

# 总结、分析与解惑！

## ◆ 基本目的：

- 考查学生学习和掌握《现代计算机网络》之知识（理论与技术）及其运用的程度
- 程度：新度、广度、深度、活度

## ◆ 考题形式：

- ❖ 一：术语解释（共10分，每小题1分） 新度（用中文说明或写出英文原文均可）
- ❖ 二：单选填空（共20分，每小题2分） 广度
- ❖ 三：分析计算（共30分） 深度
- ❖ 四：综合求解（共40分） 活（合）度

## ◆ 基本范围：

- 以授课PPT内容为准，L. Peterson为主要参考书

## ◆ 基本策略：

- 对投机者说 “不！”

# 课程结构—2006版

## ◆ Ch1: 网络体系结构-4H

- ◆ 基本概念
- ◆ 基本技术
- ◆ 基本网络

## ◆ Ch2: 互连与路由-4H

- ◆ TCP/IP/IPv6
- ◆ RIP/OSPF/BGP4

## ◆ Ch3: 端到端与组播-6H

- ◆ 端到端协议与数据
- ◆ 组播路由算法与协议

## ◆ Ch4: QOS与拥塞控制-6H

- ◆ 集成服务与区分服务
- ◆ 拥塞控制
- ◆ 主动队列管理AQM

## ◆ Ch5: 网络安全-4H

- ◆ 加密算法与安全机制
- ◆ 安全威胁与安全工具

## ◆ Ch6: P2P原理与技术-6H

- 关键技术
- 实例分析

## ◆ Others:-2H

# 例题分析-01

- ◆如果波长等于1um,那么在0.1um频道中可以有多大的带宽？

$$f = c / \lambda ; \quad df/d\lambda = -c / \lambda^2$$

$$df = -c / \lambda^2 d\lambda ; \quad \lambda = c / f$$

$$c = 3 \times 10^8 \text{ m/s} ; \quad \lambda = 10^{-6} \text{ m}$$

$$\Delta\lambda = 0.1 \mu\text{m} = 0.1 \times 10^{-6} \text{ m} = 10^{-7} \text{ m}$$

$$f = 3 \times 10^8 / (10^{-6})^2 \times 10^{-7} = 30,000 \text{ GHz}$$

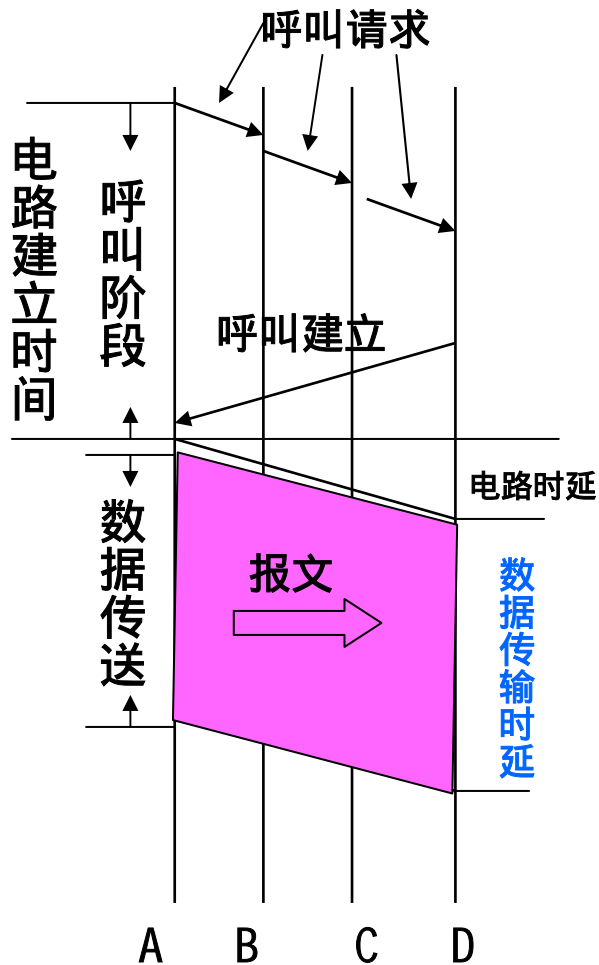
# 例题分析-02

- ◆ 假定PSTN的带宽为3000Hz，典型的信噪功率比是20dB，试确定可以取得的理论上最大的数据传输率？
  - $\text{SNR} = 10 \lg(S/N)$  : SNR –Signal Noise Rate
  - 因此， $20 = 10 \lg(S/N)$
  - $S/N = 100$
  - 现在， $C = W \log_2(1+S/N) = 3000\text{Hz} \times \log_2(1+100)$   
 $= 19936\text{bps}$
  - 所以理论上可以取得最大的数据传输速率是  
 $19936\text{bps} = 19.936\text{Kbps}$

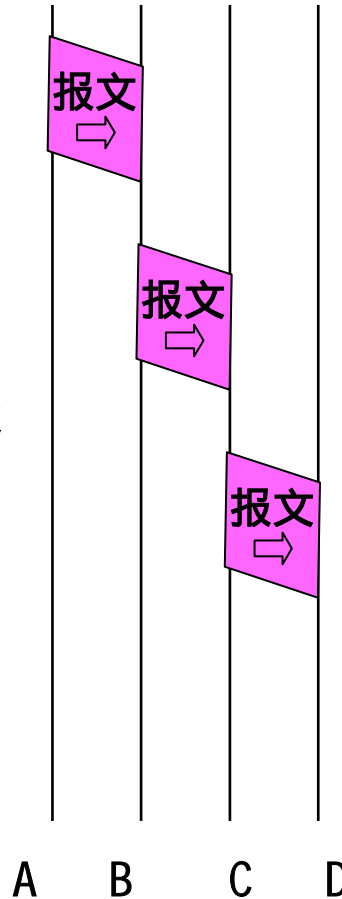
# 例题分析-04(1/4)

- ◆ 试在下列条件下比较电路交换和分组交换。设要传送的报文件为 $x$  ( bit )  
，从源站到目的站共经过 $k$ 段链路，  
每段链路的传播时延为 $d(s)$ ，数据率为 $b(b/s)$ ；电路交换的建立时间为 $s(s)$   
，分组交换时分组长长度为 $p(bit)$ ，且各结点的排队等待时间可忽略不计。问在怎样的条件下，分组交换的时延比电路交换要小？

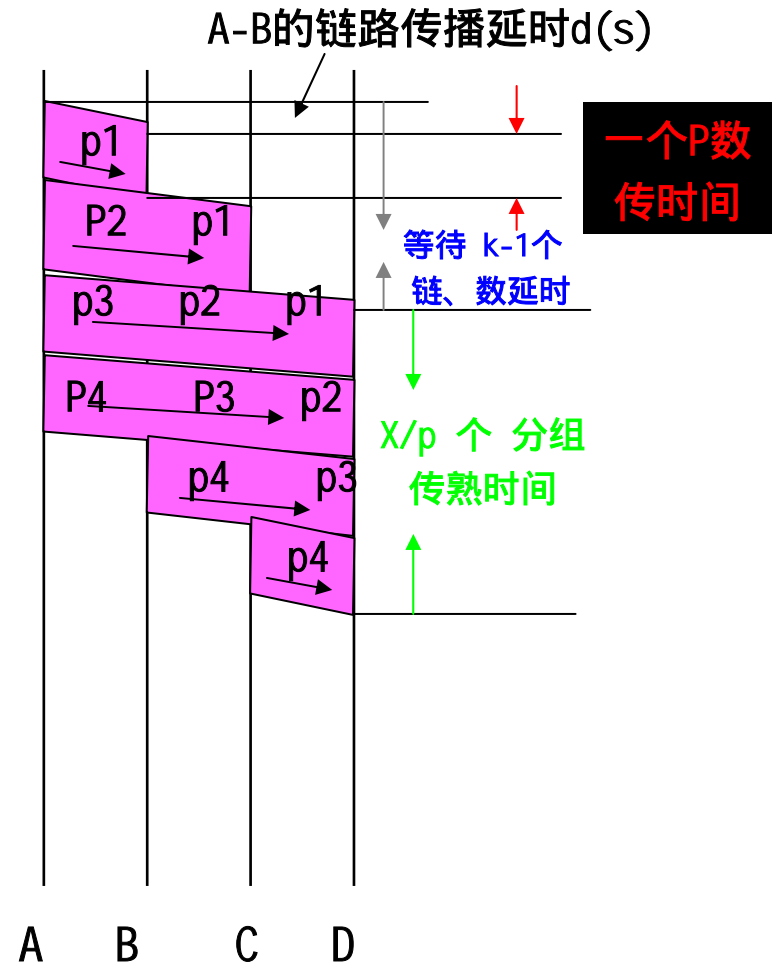
# 例题分析-04(2/4)



电路交换



报文交换



分组交换

# 例题分析-04(3/4)

## 1 电路交换的总时间 $T_1$

- $T_1$  = 链路建立时间 +  $k$ 段链路传播延迟 +  $x$ 比特数据的传输时间，即
- $T_1 = s + k \times d + x/b$

## 2 分组交换的总时间 $T_2$

- $T_2$  = 从发送端开始发第一个比特到接收端收到第一比特的时间是中间路由器重发 $k-1$ 次，每次重发时间 $p/b$ （一个包的所有比特都收齐了，才能开始重发）+ 所有比特的发送时间 $x/b$  +  $k$ 段链路的时延。即
- $T_2 = (k-1)p/b + x/b + kd$

◆ 结论：作  $T_1 - T_2 = s - (k-1)p/b > 0$

- 即  $s > (k-1)p/b$ 成立时，包交换优于电路交换。

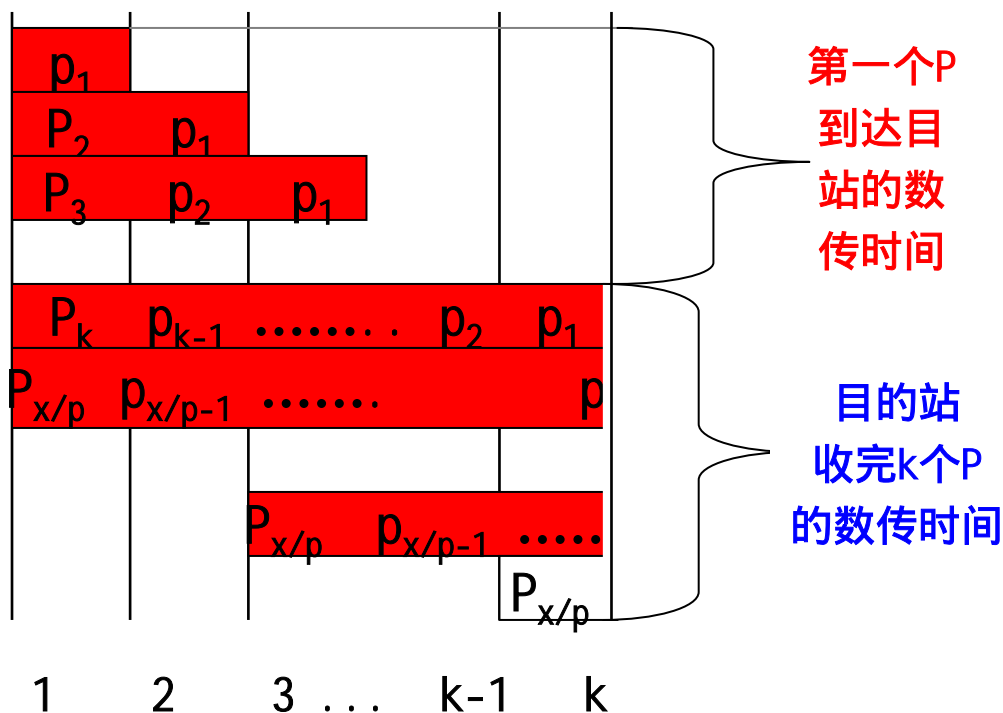
# 例题分析-05(1/2)

- ◆ 在上题的分组交换网中，若报文长度和分组长度分别为 $x$ 和 $(p+h)$  (bit)，其中 $p$ 为每个分组的数据部分的长度，而 $h$ 为每个分组所带的控制信息固定长度，与 $p$ 的大小无关。通信的两端共经过 $k$ 段链路。链路的数据率为 $b$ (b/s)，但传播时延和结点的排队时间均可忽略不计。
  - 若打算使总的时延为最小，问 $p$ 应取为多大？所谓“总的时延”是指从发送端发送第一个比特开始，到接收端收到报文的最后一个比特为止的时间。



# 例题分析-05(2/2)

- ◆ 总时延 = 第一个包经 $k$ 个站和 $k$ 条链路刚到达目站的数传时延 +  $x/p$ 个包全传到目站的数传时延
- ◆ 总数据加上报头后的传输信息量是 $(p+h)x/p$  ;源端发送完这些数据位需要时间是 $(p+h)x/(pb)$
- ◆ 中间路由器重传最后一个包所花的总时间是 $(k-1)(p+h)/b$



- $Z = (p+h)x/(pb) + (k-1)(p+h)/b$
- 令  $dZ / dp = k-1 - xh/p^2 = 0$
- ◆ 因 $p > 0$  , 所以
  - ◆  $p_{opt} = \sqrt{xh / (k - 1)}$

# 例题分析-06

- ◆ 一个完整信息（如一个数据报）被分成10帧在数据链路层传输，每一帧无损到达目的地的可能性是80%，如果数据链路层不进行差错控制，那么这一信息平均要发多少次才能完整到达接收方？
  - 每一帧的达到概率 = 0.8，整个信息到达的概率  $p = 0.8^{10} = 0.107$ .
  - 为使信息完整达到接收方，发送一次成功的概率是 $p$ ，2次成功的概率是 $(1-p)p$ ，3次成功的概率是 $(1-p)^2p$ ； $i$ 次成功的概率是 $(1-p)^{i-1}p$  [n重贝努里分布]
  - 因此平均发送次数等于
  - $E = 1 \times p + 2(1-p)p + 3(1-p)^2p + \dots + (1-p)^{i-1}p + \dots = \sum_{i=1}^{\infty} i p (1-p)^{i-1}$
  - 利用  $S = \sum_{i=1}^{\infty} a^i = a/(1-a)$ ； $S' = \sum_{i=1}^{\infty} i a^{i-1} = 1/(1-a)^2$
  - 故  $E = 1/p = 1/0.107 = 9.3$

# 几种重要的概率分布

## ◆ 贝努里分布

- 它描述**一次**贝努里实验中，成功或失败的概率。
- 其概率分布为： $P\{X=1\}=p$ ， $P\{X=0\}=1-p$
- 故也称两点分布或（0-1）分布。

## ◆ 二项分布

- 它描述n次贝努里实验中**事件A出现k次**的概率。
- $P\{X=k\}=C_n^k p^k (1-p)^{n-k}$ ， $k=0, 1, \dots, n$

## ◆ 几何分布

- 它描述在**k次贝努里实验中首次出现成功**的概率
- $P\{X=k\}=p(1-p)^{k-1}$ ， $k=1, 2, \dots$

- 重要性质--**无后效性**：在前n次实验**未出现成功**的条件下，再经过**m次实验**（即在n+m次实验中）**首次出现成功的概率**，**等于恰好需要进行m次实验出现首次成功的无条件概率**。  
用式子表达：
- $P\{X=n+m \mid X>n\}=P\{X=m\}$  （试证明之）
- **与过去历史无关的性质称为马尔可夫特性**
- 几何分布在我们下面讲的排队论中是\*\*非常重要
- 它可以描述某一任务（或顾客）的服务持续时间。

## ◆ 泊松分布（Poi sson）

- 它描述在很小时间间隔内，一个对象的到达的概率与时间间隔的长度成正比；到达间隔相互独立，具有相同的负指数分布。  $k$ 个对象的到达的概率，  $\lambda$ ：平均到达率
- $P\{X = k\} = \lambda^k e^{-\lambda} / k!$   $k=0, 1, 2, \dots$
- 在时间间隔T内到达k个报文的概率为
- $P\{T\text{秒内到达}k\text{个报文}\} = (\lambda T)^k e^{-\lambda T} / k!$

# ◆指数分布

- 它是一种连续型的概率分布，它的概率密度：
- $f(x) = e^{-x} ; x \geq 0 ; f(x) = 0 \quad x < 0$
- 它的分布函数： $F(x) = 1 - e^{-x} \quad x \geq 0$
- 指数分布的一个有用的性质是它的数学期望等于标准差：
- $\mu_X = \sigma_X = 1/\lambda$
- 在连续型随机变量中，只有**指数分布具有无后效性**。即：  
若随机变量  $X$  服从指数分布，对任意的  $s > 0, t > 0$ ，有  
 $P\{X > s+t \mid X > s\} = P\{X > t\}$
- 离散型随机变量中，只有**几何分布具有无后效性**。这两种分布可以分别用来描绘离散等待时间和连续等待时间。
- 在排队理论和随机Petri网中，指数分布是很重要的。

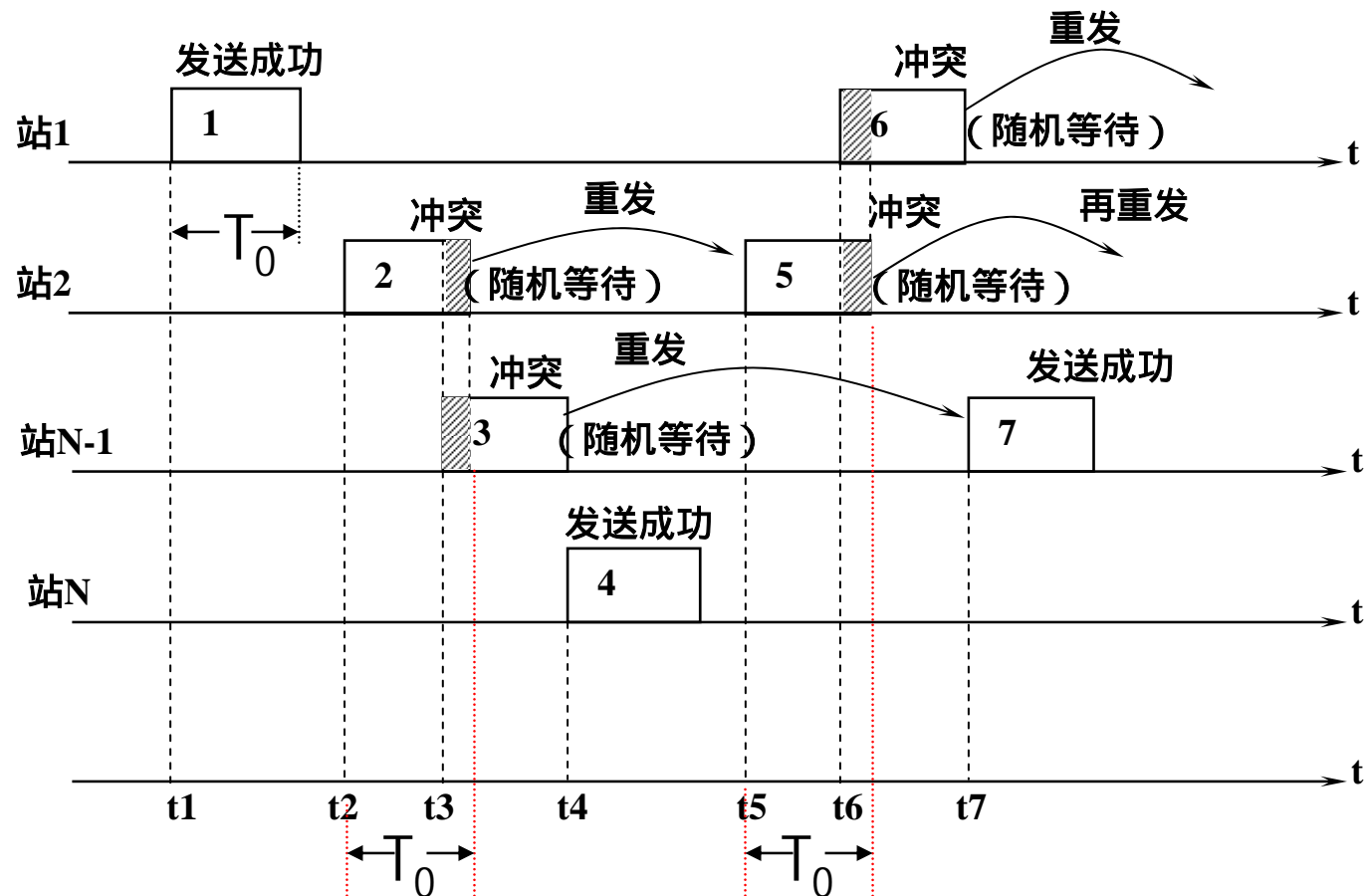
# 例题分析-07(1/5)

## ◆求纯ALOHA系统中在稳定条件下，S与G的关系？

- 设帧的到达是泊松过程，到达时间间隔随机变量的概率密度函数是： $a(t)=\lambda e^{-\lambda t}$ ， $\lambda$ 是帧的平均到达率，有 $\lambda=G/T_0$
- G：网络负载（offered load）， $T_0$ 内总共发送的平均帧数
- S：网络吞吐率（Throughput）， $T_0$ 内成功发送的平均帧数 = 信道利用率 = 效率

# 例题分析-07(2/5)

## ◆ 纯ALOHA 系统的工作原理图



# 例题分析-07(3/5)

## ◆某些结论：

- 一个帧成功发送的条件是：该帧与它前后两帧的达到时间间隔均大于 $T_0$
- 显然 $G \geq S$ ，仅在不显然  $0 \leq S \leq 1, S=1$  是极限情况即一帧接一帧地发， $S \rightarrow 1$  的程度来衡量信道的利用率
- 发生冲突时才有 $G=S$ ，另外冲突频繁时， $G$ 可远大于 $S$



# 例题分析-07(4/5)

◆  $S = G \cdot P$  [ 发送成功 ]

◆  $P$  [ 发送成功 ] =  $P$  [ 连续两间隔  $> T_0$  ]  
=  $\{ P$  [ 到达间隔  $> T_0$  ]  $\}^2$

◆  $P$  [ 到达间隔  $> T_0$  ] =  $\int_{T_0}^{\infty} a(t) dt$   
=  $\int_{T_0}^{\infty} (G / T_0) e^{-Gt/T_0} dt = e^{-G}$

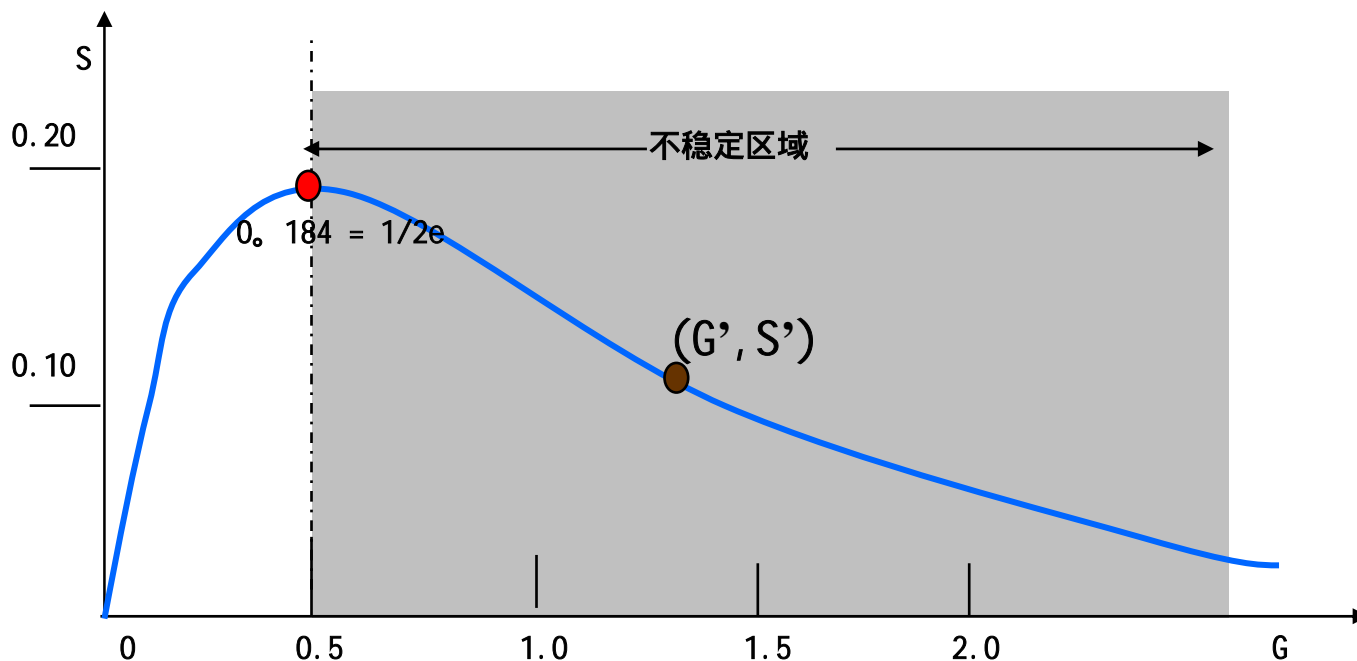
◆  $P$  [ 发送成功 ] =  $e^{-2G}$

◆  $S = G e^{-2G}$  - - Abramson 1970 年首次推出,  
且  $\text{Max} S (G=0.5) = 0.5 e^{-1} = 0.184$

◆ 纯ALOHA的负载不能超过0.5

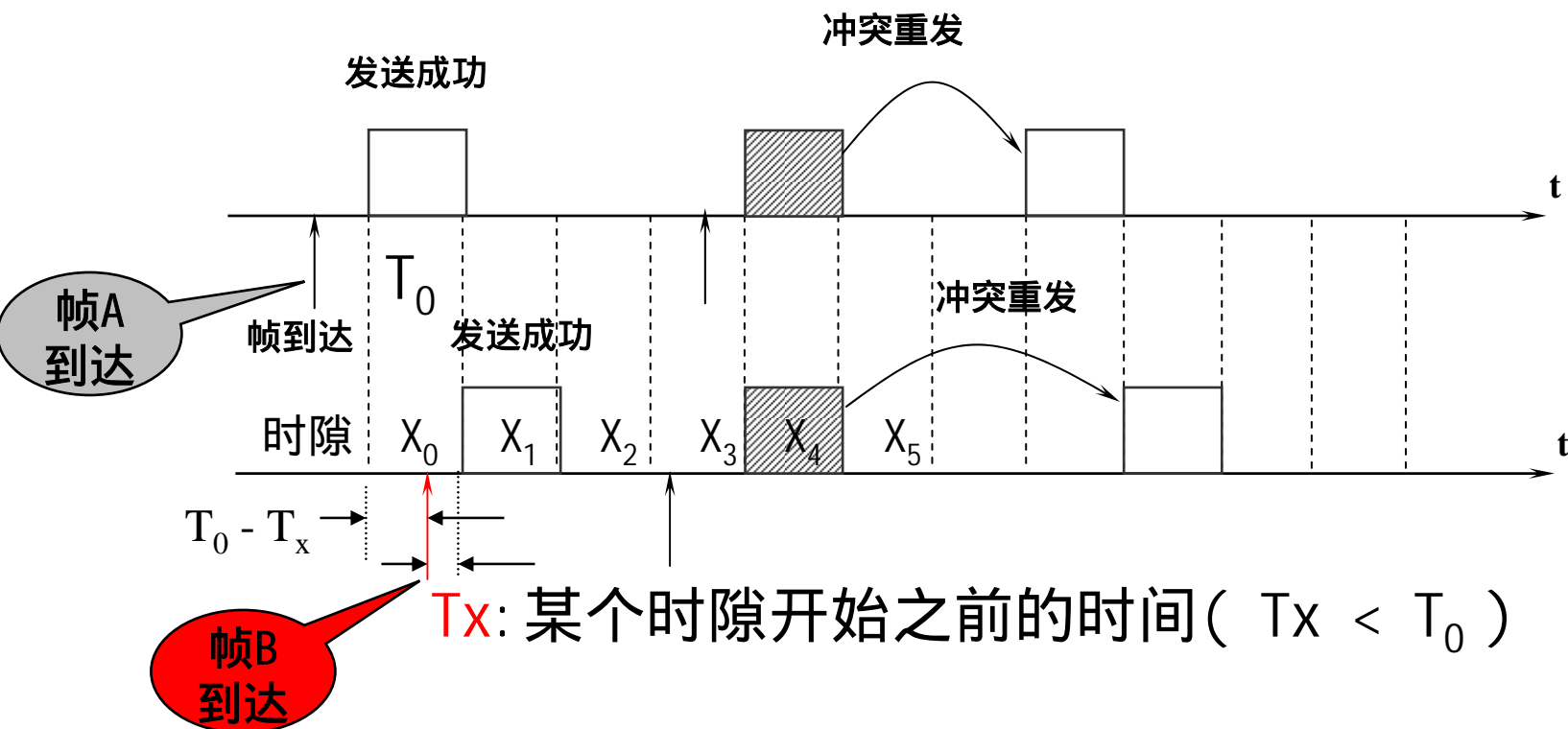
# 例题分析-07(5/5)

- ◆ 信道的最大利用率是18.4%;这不令人满意
- ◆  $G$ 大于0.5后曲线呈负的斜率,因而处于不稳定工作区,如点 $(G', S')$
- ◆ 负载 $G$ 增大,吞吐率 $S$ 减少表明成功发送的帧减少而发生冲突的帧增加,从而引起更多的重传,因而网络负载 $G$ 进一步增大,恶性循环,最终使吞吐率下降到0为止



# 例题分析-08(1/3)

- ◆ 1972年Robert提出一种能把纯ALOHA系统吞吐率提高1倍的办法，即时隙ALOHA系统，把时间分成离散的事件间隔片
- ◆ **求时隙ALOHA的S与G的关系？** 参数定义同纯ALOHA

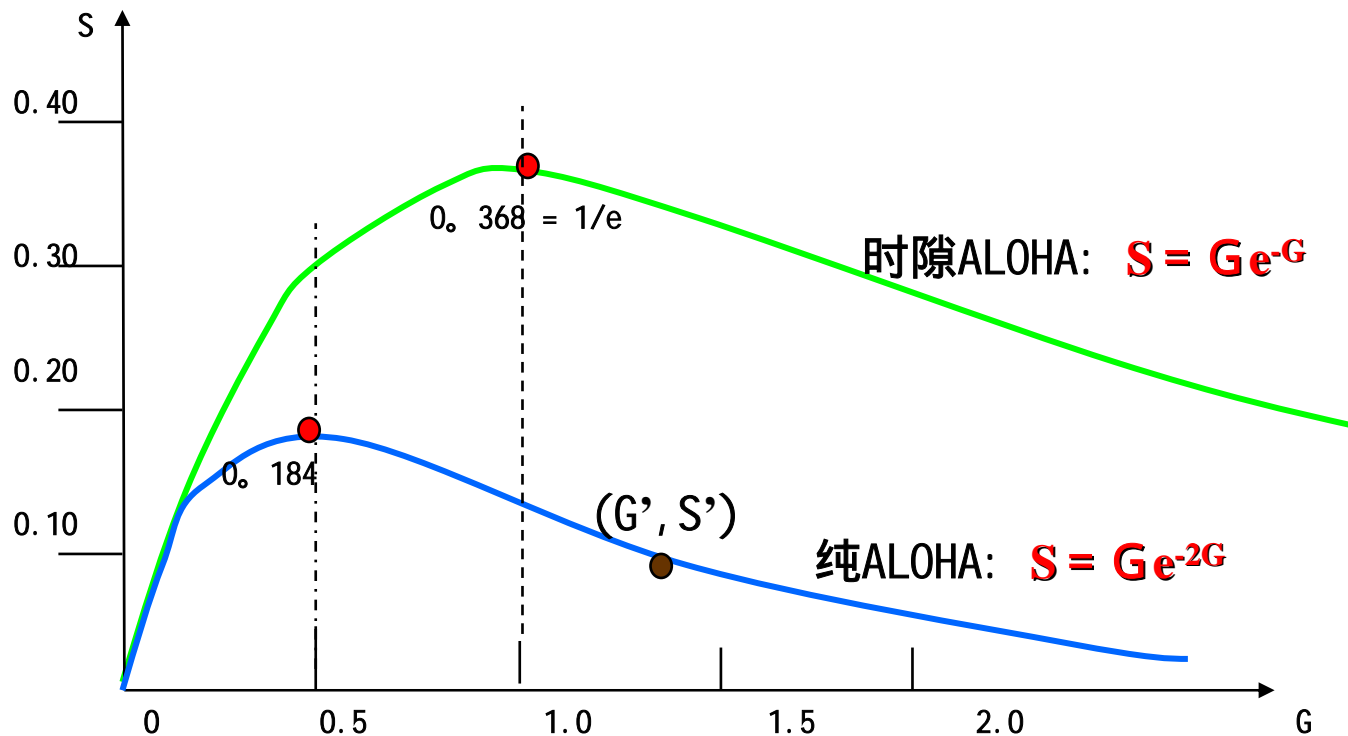


# 例题分析-08(2/3)

- ◆ 帧发送成功的条件是：没有两个或以上的帧在同一时隙内到达。即该帧与前一帧的到达时间间隔应大于  $T_0 - T_x$  且该帧与后一帧的到达时间间隔应大于  $T_x$
- ◆  $P[\text{发送成功}] = P[\text{到达间隔} > T_0 - T_x] \cdot P[\text{到达间隔} > T_x]$   
$$= \int_{T_0 - T_x}^{\infty} a(t) dt \int_{T_x}^{\infty} a(t) dt$$
- ◆  $P[\text{发送成功}] = e^{-G}$  (代进  $\lambda = G/T_0$ )
- ◆  $S = Ge^{-G}$  (时隙ALOHA), '72年Roberts
- ◆  $\text{Max}S(G=1) = e^{-1} = 0.368$ , 即  $G > 1$  为不稳定区

# 例题分析-08(3/3)

- ◆ 信道利用率36.8% ; 或36.8%的时槽无发送



# 例题分析-09(1/9)

- ◆ 一大批ALOHA用户每秒产生50次请求，包括初始请求和重传请求。单位分槽是40ms；试问？
  - 首次尝试成功的概率是多少？
  - K次冲突后成功的概率是多少？
  - 所需要的发送尝试次数的期望值是多少？
- ◆ 解答：在任一帧时（一帧的时间长度）内生成k帧的概率服从泊松分布,  $P_r[k] = G^k e^{-G} / k!$ 
  - 对纯ALOHA，生成0帧的概率是  $Pr[0] = G^0 e^{-G} / 0! = e^{-G}$ ；发送一帧的冲突危险区为2个帧时，在2帧内无其它帧发送的概率是  $e^{-G} \times e^{-G} = e^{-2G}$
  - 对分槽ALOHA，其冲突危险区比纯ALOHA减半，故任一帧内无其它帧发送的概率是  $e^{-G}$

# 例题分析-09(2/9)

- ◆ 若时槽长度为40ms，即每秒25个时槽，产生50次请求，所以平均每个时槽内产生2个请求， $G=2$ 。因此首次尝试的成功概率是 $e^{-2} = 1/e^2$
- ◆ K次冲突的概率
  - $(1 - e^{-G})^k e^{-G} = (1 - e^{-2})^k e^{-2} = 0.135 \times 0.865^k$
- ◆ 尝试k次才能发送成功的概率（即k-1次冲突，k次才成功）为
  - $P_k = e^{-G} (1 - e^{-G})^{k-1}$
  - 那么每帧传送次数的数学期望为 $E = \sum_{k=1}^{\infty} k P_k = e^2 = 7.4$

# 例题分析-09(3/9) : 有限站的吞吐率

## ◆ 以时隙ALOHA为例研究有限站数网络的吞吐率

- 假设共有N个站，各站独立随机发送，一个时隙正好发送一帧；
- 设 $S_i$ 是站i在任意时隙成功发送一个帧的概率，如是 $1 - S_i$ 就是站i在任一时隙没有发送成功（失败或没发）的概率
- 设 $G_i$ 是站i在任意时隙发送一个帧的概率，而 $1 - G_i$ 是站i在任一时隙不发送帧的概率

## ◆ 计算网络的性能，

- 对任何i,有 $S_i = G_i$  因各站独立，所以
- $S_j = G_j \prod_{i \neq j} (1 - G_i)$  ( $j = i$ )，假设各站的统计特性相同 $S_i = S/N$ ， $G_i = G/N$ ；S和G分别表示整个系统的吞吐率和网络负载，则简化为
- $S = G(1-G/N)^{N-1}$  利用极限  $\lim_{n \rightarrow \infty} (1+x/n)^n = e^x$  可得  $S = G e^{-G}$ ；
- 当 $G = G_j = 1$ 时，S达到极大值，为 $S_{\max} = (1-1/N)^{N-1}$

## ◆ 结论：N=1, $S_{\max} = 1$ ; 随着N $\rightarrow 20$ ， $S_{\max}$ 迅速下降到 $1/e = 0.368$



## 例题分析-09(4/9) : 两类用户的吞吐率

- ◆ 假定第一类用户（如FTP）有 $N_1$ 个，每个用户的吞吐率是 $S_1$ ；第二类用户（如WWW）有 $N_2$ 个，每个用户的吞吐率是 $S_2$ ；根据公式则有
  - $S_1 = G_1(1-G_1)^{N_1-1}(1-G_2)^{N_2}$
  - $S_2 = G_2(1-G_2)^{N_2-1}(1-G_1)^{N_1}$
  - 而且，必须满足  $N_1G_1 + N_2G_2 = 1$
  - 4个参数，3个等式，故可得某参数的吞吐率方程式
- ◆ 例如： $N_1=N_2=1, S_1=G_1^2; S_2=G_2^2$

# 无线与有线的不同

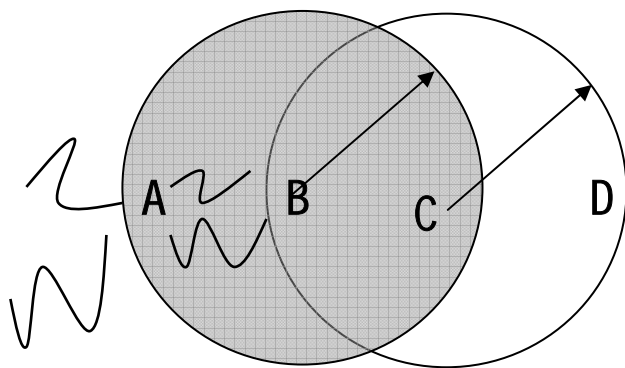
- ◆ 隐藏站点问题 (hidden station problem)
  - 由于站点**距离竞争者太远**，从而**不能发现潜在介质竞争者**的问题称为隐藏站点问题。
  - A向B发送数据的过程中，C由于收不到A的数据，也可以向B发送数据，导致B接收发生冲突。
- ◆ 暴露站点问题 (exposed station problem)
  - 由于非竞争者**距离发送站点太近**，从而导致**介质非竞争者不能发送数据**的问题称为暴露站点问题。
  - B向A发送数据，被C监听到，导致C不敢向D发送数据。



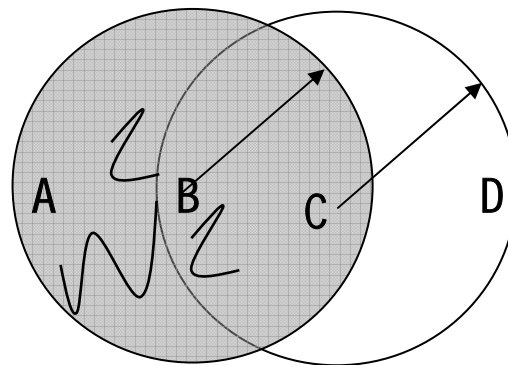
Fig. 4-11. A wireless LAN. (a) A transmitting. (b) B transmitting.

# 冲突避免

- ◆ 使用Ethernet的机制, 但比以太网复杂: **并非所有站点能相互可达**
  - B帧能达到A、C, 但不能到D; C帧能达到B、D, 但不能达A
  - A、C都向B发; 故B站冲突, 且A/C察觉不到该冲突, A/C 互为隐蔽站。 **称隐蔽站问题**
  - B在向A发, C听到这一通信; 但不应影响C向D发; 因为B A并不影响C D的通信。 **称暴露站问题**
  - 与总线网不同, 在不干扰时, **可同时多个点通信**



隐蔽站



暴露站

# 载体带宽和数据率：信道容量

## ◆ 比特率与信号带宽的关系

- 信号的有效带宽随比特率的增加而增加。即当比特率增加时，信号要有更宽的带宽，同时需要传输载体也有更宽的带宽。故媒体的带宽成了比特率的限制
- 若1000bps对应 200Hz; 则2000bps对应400Hz

## ◆ 载体的（信道）容量：

- 媒体能够传输的最大比特率
- 容量取决于编码技术和信/噪比（载体的物理特性）

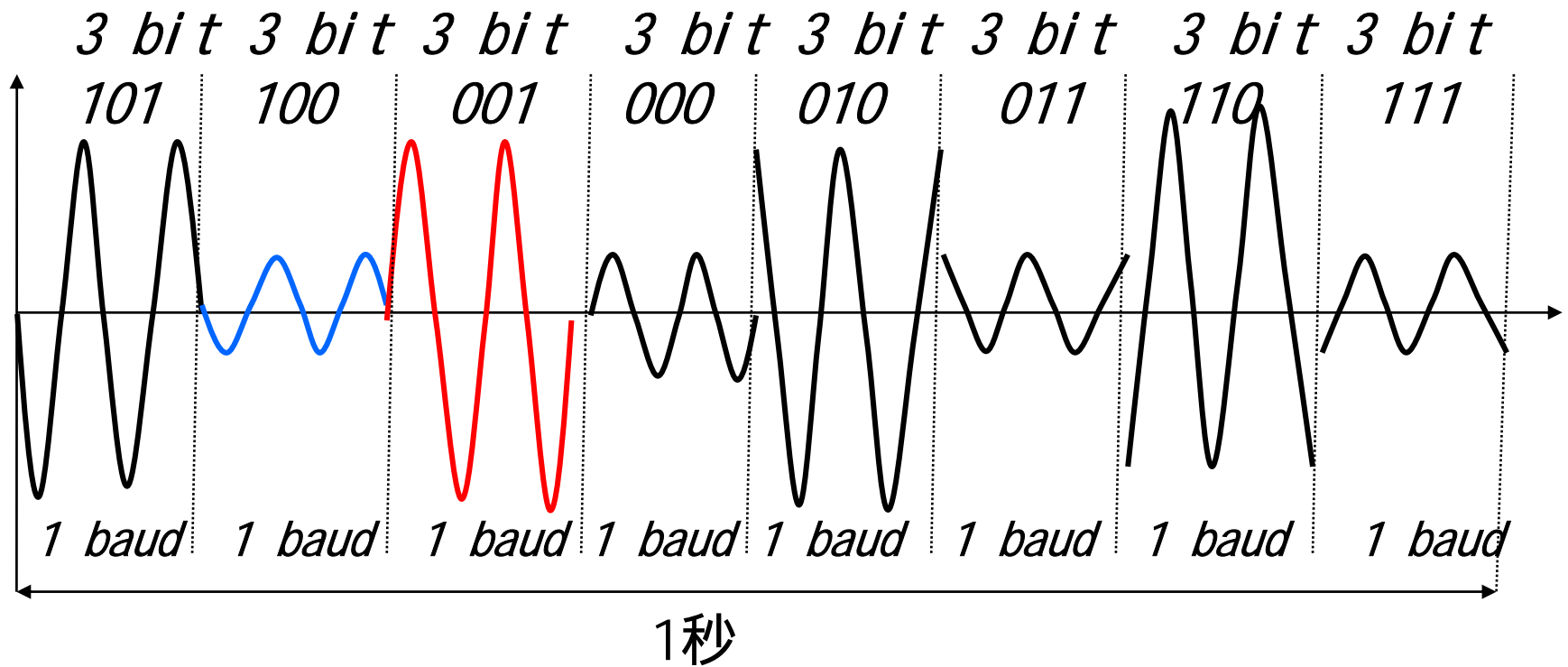
# 比特率与波德率

- ◆ 比特率Bit Rate :
  - 每秒内传输的比特数。
- ◆ 波德率Baud Rate:
  - 每秒内为表示某些比特而需要的信号单元数（或码元数）
- ◆ 当仅当一个信号单元表示一比特时，比特率才等于波德率。

# $8(2^3) - \text{QAM}$

比特率=24

波特率=8



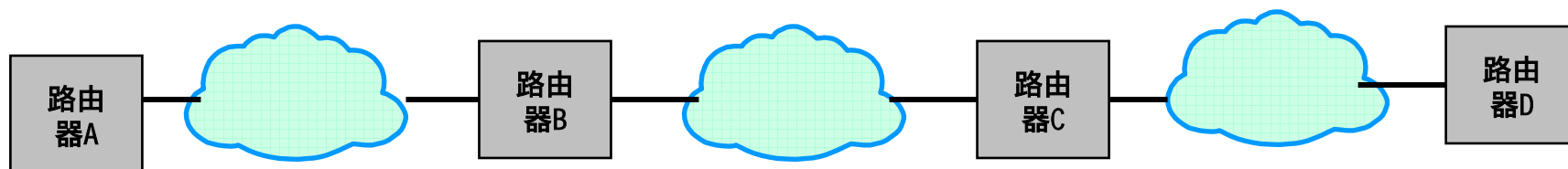
# 比特率和波德率的关系

编码	单位	比特/波德	波德率	比特率
<i>ASK,FSK,2-PSK</i>	<i>Bit</i>	<i>1</i>	<i>N</i>	<i>N</i>
<i>4-PSK,4-QAM</i>	<i>DiBit</i>	<i>2</i>	<i>N</i>	<i>2N</i>
<i>8-PSK,8-QAM</i>	<i>TriBit</i>	<i>3</i>	<i>N</i>	<i>3N</i>
<i>16-QAM</i>	<i>QuadBit</i>	<i>4</i>	<i>N</i>	<i>4N</i>
<i>32-QAM</i>	<i>PentBit</i>	<i>5</i>	<i>N</i>	<i>5N</i>
<i>64-QAM</i>	<i>HexBit</i>	<i>6</i>	<i>N</i>	<i>6N</i>
<i>128-QAM</i>	<i>SepBit</i>	<i>7</i>	<i>N</i>	<i>7N</i>
<i>256-QAM</i>	<i>OctBit</i>	<i>8</i>	<i>N</i>	<i>8N</i>

$$\text{Bit Rate} = \log_2(\text{Bit Units}) \times \text{Baud Rate}$$

# 例题分析-10

TraceRoute = TTL 与 ICMP 的活用



1、发送IP包到B TTL=1

2、超时ICMP报文

3、发送IP包到C TTL=2

4、超时ICMP报文

5、发送IP包到D TTL=3

6、**端口不可达ICMP报文**

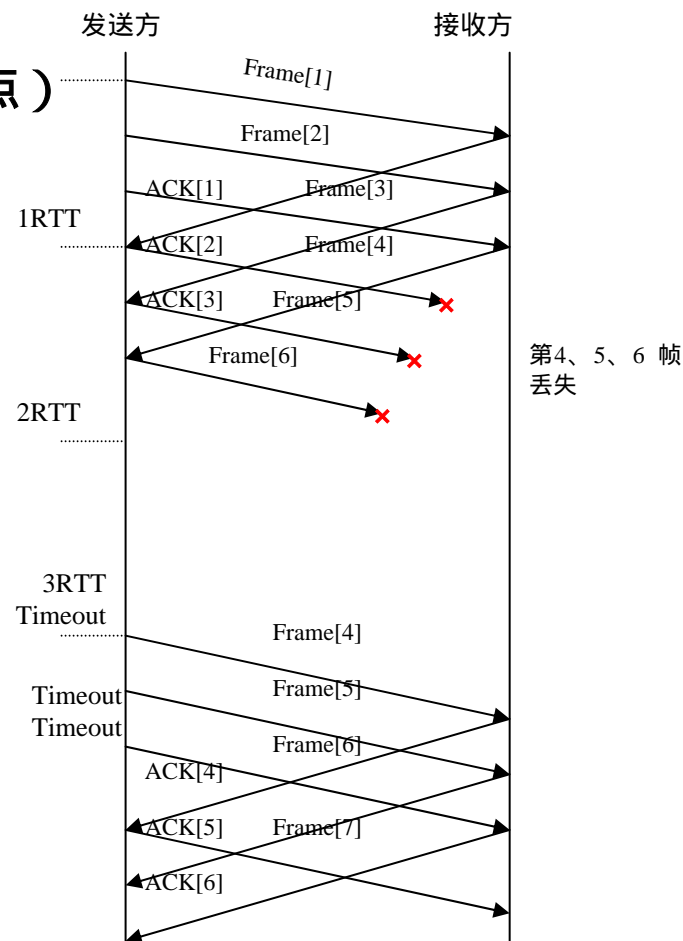
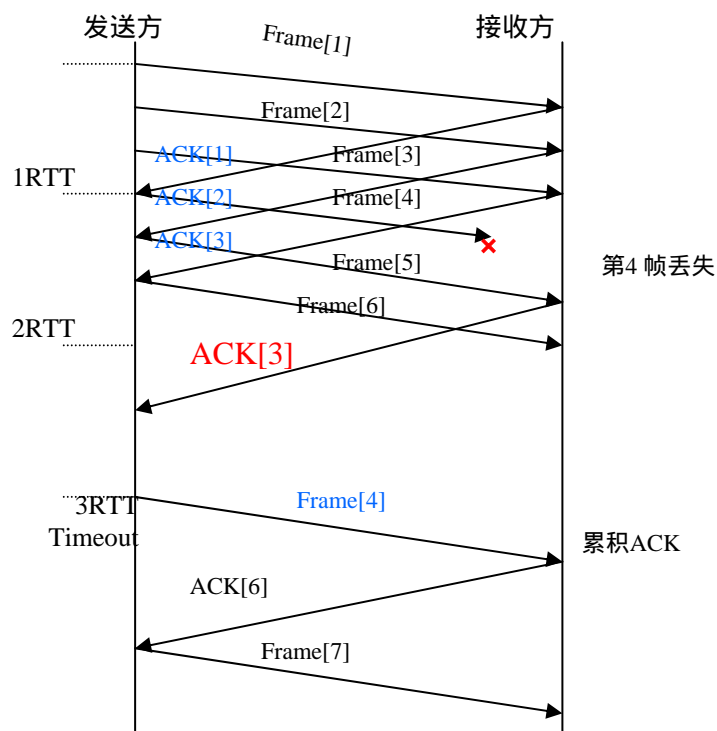
减1使TTL=0，已到目的地不再丢弃而是试图发往一非活动目的端口



# 例题分析-13

- ◆ 对下列情况，画出sws=rws=3 帧的滑动窗口算法的时线图，超时间隔大约为 $2 \times \text{RTT}$

- 帧4丢失了？（注意要标记出Timeout点）
- 帧4 - 6丢失了？（注意要标记出Timeout点）



# 例题分析-17 ( 2/3 )

- ◆ 对慢启动，每个RTT都把窗口尺寸翻倍，在第 $i$ 个RTT之后 ( $i = 1, 2, 3, \dots$ )，其窗口尺寸是 $2^i \times 1\text{KB}$ ，在发送窗口达到1MB，即为 $1\text{MB} = 2^{10} \text{KB}$ 之前，将耗费10个RTT
- ◆ 10个RTT之后，已经发送了  $= \sum_{i=0}^{10} 2^i \text{KB} = (2^{11} - 1) \text{KB} = 2047 \text{KB}$ 
  - 此时发送方拥塞窗口尺寸也达到了 $\text{CW} = 1\text{MB} = 8\text{M bits}$ 。且还未达到网络的最大容量 ( $100 \text{ ms} \times 1000 \text{ Mbps} = 100\text{Mbits} = 12.5\text{MB}$ )
  - 满启动继续在每个RTT继续double上次窗口；2M, 4M, 直到2.049M  $= (10 - 2\text{M} - 2\text{M} - 4\text{M} + 1\text{K})$ ，并等待最后一个RTT后的确认
  - 于是用4个带RTT便可发送完一个10MB文件。故共需14个RTT.
- ◆ 发送全文件的时间  $= 100\text{ms} \times 14 \text{ RTT} = 1.4\text{Sec}$ , 有效吞吐率  $= 10\text{MB} / 1.4\text{s} = 7.1 \text{ M Bps} = 57.1\text{Mbps}$ ；链路带宽的利用率  $= 57.1\text{Mbps} / 1000\text{Mbps} = 5.7\%$

# 例题分析-17 ( 3/3 )

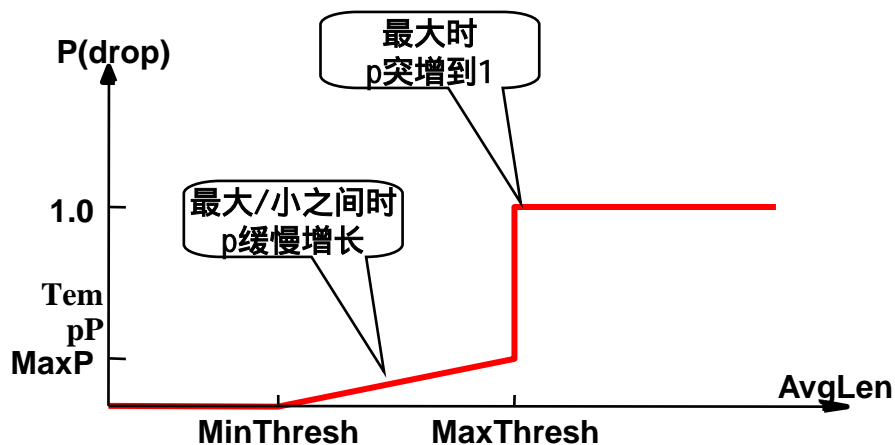
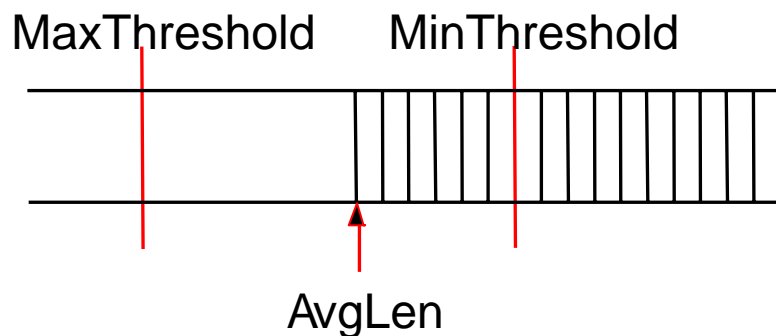
RTT	窗口大小	已发送量	说明
00 th	$2^0$ KB	1 KB	开始发送
01 th	$2^1$ KB	3 KB	Double
02 th	$2^2$ KB	7 KB	...
03 th	$2^3$ KB	15 KB	
04 th	$2^4$ KB	31 KB	
05 th	$2^5$ KB	63 KB	
06 th	$2^6$ KB	127 KB	
07 th	$2^7$ KB	255 KB	
08 th	$2^8$ KB	511 KB	
09 th	$2^9$ KB	1023 KB	
10 th	$2^{10}$ KB	2047 KB	...
11 th	$2^{11}$ KB	4095 KB	Double
12 th	$2^{12}$ KB	8191 KB	剩余不足2M
13 th	2049 KB	10240 KB	10M全部发完
14 th		10240 KB	等待最后确认

## ◆ 注意事项

- 发送方最大窗口、拥塞窗口的概念
- $\text{MaxWindow} = \text{MIN}(\text{CongestionWindow}, \text{AdvertisedWindow})$
- $\text{EffectiveWindow} = \text{MaxWindow} - (\text{LastByte} - \text{LastByteAcked})$
- 慢启动的方法
- 网络容量与接收窗口的概念
- 最后一个RTT的确认

# 例题分析-18 ( 1/2 )

- ◆ 第6章 32题：考虑一个RED网关，其平均队列长度在2个阈值之间且  $\text{MaxP} = 0.02$ 
  - 分别计算  $\text{count} = 1$  和  $\text{count} = 50$  时的丢弃概率  $P_{\text{out}}$ ？
  - 如果头50个分组未被丢弃，再计算该概率。注意它等于  $(1-p_1) \times \dots \times (1-p_{50})$ ？



# 例题分析-18 ( 2/2 )

- ◆ 先求斜线的方程：
  - $\text{TempP} = \text{MaxP} \times (\text{AvgLen} - \text{MinThreshold}) / (\text{MaxThreshold} - \text{MinThreshold})$
- ◆ 使丢弃概率随时间分布：Count记录刚到队列且未被丢弃的包，P随Count增加而缓慢增加；当开始丢弃最后一个包后，丢包概率会随时间增加而增加。
  - $P = \text{TempP} / (1 - \text{count} \times \text{TempP})$
- ◆ 平均队长AvgLen是MaxThreshold和MinThreshold的一半，故倍数 = 1/2. 所以：
  - TempP：即丢弃概率，是P的初始值，此时是MaxP的一半 = 0.01
- ◆ 所以  $P_{\text{cout}} = P = \text{TempP} / (1 - \text{count} \times \text{TempP}) = 1/99 (\text{count}=1) ; 1/50 (\text{count}=50)$
- ◆ 头50个包未被丢弃，再计算此后的丢弃概率
  - 计算  $(1-p_1) \times \dots \times (1-p_{50}) = 49/99 = \frac{98}{99} \times \frac{97}{98} \times \frac{96}{97} \times \dots \times \frac{50}{51} \times \frac{49}{50}$

## 2.3.6 IPv6的地址结构

### ◆ A、地址的三种文本表示，以方便怎样阅读、输入和操作

- 104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255 - 点分十进制表示的128位v6地址
- 冒号16进制，共8个，相同字间距。上面地址为68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF
- 0压缩::表示，对连续长串0用::代替，一个地址中仅出现一次，例如：
  - ☞ 1080:0:0:0:8:800:200C:417A    1080::8: ...    ; unicast address
  - ☞ FF01:0:0:0:0:0:0:101    FF01::101    ; multicast address
  - ☞ 0:0:0:0:0:0:0:1    ::1    ; loopback address
  - ☞ 0:0:0:0:0:0:0:0    ::    ; undefined address
- 混合表示，x: x: x: x: x: x: d.d.d.d, x :表示16进制(16 Bits), d.表示10进制(8 Bits)
  - ☞ 0:0:0:0:0:0:13.1.168.3    或    ::13.1.168.3
  - ☞ 0:0:0:0:0:FFFF:129.144.52.38    或    ::FFFF:129.144.52.38

# IPv6地址表示

- ◆ 16字节长的地址表示成用冒号（:）隔开的8组，每组4个16进制位，例如，

**8000:0000:0000:0000:0123:4567:89AB:CDEF**

- ◆ 由于有很多“0”，有三种优化表示

- 打头的“0”可以省略，0123可以写成123；
- 一组或多组16个“0”可以被一对冒号替代，但是一对冒号只能出现一次。上面的地址可以表示成

**8000::123:4567:89AB:CDEF**

- IPv4地址可以写成一对冒号和用“.”分隔的十进制数，例如  
**::192.31.20.46**

# B、地址结构前缀的表示

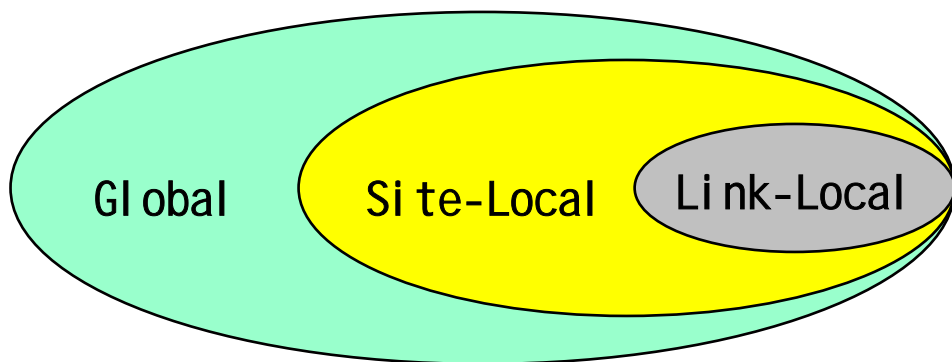
## ◆ 类似CIDR形式

- IPv6地址/前缀长度，长度是10进制，表明地址最左端连续比特个数
- 正确表示12AB00000000CD3的60bits前缀是
  - ☞ 12AB: 0000: 0000: CD30: 0000: 0000: 0000: 0000/60
  - ☞ 12AB::CD30: 0: 0: 0: 0/60
  - ☞ 12AB: 0: 0: CD30::/60
- 不正确的表示为
  - ☞ 12AB: 0: 0: CD3/60 ; 可能丢失前面的0而是0CD3
  - ☞ 12AB::CD30/60 ; 可理解为12AB: 0: 0: 0: 0: 0: 0: CD30
  - ☞ 12AB::CD3/60 ; 可理解为12AB: 0: 0: 0: 0: 0: 0: 0CD3



# IPv6的地址模式

- ◆ 地址分配到接口：
  - 这同v4一样，没有变化
- ◆ 一个接口可有多多个地址
- ◆ 地址有范围之分
  - Link Local
  - Site Local
  - Global
- ◆ 地址有寿命
  - 有效的
  - 永久的
- ◆ 地址结构
  - 前缀 + 接口ID



# I Pv6的寻址

## ◆地址类型：

- **Unicast: One to One( Global,Link local,Site local, Compatible)**
- **Anycast:One to Nearest(Allocated from Unicast)**
- **Multicast:One to Many**
- **Reserved**

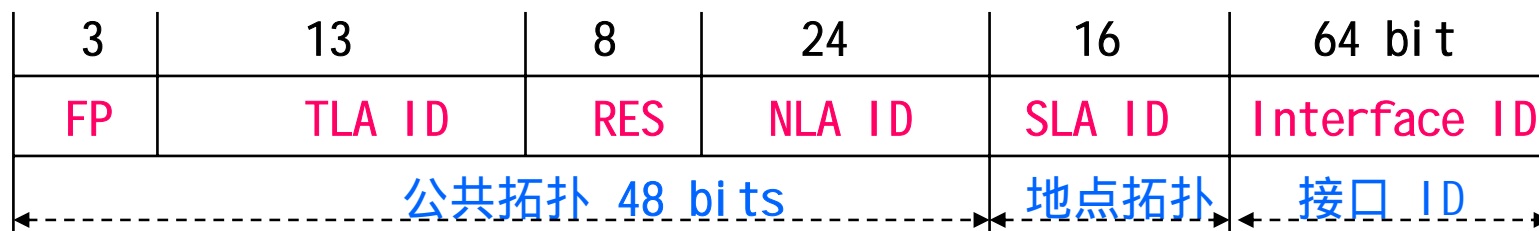
## ◆单个接口可能分配有任何单播、近播和组播地址

## ◆广播由组播代替

# IPv6全局可聚类Uni cast地址格式

## ◆ 共分3级，6个段

- 001: 全球可聚合Uni cast地址
- TLA ID：顶级聚合标识符，可指派给ISP或拥有这些地址的汇接点
- RES: 留做将来使用 - Reserved for future use
- NLA ID: 次级聚合标识符，可指派给特定的用户
- SLA ID：地点级聚合标识符，指明单个的地点
- 接口ID：接口标识符Interface Identifier



# 信道容量的三大定理

- ◆ Shannon(仙农1948年)定理：信道带宽与信道容量的关系
  - 把奈氏定理扩展到受随机热噪声影响的信号
  - 对任何带宽为 $W$  Hz，信噪比为 $S/N$ 的信道，其最大数据传输率(容量)  $C = W \log_2(1+S/N)$  bps； $S, N$ 分别为信号和噪声的功率
- ◆ Nyquist(奈奎斯特)定理：信道对传输速率的限制
  - 如果一个任意的信号通过带宽为 $H$ 的低通滤波器，那么每秒采样 $2W$ 次就能完整重现通过这个滤波器的信号，更高频分量已被滤掉，无法再恢复(1924年)。
  - 如果信号电平分为 $V$ 级，奈氏推导出一个有限带宽无噪声( $N=0$ )时， $C = 2W \log_2 V$  bps，信道的最大数据传输率由带宽决定， $D = 2W \log_2 V$  bps（是仙农的特例）
- ◆ 在有白色高斯噪声( $N=Wn_0$ )时， $n_0$ 为噪声的单边功率谱密度，从而有
 
$$\lim_{S/n_0 \rightarrow \infty} C = \lim_{S/n_0 \rightarrow \infty} W \log_2 (1+S/(Wn_0)) = S/n_0 \log_2 e = 1.44 S/n_0$$
- ◆ **信道容量：单位时间内信道上能传输的最大比特数**

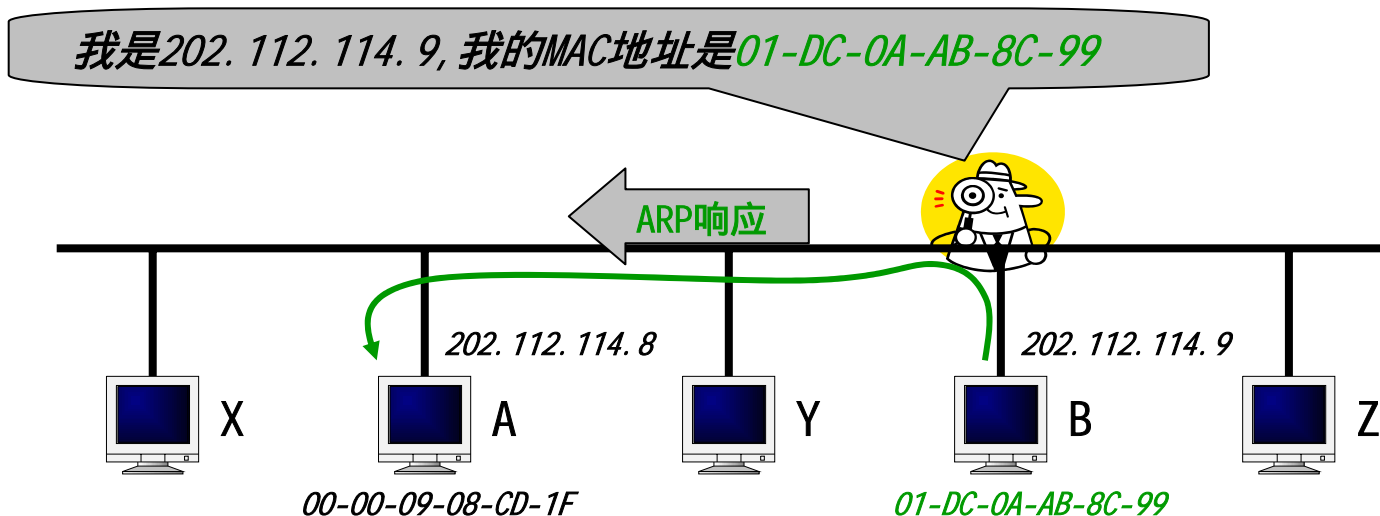
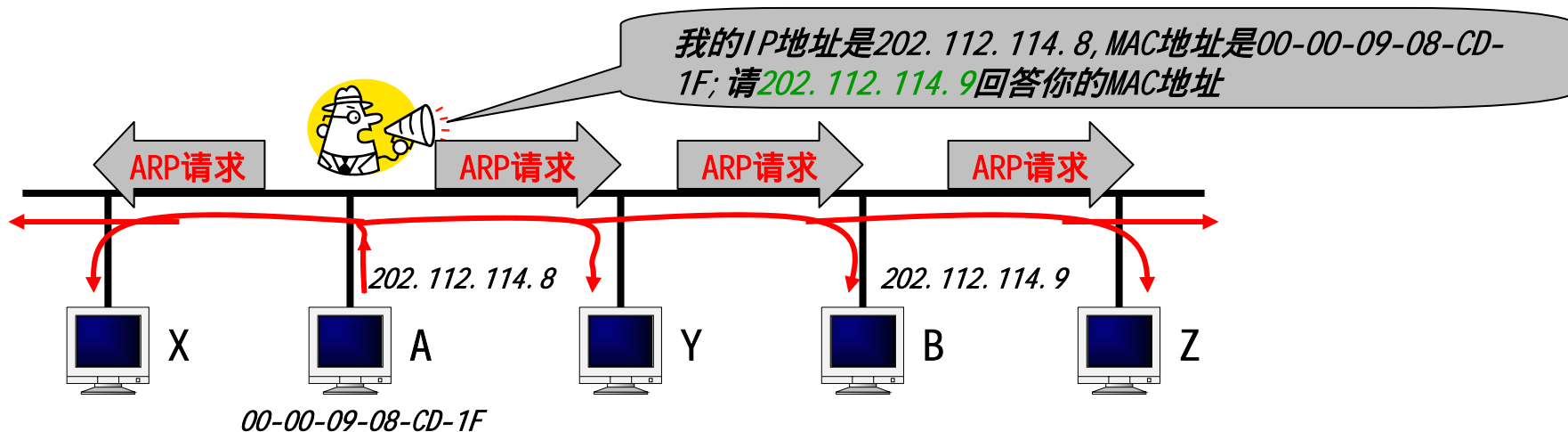
# 三大定理的意义

- ◆ 提高信号与噪声的功率比能够增加信道容量
- ◆ 当  $N \rightarrow 0$ ，信道容量  $C \rightarrow \infty$ ，无干扰信道的容量为  $\infty$ ，信道的最大数据传输率完全由带宽决定
- ◆ 信道容量一定时，带宽  $W$  和  $S/N$  可以互换
- ◆ 近距离传输时，噪声小、信号功率损耗低，可以采用未经调制的脉冲直接传输，此时容量只与信号带宽有关
- ◆ 如果远距离传输，则必须提高信噪比，除选择好的调制方式外，很重要一点是增加信号的功率
- ◆ 当信道容量一定时，如果信号频率过低，则造成信道浪费，此时可让不同信息源共享信道，即信道复用
- ◆ 只要信道容量  $C \geq R$ （数据发送速率），就可找到一种信道编码方式实现无误传输；若  $C < R$ ，则不可能实现无误传输，因此不同信源发送数率和应小于信道容量  $C$

# ARP寻址过程

- B接收A的广播包并把A的 $IP_A \rightarrow MAC_A$ 放入自己的 $ARP_B$ 表
- B组织ARP响应包，其中填入自己的 $IP_B \rightarrow MAC_B$ 地址并直接发给A
- A受到包后，提取 $IP_B \rightarrow MAC_B$ 放入 $ARP_A$ 并把原缓存队列的数据包加上 $MAC_B$ 发送出去。
- 若一ARP表项很久未使用则从ARP表中删除，以节约空间和时间

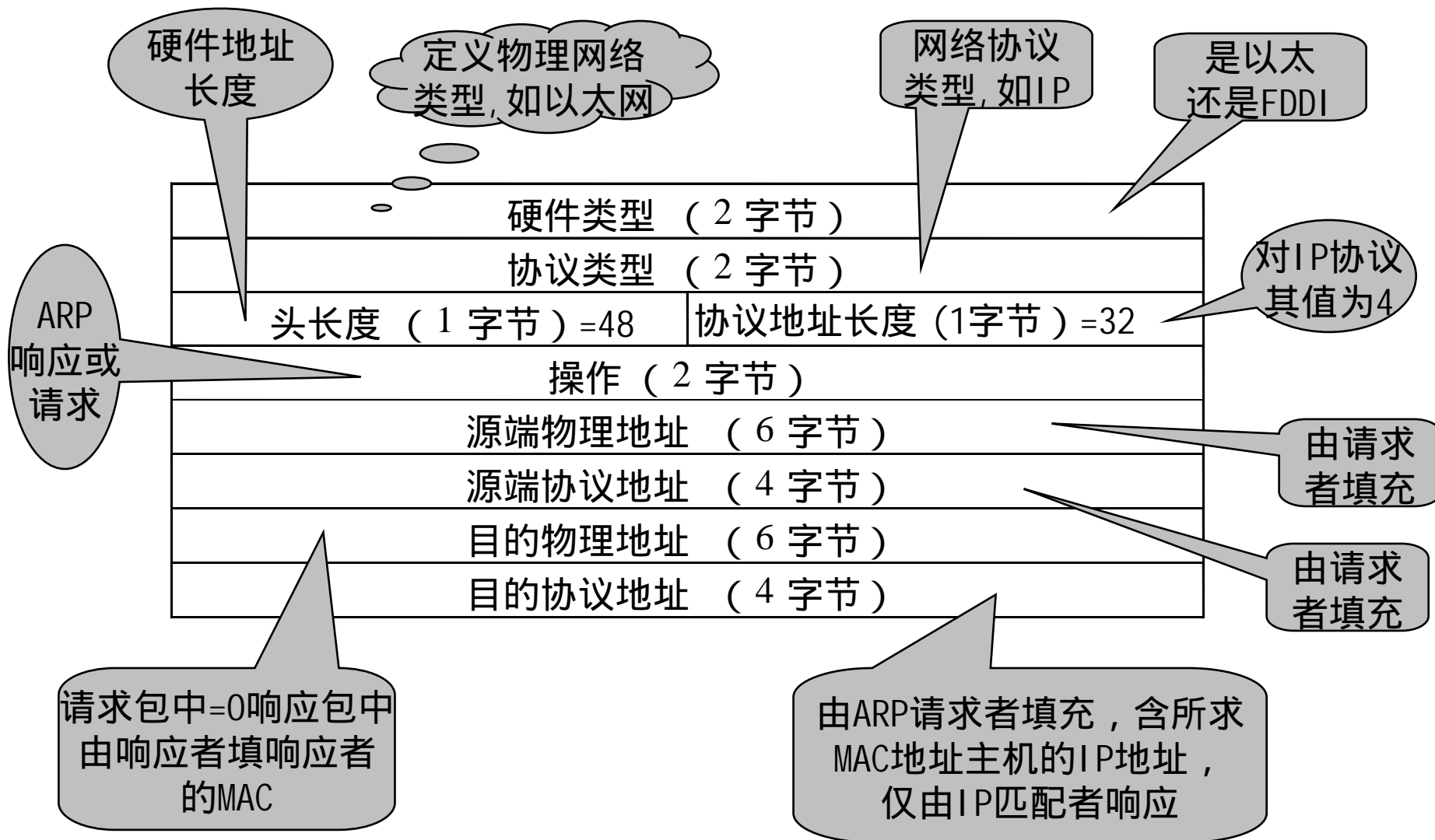
# ARP解析过程



- ◆ 每个主机设有一个ARP Cache表,是主机(或路由器)IP地址到MAC地址的映射,生存时间10-20分钟
  - A主机上的ARP进程在本LAN上发广播请求主机B的MAC帧,目的MAC为全0
  - 本LAN上所有主机都接收此ARP包
  - 主机B在包中发现自己的IP地址,存A的MAC入自己的Cache,并向A发响应,单播发送自己的MAC地址
  - 主机A收到主机B的MAC地址,并存入Cache
- ◆ 只解析同一LAN上主机的映射,解析对用户透明
- ◆ 直接用MAC的困难--异构硬件/网卡更换的不透明
- ◆ RARP: 无盘工作站只有ROM,不能记忆其IP地址; DHCP不能固定分配其IP地址,但其MAC是相对固定



# ARP 包格式

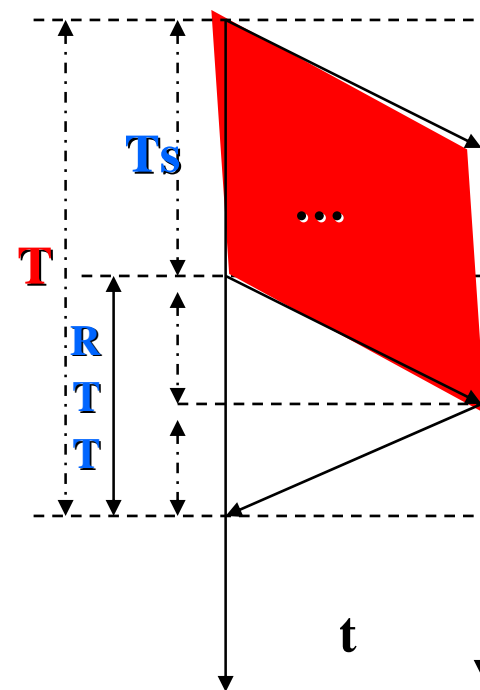


# 链路速率与端端吞吐率

- ◆ **问题**：一个TCP连接下面使用256 Kbps的链路，其端到端的单向时延是128ms。经测试，发现其**吞吐率**只有128 Kbps。如果忽略PDU封装的协议开销以及接收方应答分组的发送时间，试问**发送窗口**是多少字节？

- ◆ **解答**:

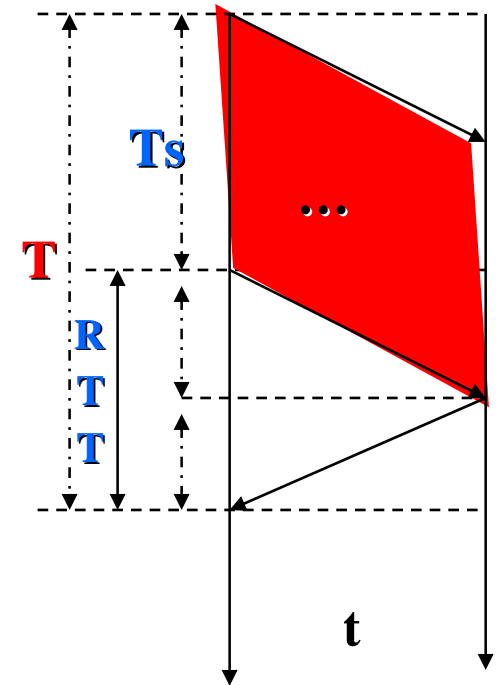
- 设发送窗口为W比特，则有效发送（正确收到ACK）W字节所需要的时间是 $T = [W/256 \text{ Kbps} + 2 \times 128 \times 10^{-3} \text{ s}]$
- 发送w比特的实际吞吐率是 $W / [W/256 \text{ Kbps} + 2 \times 128 \times 10^{-3} \text{ s}] = 128 \text{ Kbps}$
- $x = 65536 \text{ 比特} = 8192 \text{ 字节}$



- ◆ **问题：**通信信道带宽为1Gbps, 端端时延10ms, TCP的发送窗口65535字节, 可能到达的最大吞吐量是多少? 信道的利用率是多少?

- ◆ **解答：**

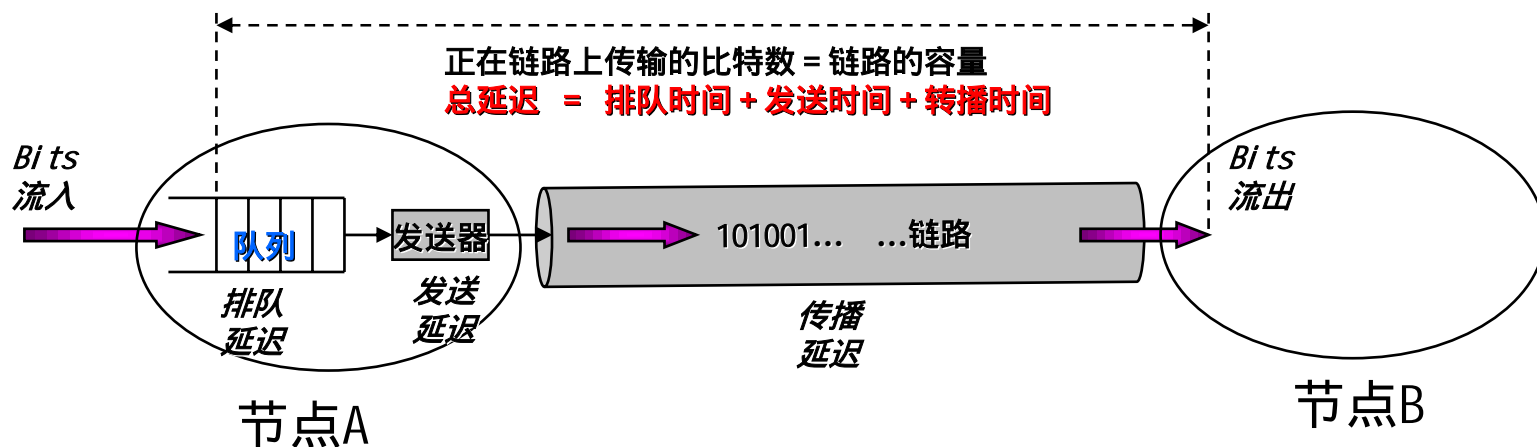
- 关键是理解链路或信道的带宽与端端吞吐量的概念。
- $\text{Max T}_{\text{Pee}} = \text{Win} / (\text{Ts} + 2 \text{RTT})$ ;  $\text{Ts} = \text{Win} / \text{Ls}$ ;  $\text{win} = 65535, \text{Ls} = 1\text{Gbps}$
- $\text{Ts} = 0.52428\text{ms}$ ;  $\text{T}_{\text{pee}} = 25.5\text{Mbps}$ ;  $\text{利用率} = 2.5\%$



◆ **问题**：源站到目站相距20Km，信号在媒体中的传播速率为200Km/ms；若一分组长度为1KB，而其发送时间等于信号的往返传播时延，求数据的发送速率？

◆ **解答**：

- 排队时延 = 数据存储转发所需的时间（可称处理时间）
- 发送时延 = 数据长度/信道带宽
- 传播时延 = 信道长度/电磁波在信道上的传播速率
- 往返时间 =  $2 \times 20\text{Km} / (200\text{Km/ms}) = 0.2\text{ms}$ ;
- 数据传输速率 =  $1\text{k} \times 8 \text{ bit} / 0.2\text{ms} = 40.96 \text{ Mbps}$



# DHT - Chord

- ◆ 在一维空间（环）中给每个节点和文件一个唯一的id
  - 例如从 $[0 \dots 2^m]$ 中选取， $m=128-160$ bits
  - 节点ID通常是其IP地址的hash
  - 文件ID通常是其关键字或内容的hash
- ◆ 属性
  - 全部节点数是 $N$
  - 每个节点路由表大小是 $O(\log N)$
  - 保证在 $O(\log N)$  跳内找到文件

# 路由表

<i>ID</i>	<i>Next_hop</i>	<i>file</i>
	...	

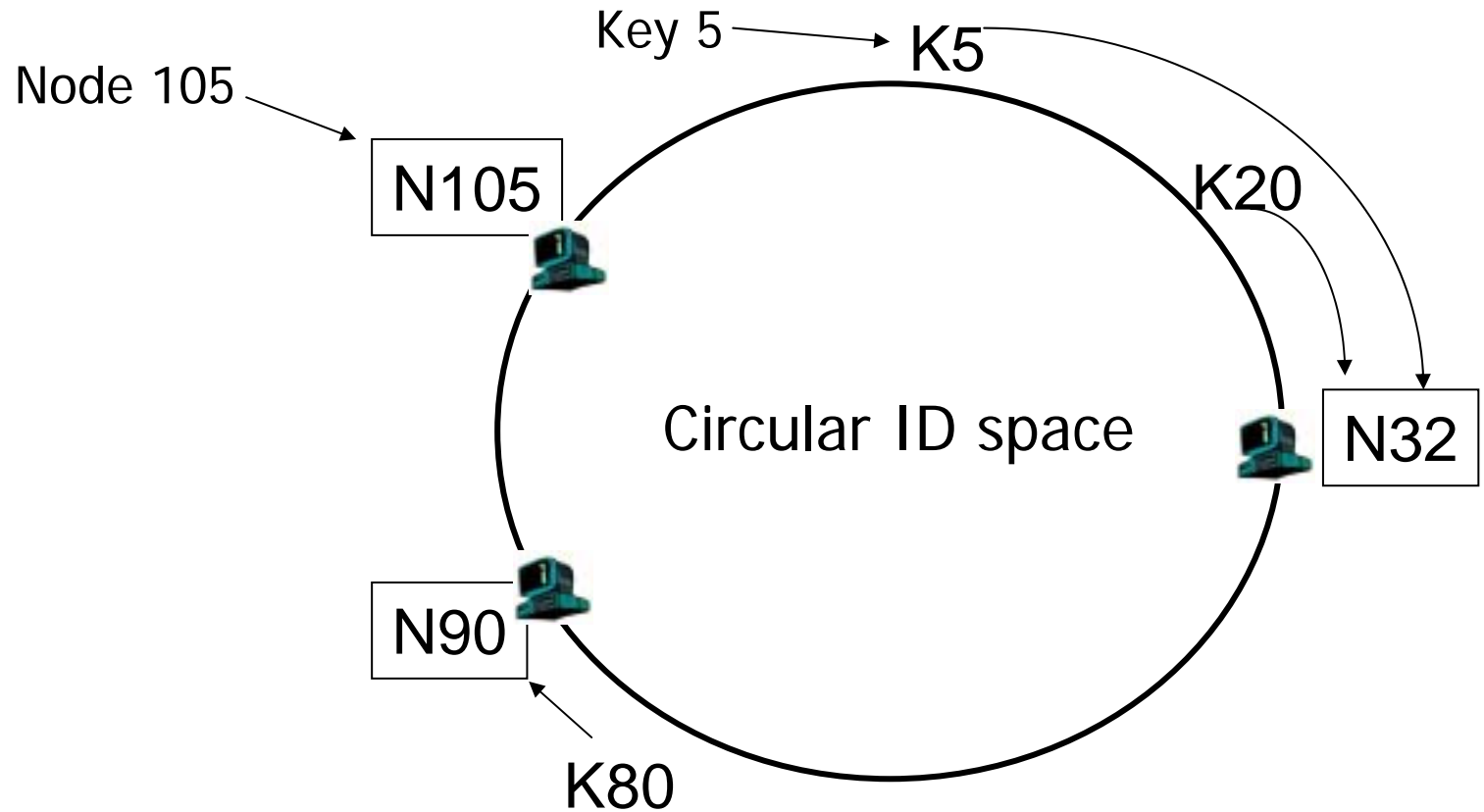
## ◆ 路由表内容

- id - 文件标识符
- next\_hop - 存储文件id的另一个节点
- file - 保存在本地的id标识文件

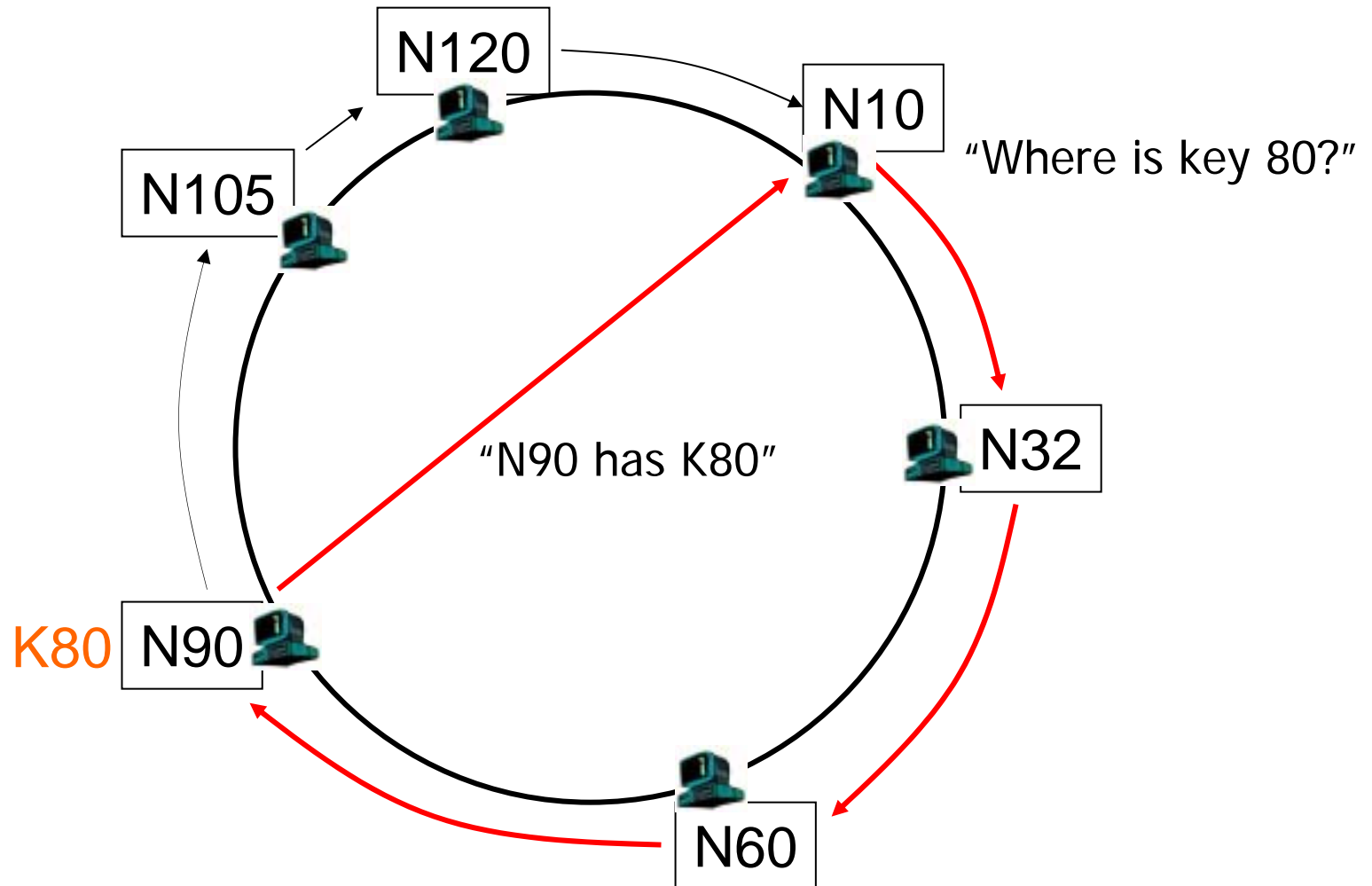
## ◆ 搜索过程

- 如果文件id存储在本地，停止搜索，上传文件
- 如果不在本地，搜索路由表中**最接近**的id，将请求转到next\_hop
- 如果所有节点都没有找到，返回失败，返回路由表中下一个最接近的id

# Chord : 相容哈希



# Chord : 基本查找

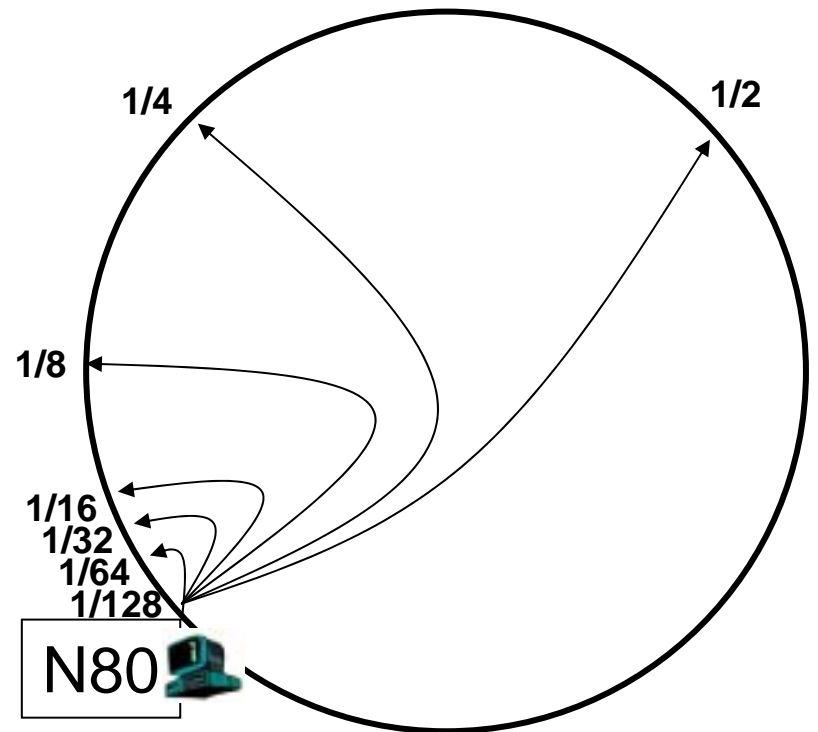




# Chord : “Finger Table”

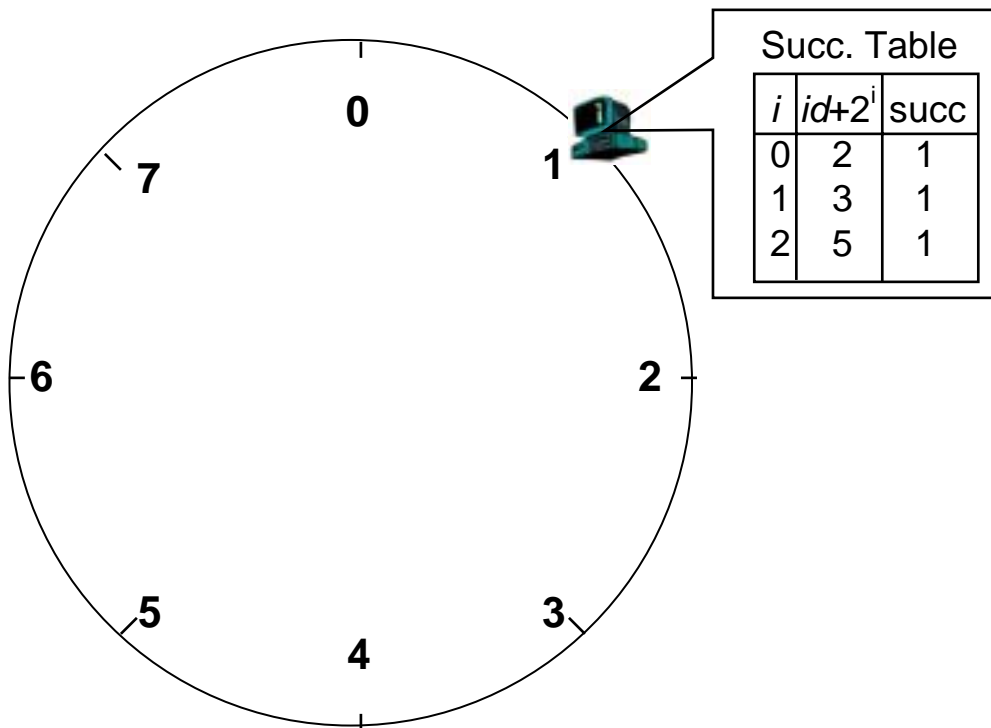
◆改进查找

◆以1/2弦



# Chord原理

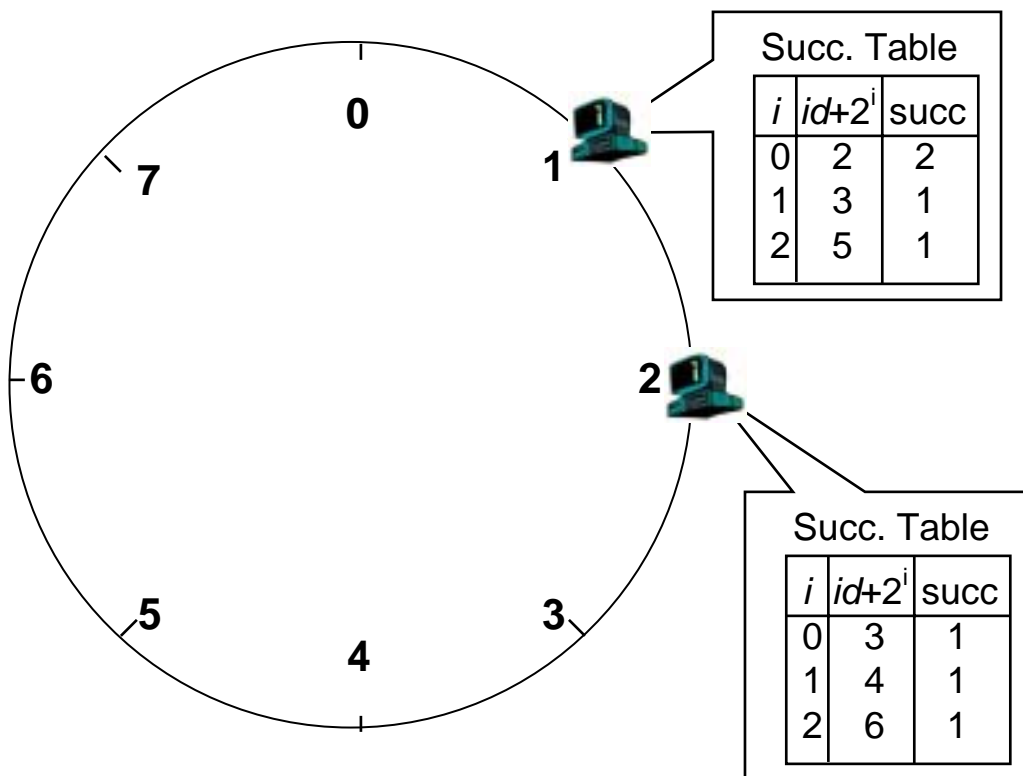
## ◆节点1加入



假设id空间是  $[0..8]$

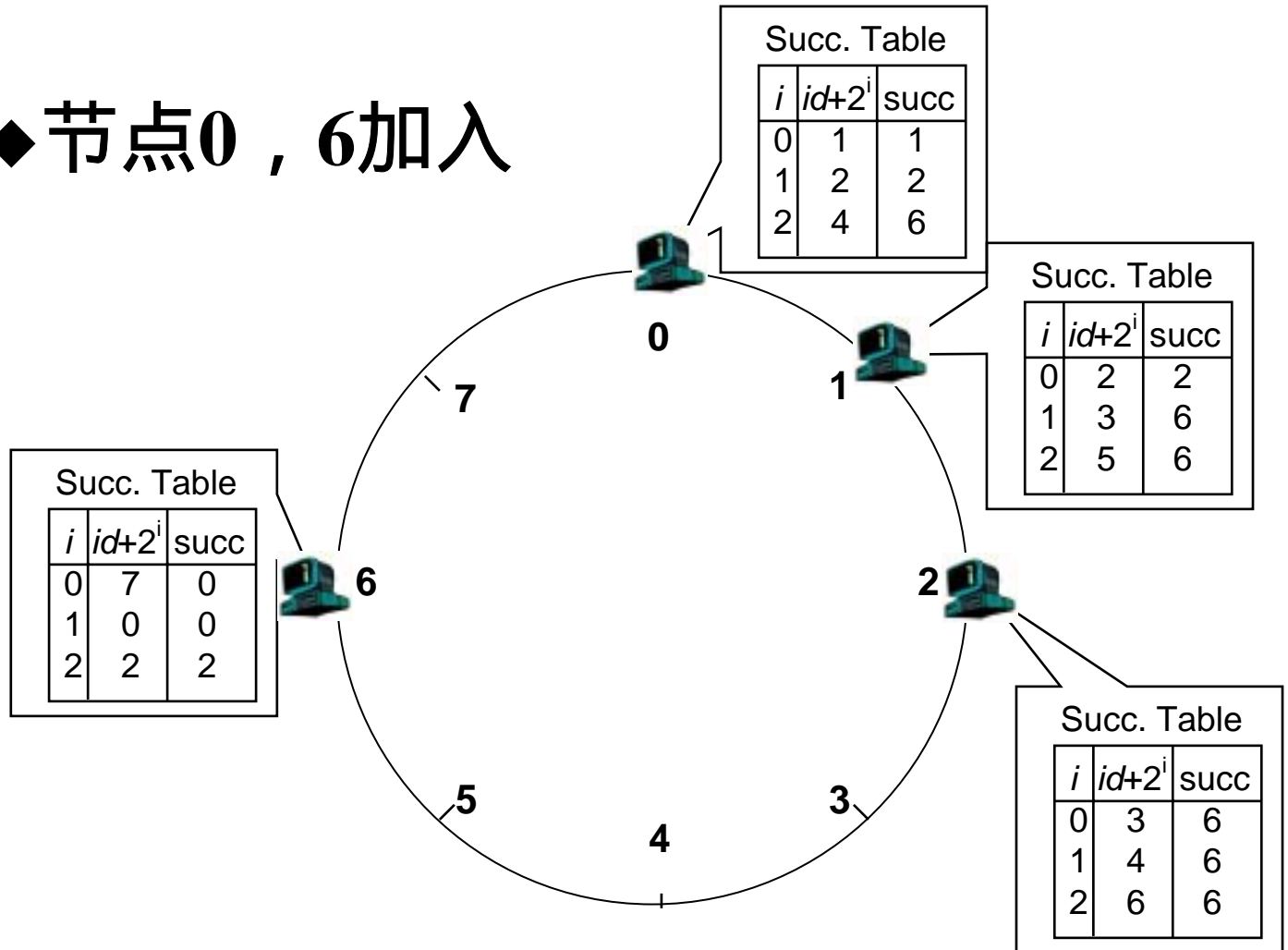
# Chord原理

## ◆节点2加入



# Chord原理

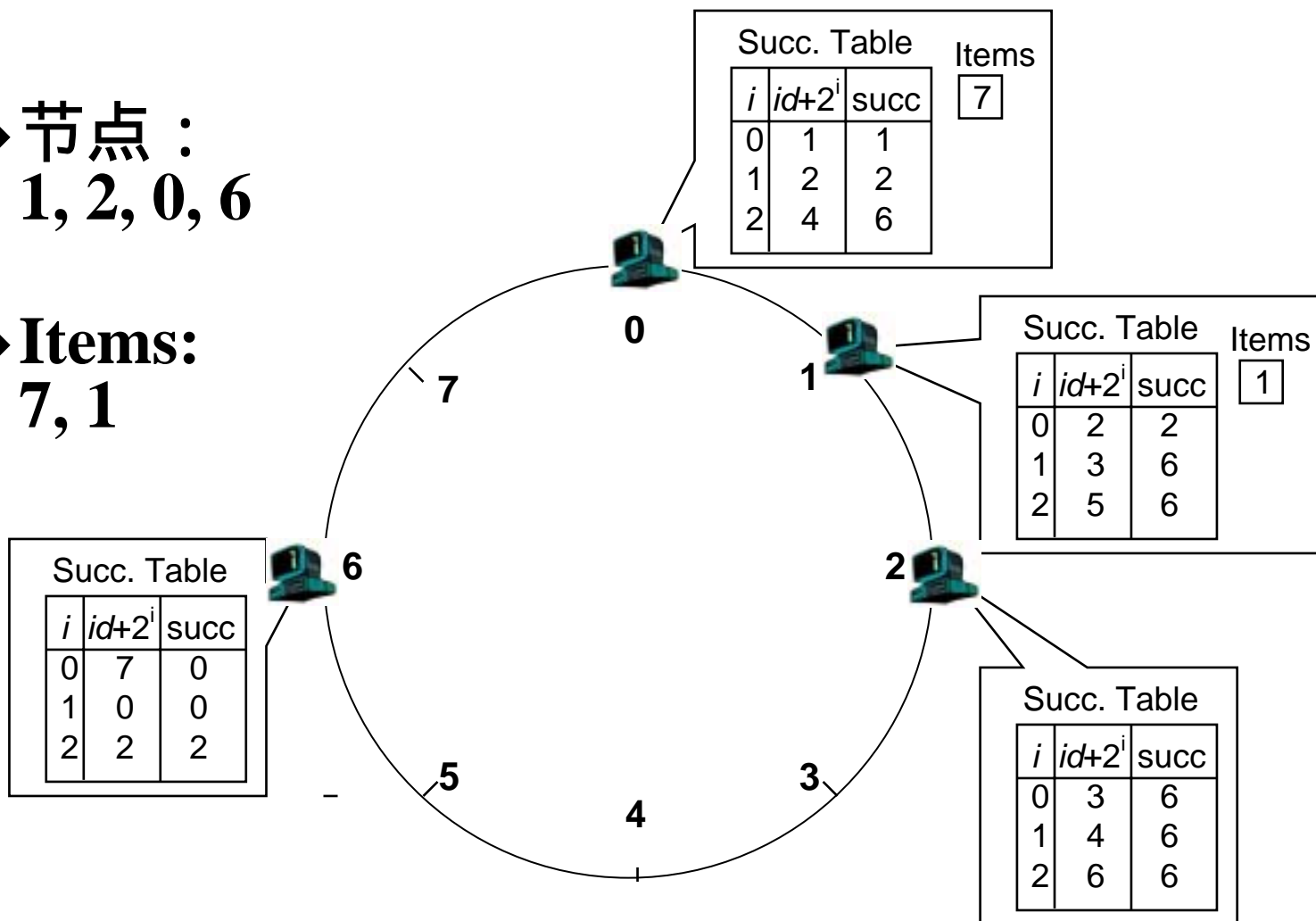
◆节点0, 6加入



# Chord原理

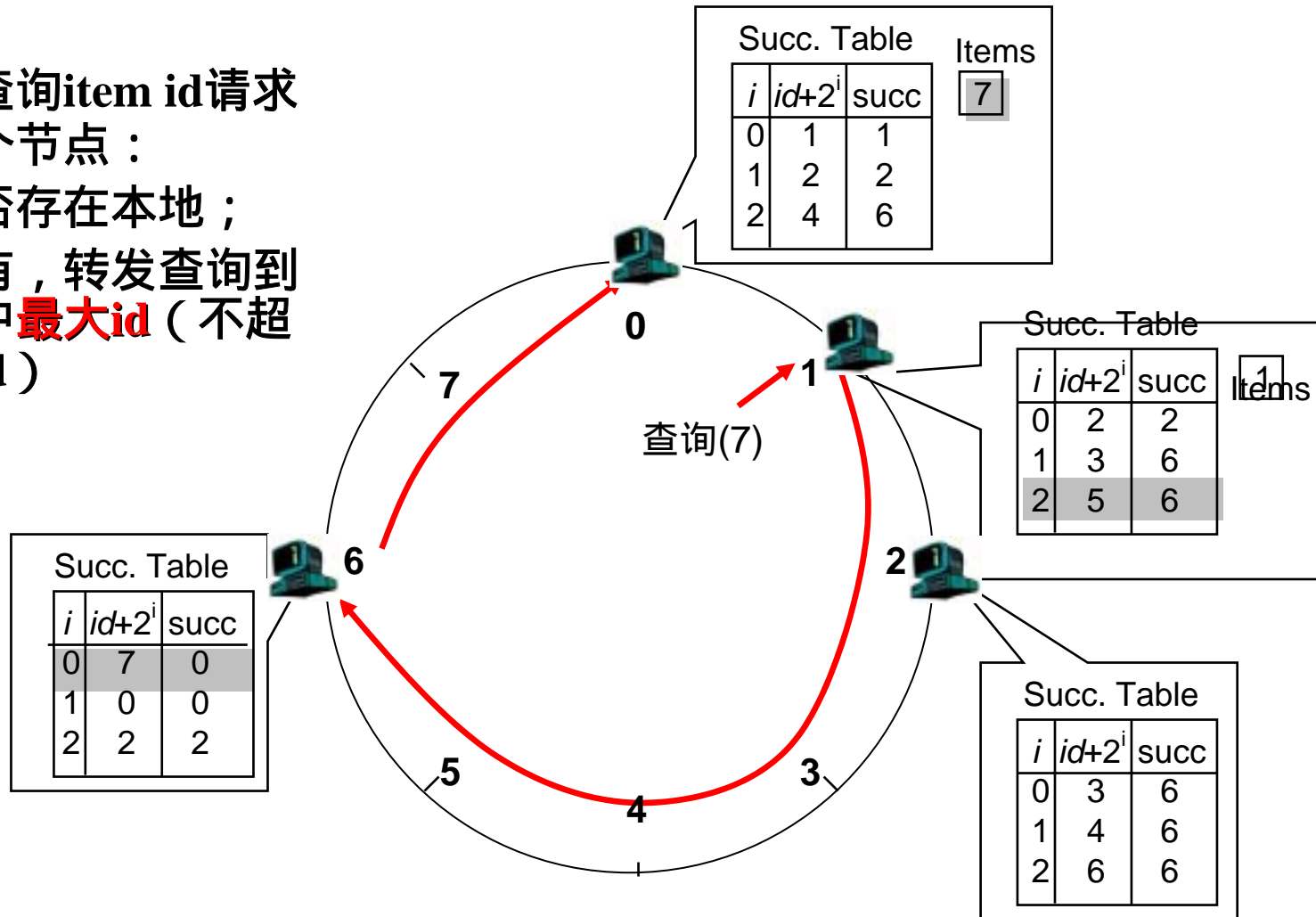
◆节点：  
1, 2, 0, 6

◆Items:  
7, 1



# Chord原理

- ◆ 当收到查询item id请求时，一个节点：
- ◆ 检查是否存在本地；
- ◆ 如果没有，转发查询到路由表中**最大id**（不超过请求id）



# Chord : 总结

## ◆路由表大小?

- $\log N$

## ◆路由时间?

- 每跳逼近剩余空间的 $1/2$  ; 最坏情况下, 数据文件在其前继节点, 则每跳路由表内可提供剩余空间的 $1/2$  供新的查询扫描, 直至最后命中空间 = 1, 固有:

- $N/2^1 \quad N/2^2 \quad \dots \quad N/2^m (1) \Rightarrow$  预期  $O(\log N) = m$  跳

## ◆优点:

- 可靠查询
- 每个节点搜索和维护  $O(\log N)$

## ◆缺点:

- 没有实际使用 ( 只有一个文件共享应用 )
- 支持非精确查找比较困难

# 4.6.8 令牌桶算法

## (The Token Bucket Algorithm)

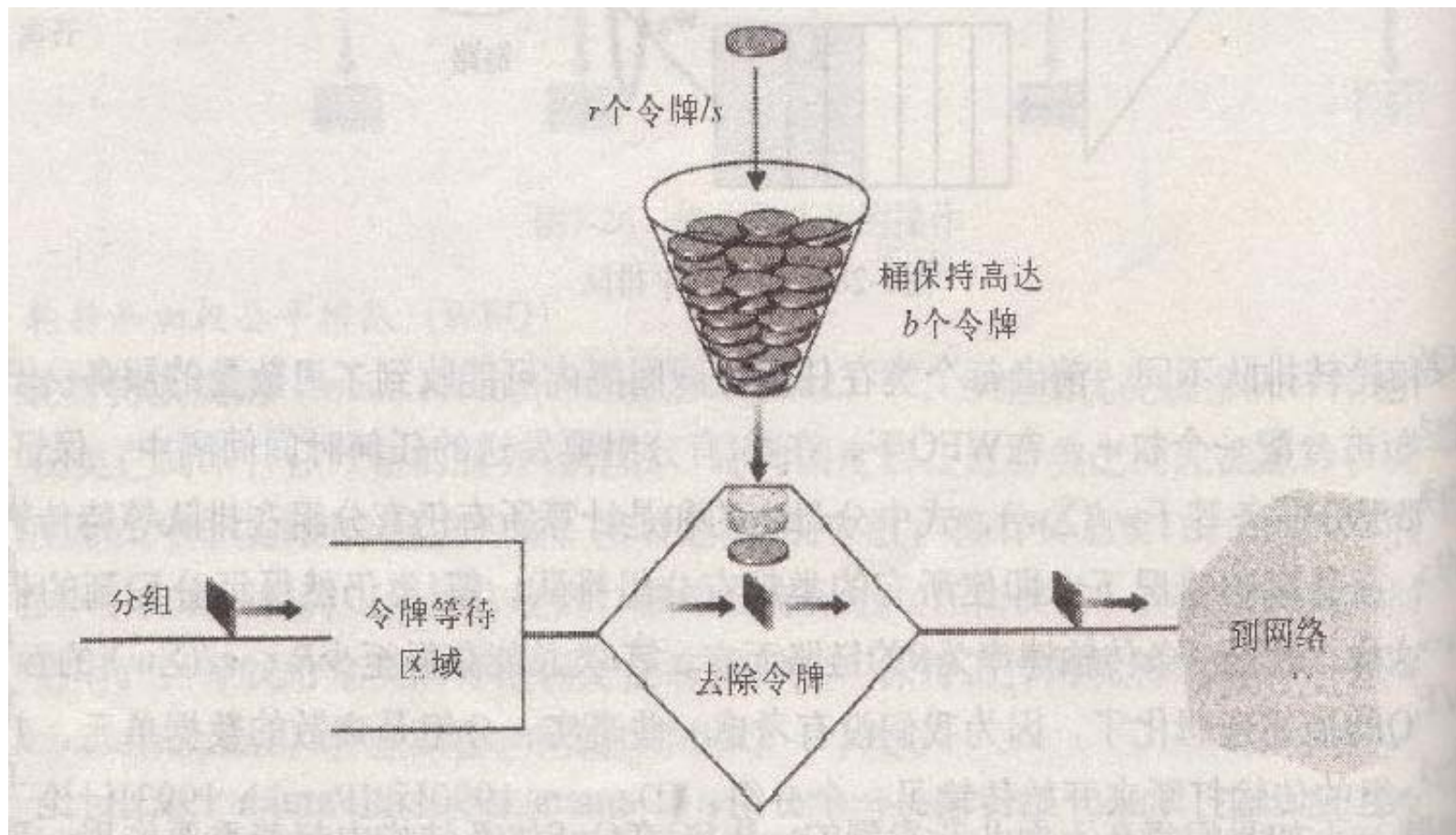
### ◆ 基本思想

- 漏桶算法不够灵活，因此加入令牌机制；
- 基本思想：漏桶存放令牌，每 $\Delta T$ 秒产生一个令牌，令牌累积到超过漏桶上界时就不再增加。包传输之前必须获得一个令牌，传输之后删除该令牌；

### ◆ 漏桶算法与令牌桶算法的区别

- 流量整形策略不同：漏桶算法不允许空闲主机**积累发送权**，以便以后发送大的突发数据；令牌桶算法允许，最大为桶的大小。
- 漏桶中存放的是**数据包**，桶满了丢弃数据包；令牌桶中存放的是**令牌**，桶满了丢弃令牌，**不丢弃数据包**。





### ➤ 令牌漏桶整形算法

➤ 在时间间隔  $t$  内, 网络中的最大包数  $= rt + b$

➤ 故令牌长生率  $r$  成为包进入网络的平均速率

➤ 用2个串连漏桶可能整形网络峰值速率

## 4.7.1 加法式增加/乘法式减少

- ◆ Congestion Window 是源端TCP为每个连接维护的一个状态变量：
  - 以限制在给定时间内容许传输的数据量
  - 是流控 Advertised Window 的一个副本
- ◆ 容许的最大未确认字节数就是现在的最小 Congestion Window 和 Advertised Window
  - $\text{MaxWindow} = \text{MIN}(\text{CongestionWindow}, \text{AdvertisedWindow})$
  - $\text{EffectiveWindow} = \text{MaxWindow} - (\text{LastByte} - \text{LastByteAcked})$

# TCP怎样学到CW

- ◆AW靠连接的接收方来发送，没有谁发送CW到发送方
- ◆答案：TCP源根据它从现在网络觉察到的拥塞程度来设置CW
  - 当拥塞程度上升时，就减少CW
  - 当拥塞程度下降时，就增加CW
  - 上述二者合在一起就叫：加法式增加/乘法式减少

# 怎样决定是否拥塞？

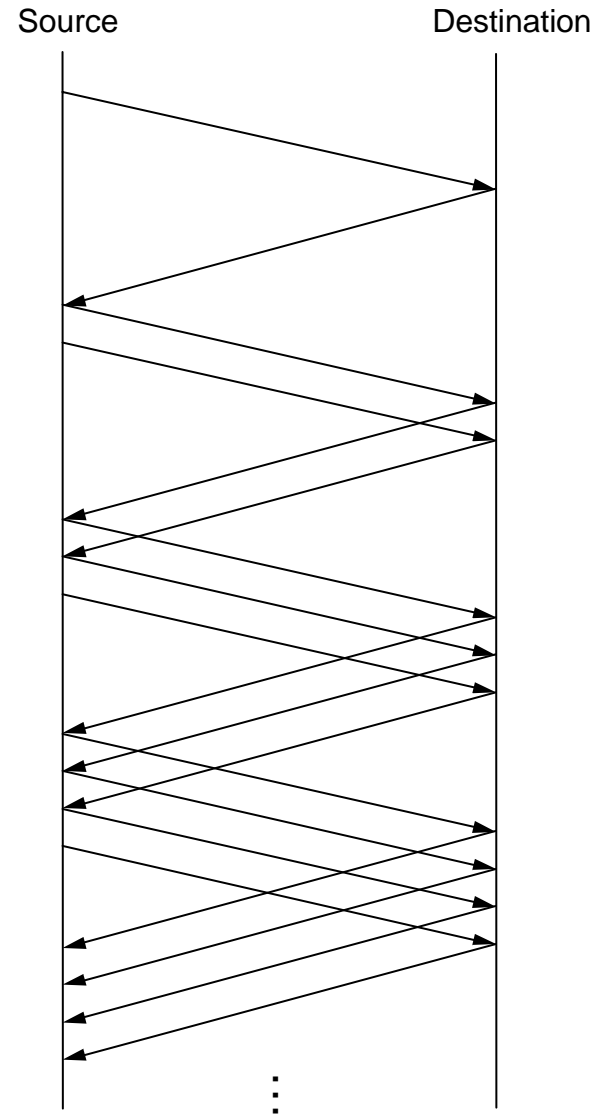
- ◆观察主要原因：包未交付，并导致超时结果，就是由于拥塞丢包，由于传输错误丢包是很少的
- ◆TCP把超时解释为拥塞的信号并降低传输速率
- ◆特别，每次超时发生，**TCP源把原先的CW值的一半作为CW现值**，这就是乘法减少机制的一部分

# 折半减少CW

- ◆虽然CW定义为字节单位，但也容易以包为单位考虑。例如：
- ◆CW现在是16个包，若检测到丢包，则设置CW  $8 = 16/2$ ，再丢一次包将为  $8/2 = 4$ ， $4/2 = 2$ ，最后到  $2/2 = 1$ ；TCP不再允许比1个包更小的CW，或最大段尺寸（MSS）

# 加法式增加CW

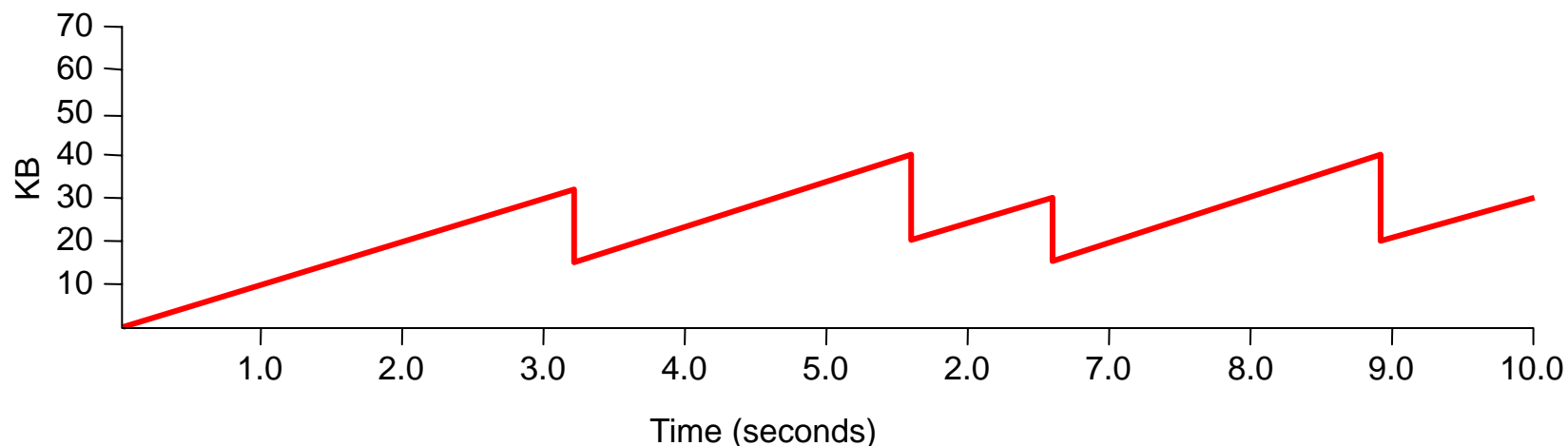
- ◆ 当TCP源每次成功发送CW个包后即最后一个包后一个RTT中每个包都收到了其ACK，则  $CW \leftarrow CW+1$ ，这种线性增加如图2.8所示：



在每个RTT后加1个包

# 实际是锯齿模式加

- ◆ 实际并不是等待全部CW个包的ACK都到后才  $cw \leftarrow cw+1$ ，而是在每一个ACK到达后给CW增加一点，其增量为
  - $\text{Increment} = \text{MSS} \times (\text{MSS}/\text{CW})$
  - $\text{CW} += \text{Increment}$



Thank  
you!

