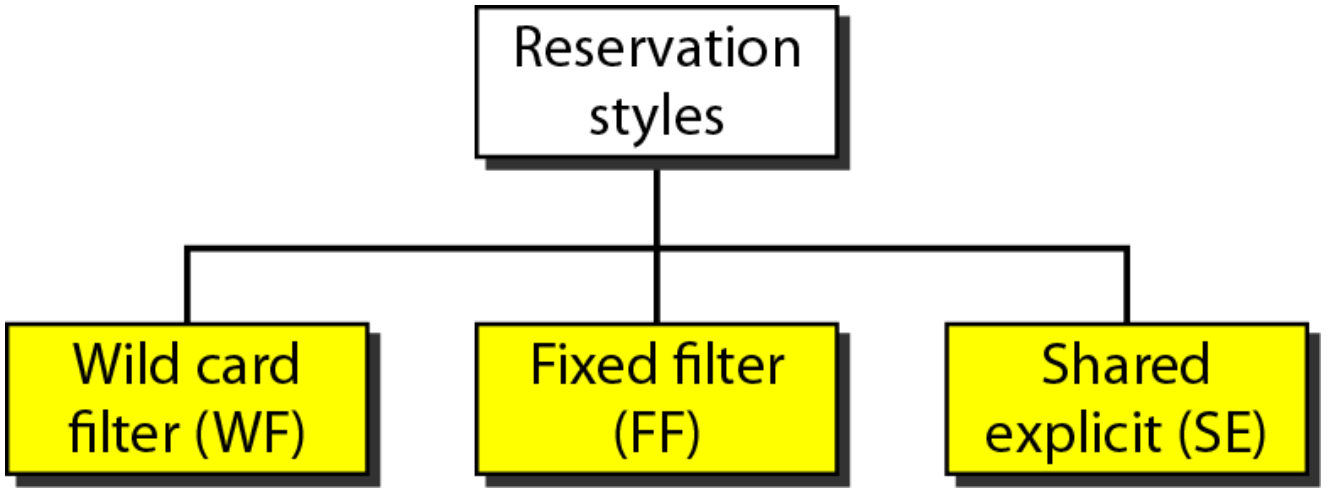


图24.25 预留方式

p通配过滤器：路由器为所有发送方创建一个预留（基于最大的请求），用于来自不同发送方数据流不同时出现的情况；

p固定过滤器：路由器为每个数据流创建一个不同的预留，用于来自不同发送方的数据流同时出现的概率很高的情况；

p共享显式方式：路由器只创建一个预留，该预留可以被一组数据流共享。



综合业务存在的问题

- 可伸缩性：综合业务模型要求每台路由器都要为每个数据流保存信息，但随着因特网规模的日益增长，这是一个严重的问题；
- 业务类型限制：综合业务只提供了两种类型的业务，即保证型业务和受控载荷型业务，可能需要比这两种业务类型更多的业务类型。

24-8 差分业务

- 差分业务（Differentiated Services，DS或Diffserv）是由因特网工程任务组IETF提出的用于弥补综合业务缺陷的一种业务；
- 与综合业务相比，有两个基本变化：
  - 主要处理过程从网络核心转移到网络边缘，解决了可伸缩性的问题；路由器不需要存储有关数据流的信息，每次发送分组时，应用程序或主机定义它们所需要的业务类型；
  - 将针对每条数据流的业务改变为针对每种类型的业务，路由器根据在分组中定义的业务类型，而不是根据数据流来发送分组，解决了业务类型限制的问题，可根据应用程序的需要定义不同的业务类型。

差分业务是用于IP的基于类的QoS模型。

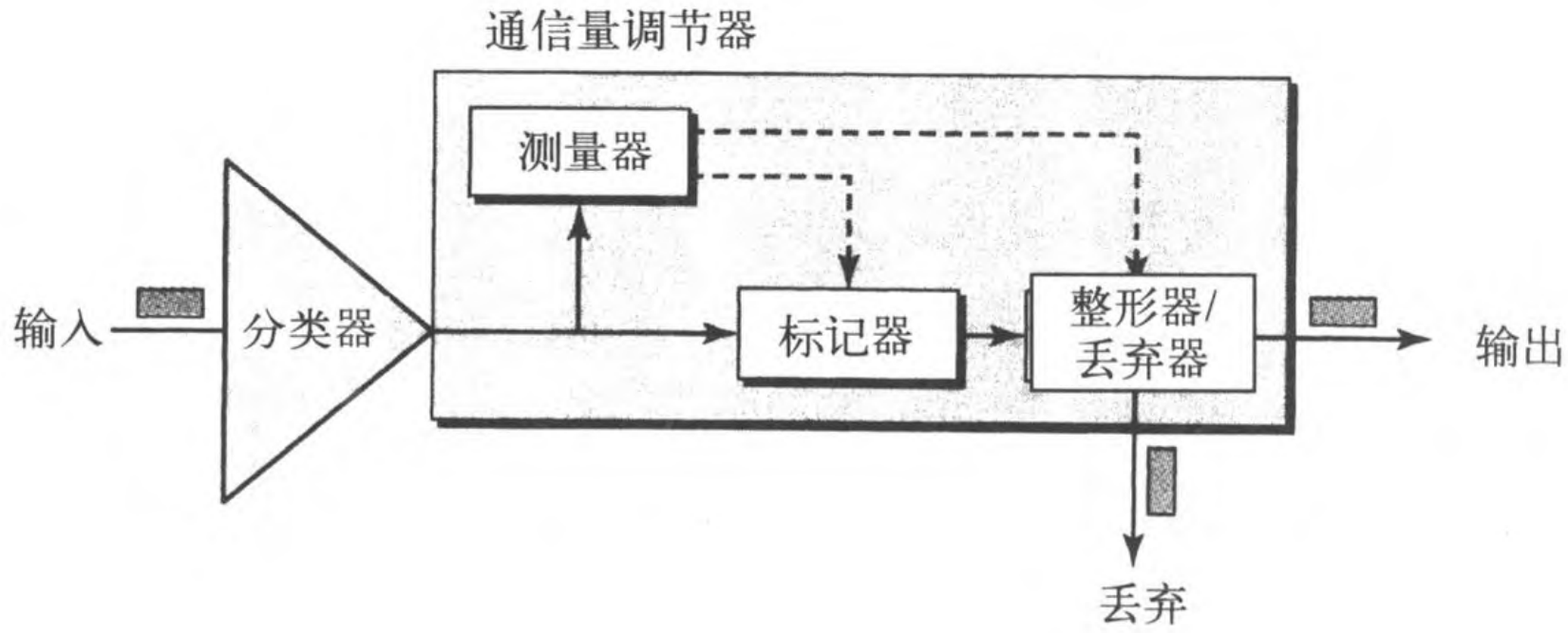
图24.26 差分业务DS字段

- 每个分组都包含一个称为DS的字段，字段的值在网络边界由主机或用作边界路由器的第一台路由器来设置；
- IETF提议用DS字段替换现存的IPv4中的TOS（服务类型）字段或IPV6中的类字段；
- 差分业务编码点DSCP是一个6位的子字段（另两位CU未使用），它定义了逐跳行为（per-hop behavior，PHB）；
- 目前已定义了三种PHB：默认（DE）PHB、加速转发（EF）PHB和保证转发（AF）PHB。

通信量调节器

- 为了实现Diffserv，DS节点使用了几种调节器，如测量器、标记器、整形器和丢弃器；
- 测量器：检查输入数据流和协商的通信量规范是否匹配，并将这个结果发送给其他组件；
- 标记器：重新标记正使用尽力传递的分组或依据接收自测量器的信息给分组作降标记；如果数据流与其对应的规范不匹配，就会对其作降标记（降低数据流的等级）；
- 整形器：当通信量与协商的规范不匹配时，整形器使用接收自测量器的信息对该通信量进行重新整形；
- 丢弃器：如果数据流严重违反协商的规范，那么丢弃器就要丢弃这些分组。

图24.27 通信量调节器



24-9 交换网络中的QoS（了解）

- 前面介绍了IP协议中提出的QoS模型；
- 两个交换网络中的QoS：帧中继和ATM；这两个网络都是虚电路网络，它们需要一个诸如RSVP的信令协议。