

第五章 网络层

流量整形



流量整形

- 调节数据传输的平均速率（和突发数据流）
- 算法
 - 漏桶（leaky bucket）
 - 令牌桶（token bucket）
 - 其它：资源预留、准入控制、分组调度等
- 可以减少拥塞



漏桶算法

算法描述

- ▼ 每个主机连接到网络的接口中都有一个漏桶，即一个优先长度的内部队列
- ▼ 当桶中有分组的时候，输出速率是恒定的，当桶空的时候，输出速率是0
- ▼ 当一个分组到达满的桶的时候，分组将被丢弃（满则溢）
- ▼ 每个时钟嘀嗒(**tick**)，仅允许一个分组或固定数量的分组发送出去



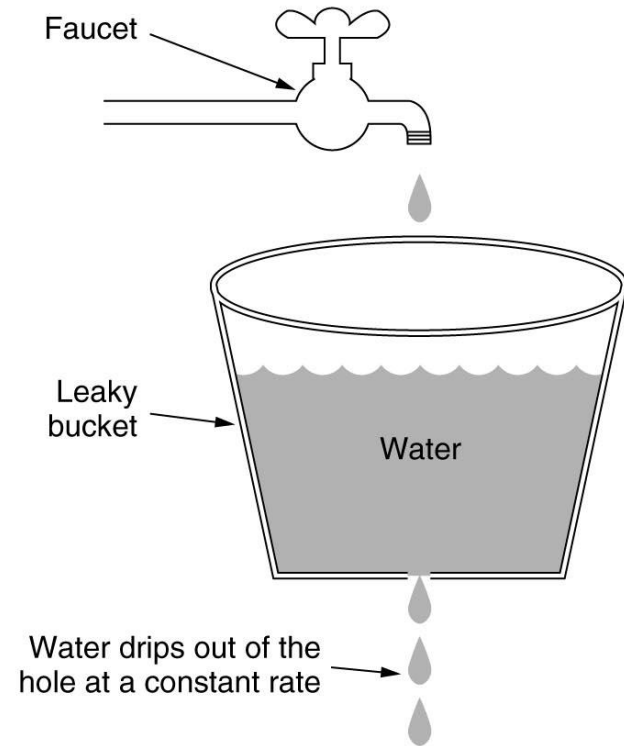
漏桶算法

算法效果

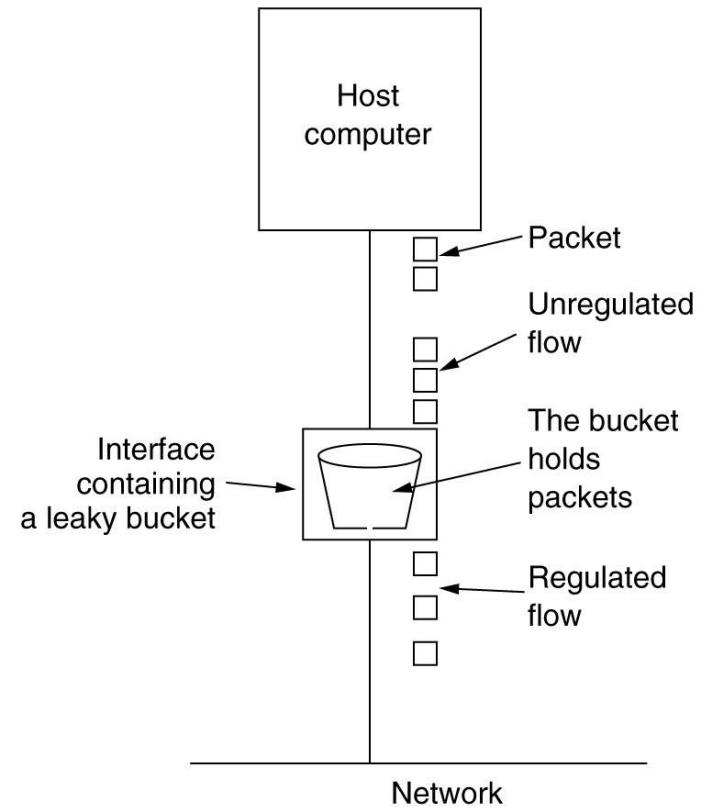
主机内用户进程产生的分组流往往是一个不稳定的流，漏桶可以让它输出到网络时变成一个稳定流，抹平了突发尖峰，极大地减少了发生拥塞的机会。



漏桶算法



(a)

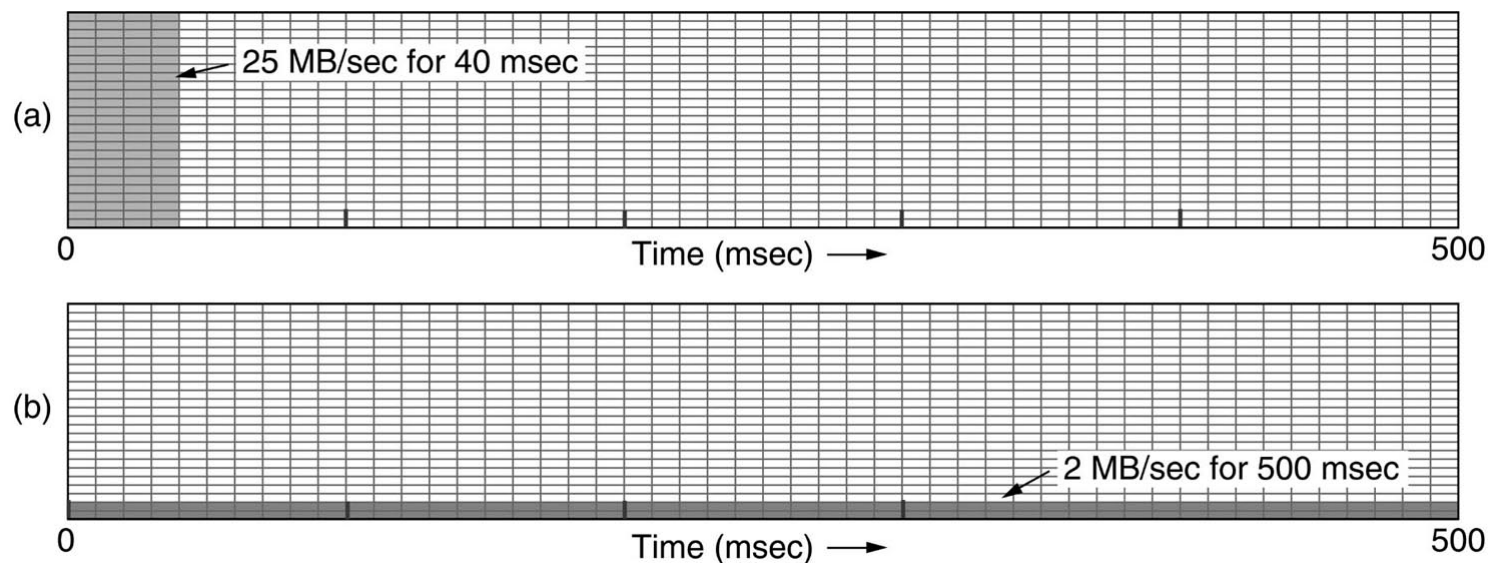


(b)



例

- 数据产生速率：25MBps
- 路由器的工作速率：2MBps
- 数据突发时间持续：40ms
- 漏桶输出速率：2MBps
- 漏桶容量：1MB





漏桶的缺点

当漏桶满了之后，数据将被丢弃

不能大量地突发数据



改进

令牌桶



令牌桶算法

- 当大量数据突发的时候，令牌桶算法允许输出加快到某种程度
- 令牌桶拥有令牌（tokens），且以每 ΔT 秒产生一个令牌的速度往桶中输入令牌
- 一个分组要发送的时候，它必要从桶中取出和获取到一个令牌
- 令牌桶算法允许累积令牌，但最多可以累积 n （令牌桶的容量）个令牌



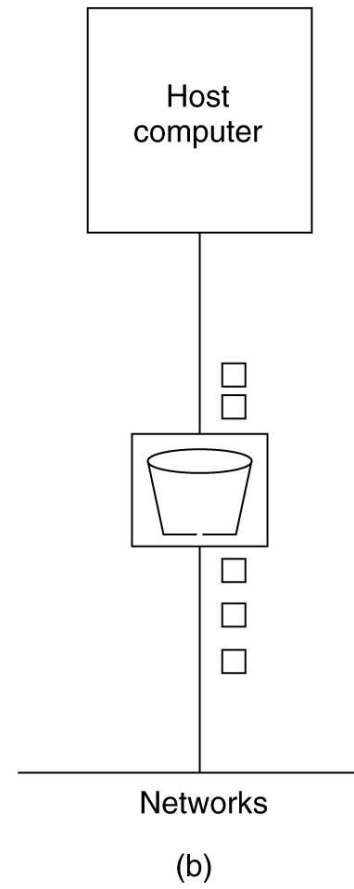
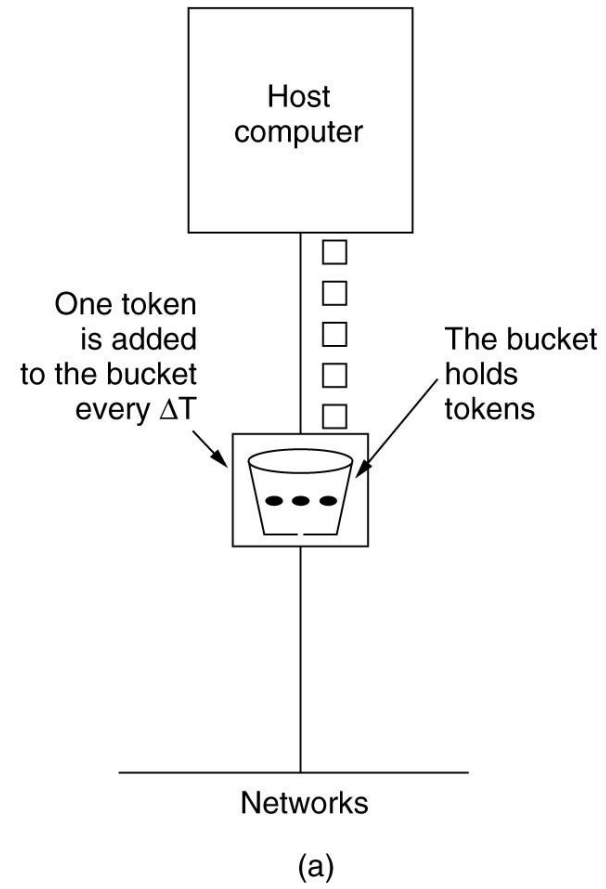
令牌桶算法

和漏桶算法相比：

- 令牌桶允许突发，但是最大突发受制于令牌桶容量的限制
- 当桶满的时候，令牌桶算法丢掉的是令牌（不是分组）



令牌桶算法





令牌桶算法

计算最大突发时间

- 突发时间: S 秒
- 令牌桶容量: B 字节
- 令牌到达的速率: R 字节/秒
- 最大输出速率: M 字节/秒

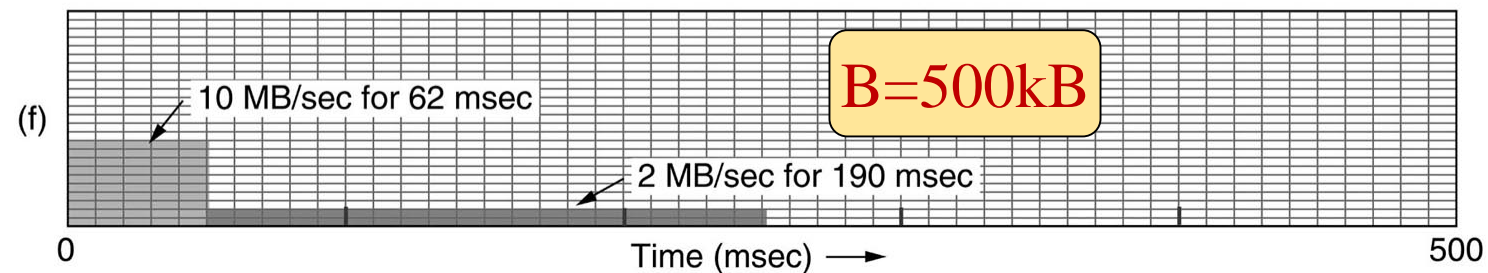
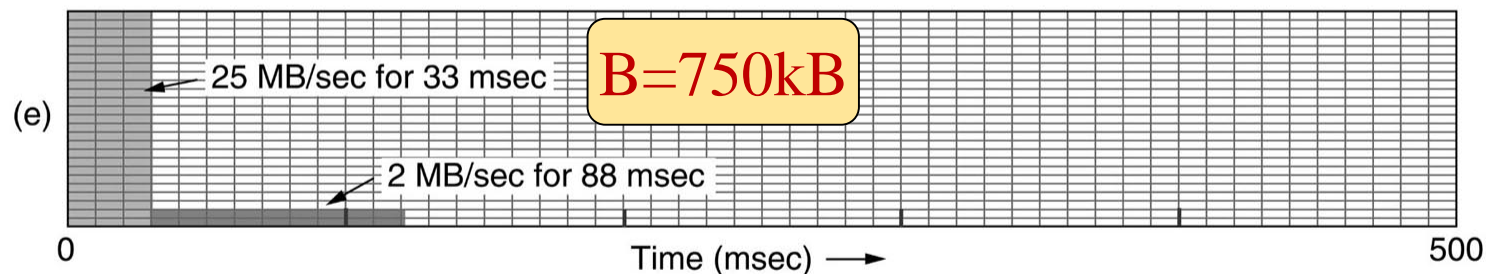
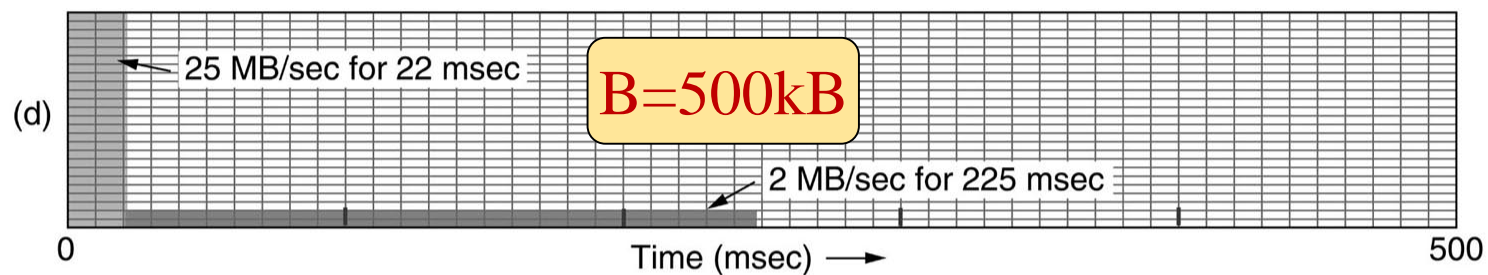
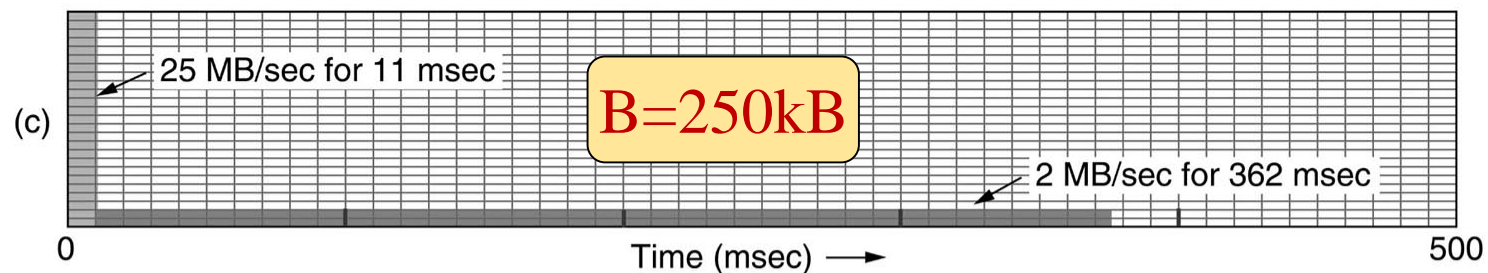


$$B + RS = MS$$

$$S = B / (M - R)$$

M=25MB, R=2MBps, 当 1M 突发数据到达时

$$(1\text{MB} - 25\text{MB/s} \times 0.011\text{s}) \div 2\text{MB/s} = 362\text{ms}$$



课堂练习

一个10Mb/s的网上有一台由令牌桶控制的主机，令牌桶以2Mb/s的速率填充。假定令牌桶有20Mb的容积。

- (1) 问主机能以10Mb/s全速发送数据的最大可能的时间是多少？
- (2) 主机能以10Mb/s全速发送数据的最大数据量是多少？

参考答案

□ 已知：

令牌桶容量： $B = 20\text{Mb}$

令牌产生速率： $R = 2\text{Mb/s}$

最大突发速率： $M = 10\text{ Mb/s}$

设突发时间为 s ，则：

$$s = B / (M - R) = 20 / (10 - 2) = 2.5 \text{ (s)}$$

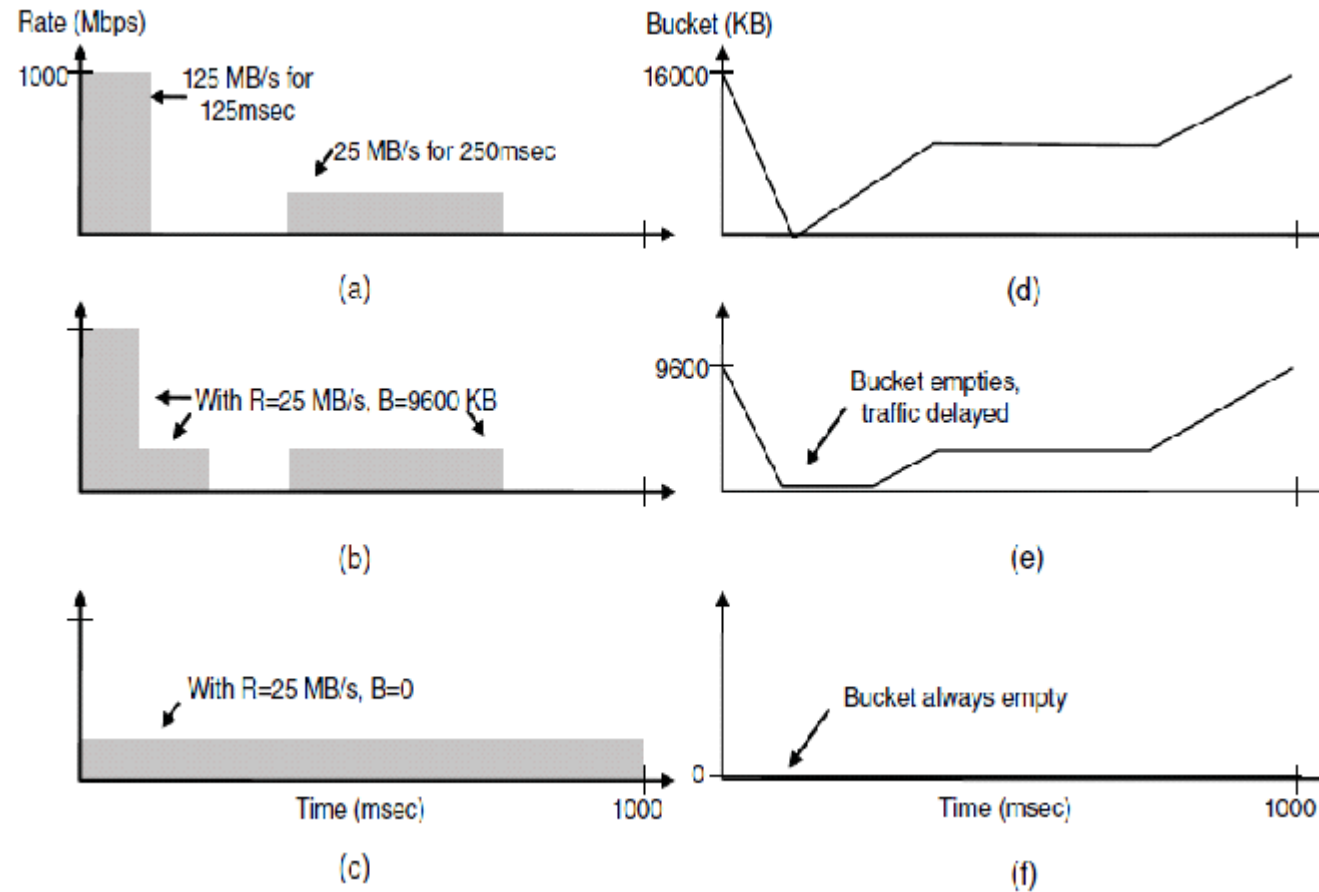
□ 设最大突发数据量是 V_{\max} ，则：

$$V_{\max} = s \times M = 2.5(\text{s}) \times 10\text{Mb/s} = 25\text{M (bits)}$$

Host traffic
 $R=200$ Mbps
 $B=16000$ KB

Shaped by
 $R=200$ Mbps
 $B=9600$ KB

Shaped by
 $R=200$ Mbps
 $B=0$ KB





小结

- 漏桶算法可让一个不稳定的流整为一个稳定的流
 - 无法突发
 - 桶满时丢掉的是分组
- 令牌桶可以累计突发数据的令牌，允许有上限的突发
 - 桶满时丢掉的是令牌
- 通过整形算法，可以减少拥塞可能，提高服务质量QoS

思考题

- 什么流量整形？
- 漏桶的工作原理是怎样的？
- 令牌桶的工作原理是怎样的？
- 试比较漏桶和令牌桶两种整形算法。

谢谢观看

致谢

本课程课件中的部分素材来自于：（1）清华大学出版社出版的翻译教材《计算机网络》（原著作者：Andrew S. Tanenbaum, David J. Wetherall）；（2）思科网络技术学院教程；（3）网络上搜到的其他资料。在此，对清华大学出版社、思科网络技术学院、人民邮电出版社、以及其它提供本课程引用资料的个人表示衷心的感谢！

对于本课程引用的素材，仅用于课程学习，如有任何问题，请与我们联系！