



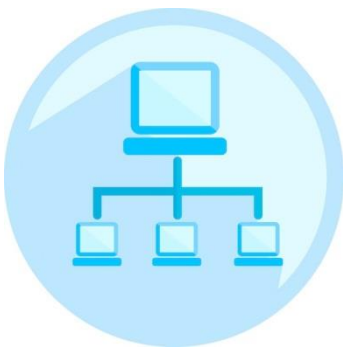
计算机网络



顾 军

计算机学院

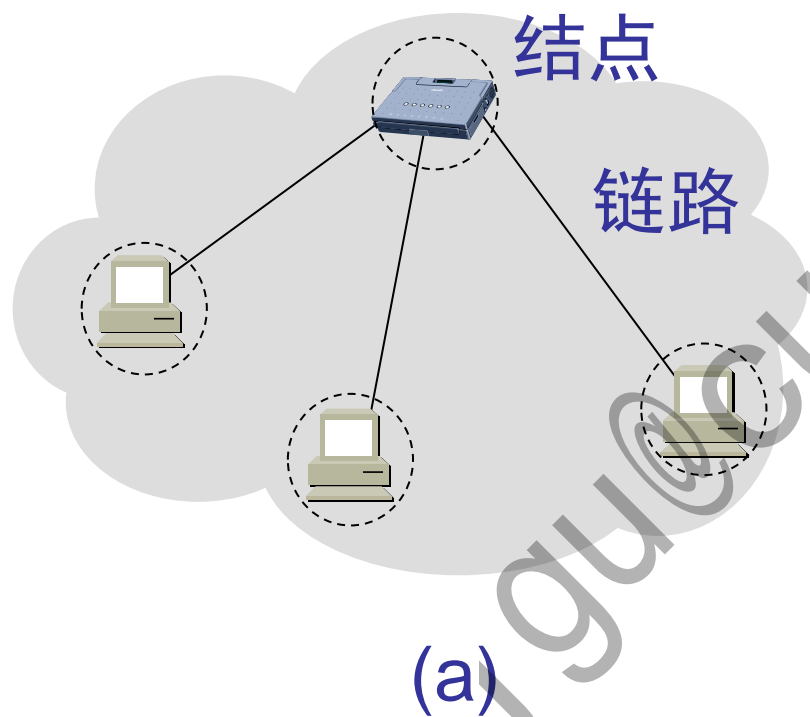
jgu@cumt.edu.cn



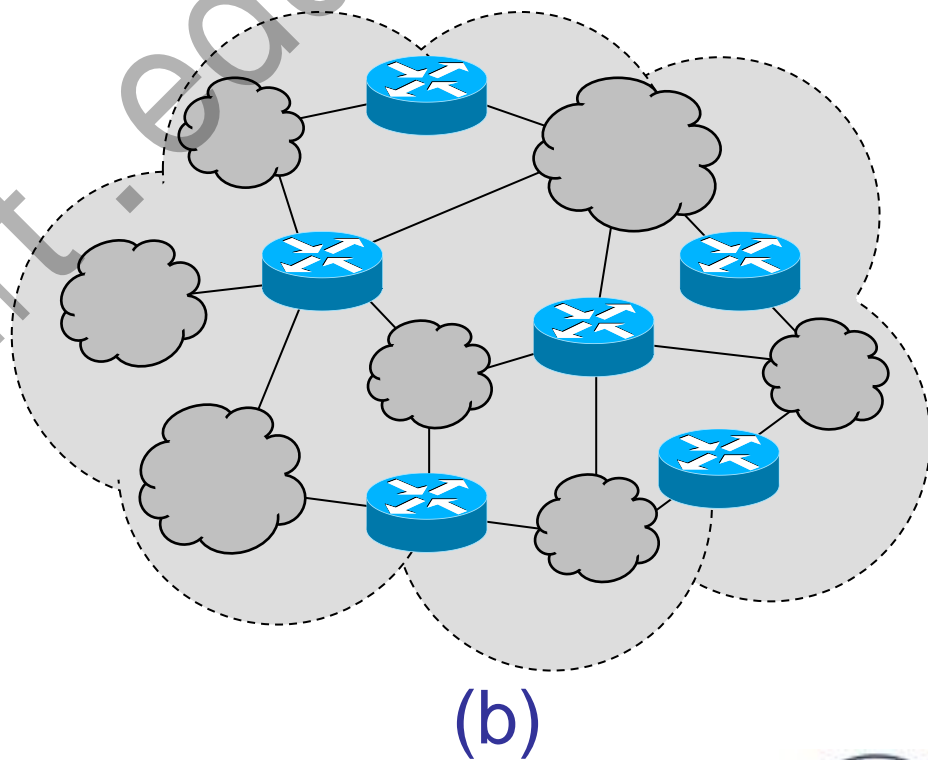


专题1：计算机网络长啥样

网络

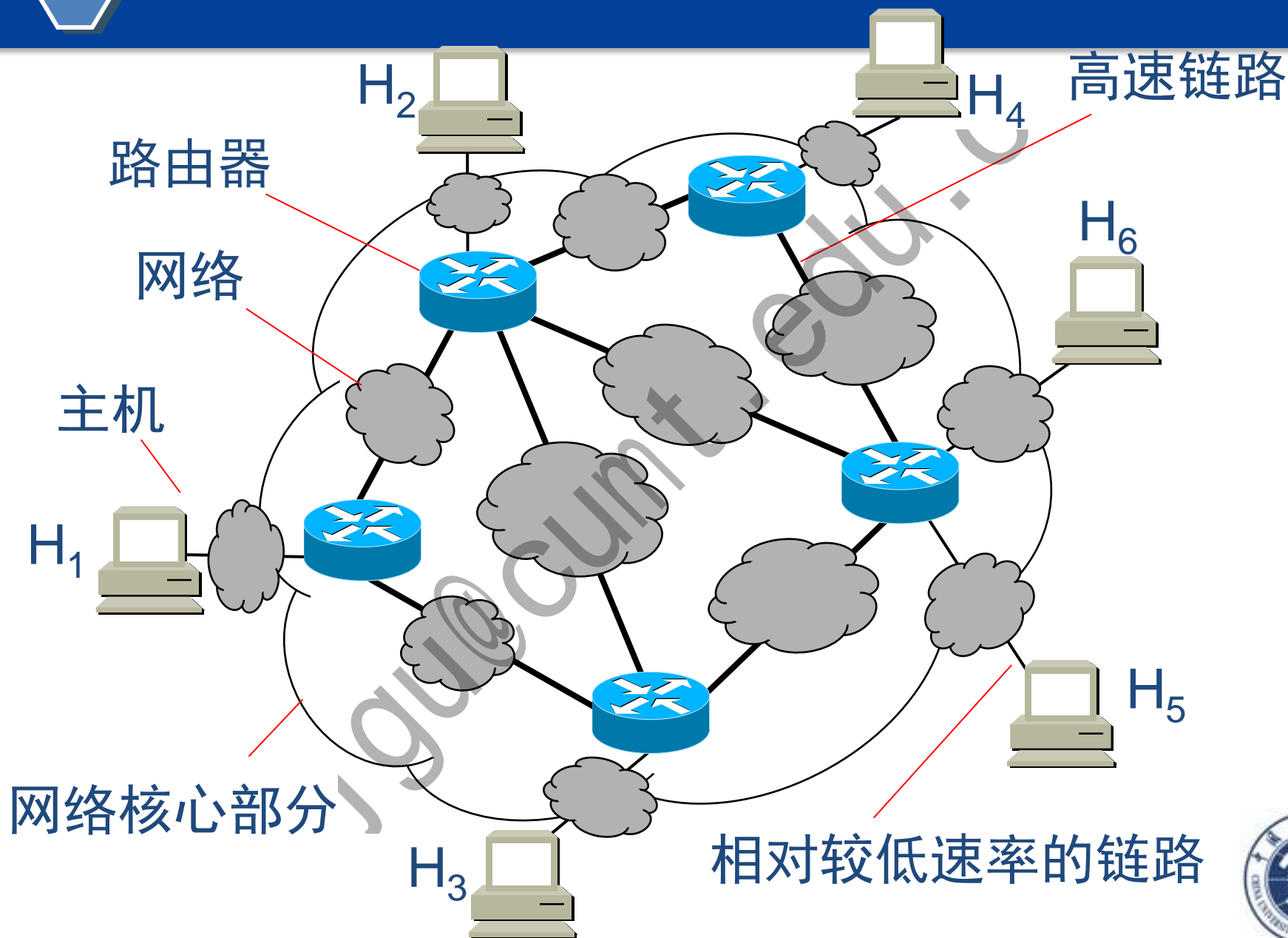


互联网（网络的网络）





Q18: 计算机网络的性能指标?







1. 速率

- **比特** (bit) 是计算机中数据量的单位，也是信息论中使用的信息量的单位。
- Bit 来源于 binary digit，意思是一个“**二进制数字**”，因此一个比特就是二进制数字中的一个 1 或 0。
- **速率**即**数据率**(data rate)或**比特率**(bit rate)是计算机网络中最重要的一个性能指标。速率的单位是 b/s，或 kb/s, Mb/s, Gb/s 等
- 速率往往是指**额定速率**或**标称速率**。





常用的速率单位

- 当数据率较高时，常在b/s的前面加上一个字母。
 - 千(kilo)比每秒，即 kb/s (10^3 b/s)
 - 兆(Mega)比每秒，即 Mb/s (10^6 b/s)
 - 吉(Giga)比每秒，即 Gb/s (10^9 b/s)
 - 太(Tera)比每秒，即 Tb/s (10^{12} b/s)
 - 拍(Peta)比每秒，即 Pb/s (10^{15} b/s)
 - 艾(Exa)比每秒，即 Eb/s (10^{18} b/s)
 - 泽(Zetta)比每秒，即 Zb/s (10^{21} b/s)
 - 尧(Yotta)比每秒，即 Yb/s (10^{24} b/s)
- 注意：通信领域中，只有“1000”使用小写“k”，其余都用大写。





注意数据单位与速率单位的区别

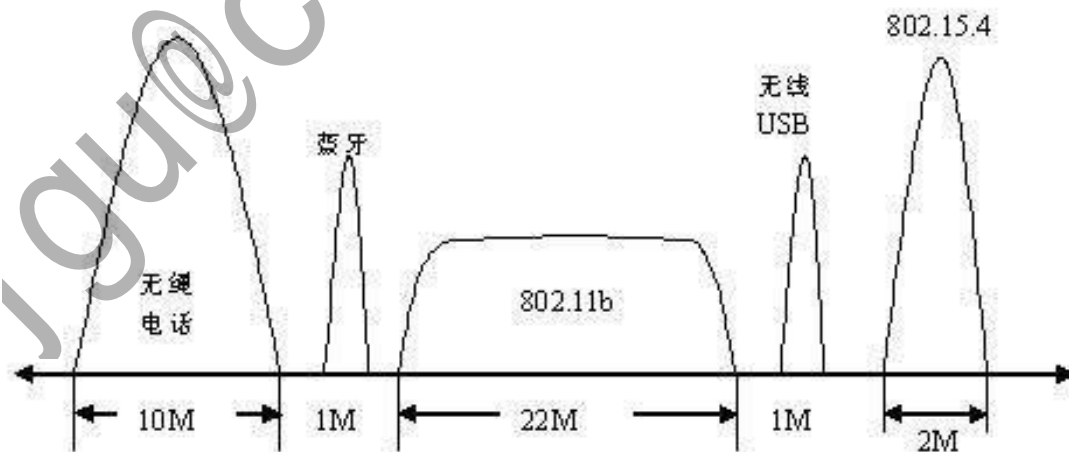
- 在计算机领域中，数的计算使用二进制，数据量往往用字节B(byte)作为度量的单位，通常一个字节代表8个比特(bit)。
 - 千 = K = $2^{10} = 1024$
 - 兆 = M = 2^{20} ,
 - 吉 = G = 2^{30} ,
 - 太 = T = 2^{40} 。
 - 拍 = P = 2^{50} ,
 - 艾 = E = 2^{60} ,
 - 泽 = Z = 2^{70} ,
 - 尧 = Y = 2^{80} 。





2. 带宽

- “带宽” (bandwidth)本来是指信号所占据的频带宽度，单位是赫兹(Hz)。
- 在模拟信号通信线路里，“带宽”表示通信线路允许通过的信号频带范围。
 - 例如模拟语音电话的信号带宽为3400Hz，一个PAL-D电视频道的带宽为8MHz（含保护带宽）。





计算机网络中的带宽

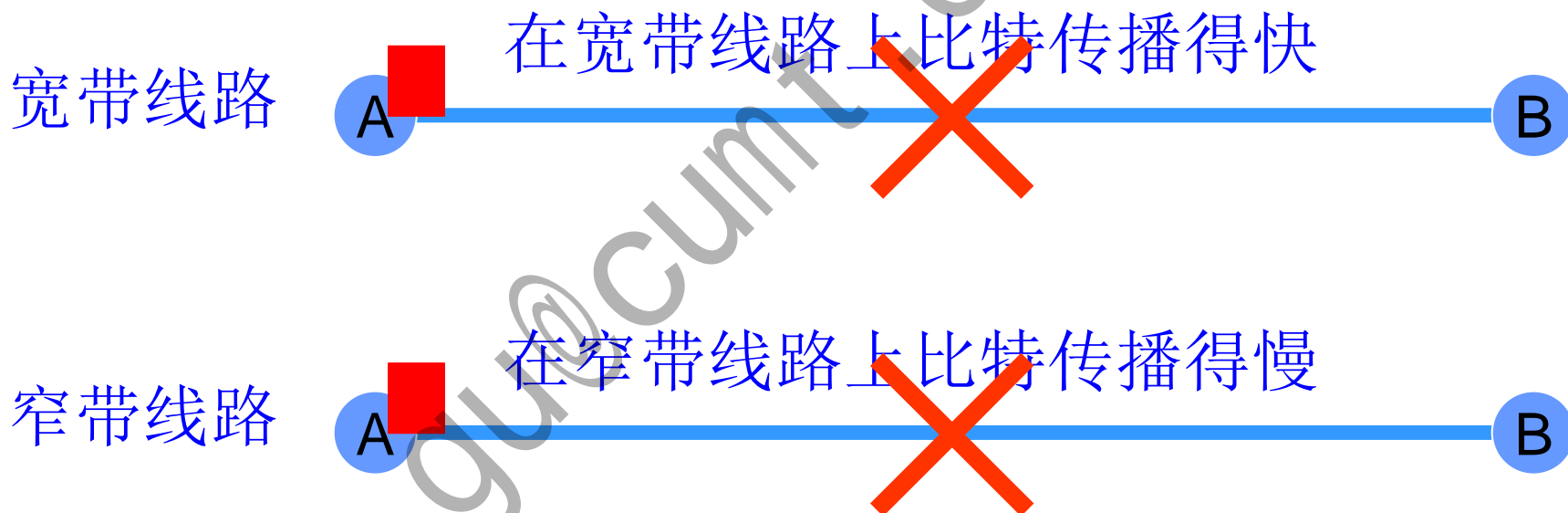
- 数字信号的“带宽”是指单位时间内通信链路能够通过的数据量。
 - 即，数字信道所能传送的“最高数据率”
 - 单位是“比特每秒”，或 b/s (bit/s) 。
 - 例如ISDN的B信道带宽为64Kbps。
- 无论是模拟信号还是数字信号，传输它们都要占用一定的频率资源。
 - 信号的传输速率越高，所占的频带也就越宽，所以高速率通信也叫“宽带通信”。





容易产生的错误概念

- 在高速链路（或高带宽链路）上，比特会传送得更快些。





传播速率

- 传播速率的单位是每秒传播多少千米，是指在某一段传输线路上比特的传播速率。
- 载荷信息的电磁波在通信线路上的传播速率取决于通信线路的介质材料。

$$v = c / n$$

- 其中， c 是真空中光速 (3.0×10^8 m/s)， n 是介质的折射率，即在有媒介的地方小于光速。

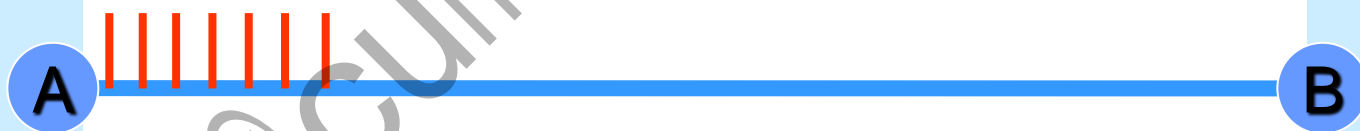




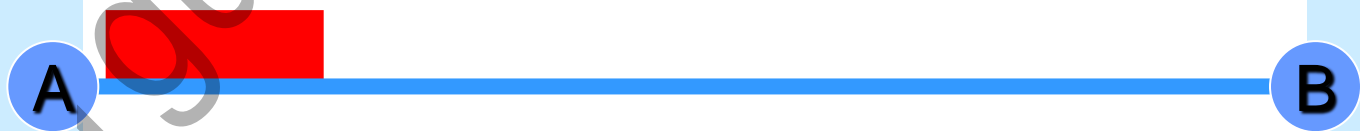
正确的概念

- 宽带线路和窄带线路上比特的传播速率是一样的。
- 宽带线路：每秒有更多比特从计算机注入到线路，即发送速率更高。

宽带线路



窄带线路

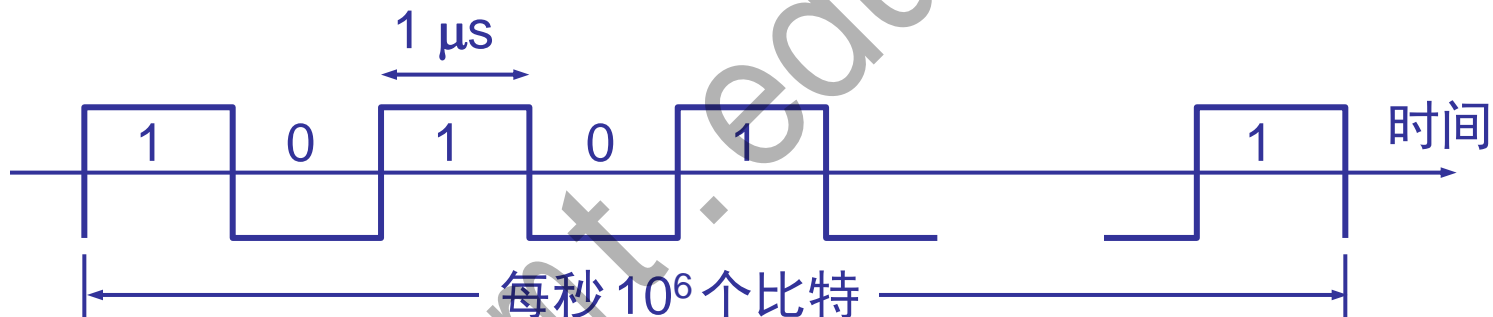




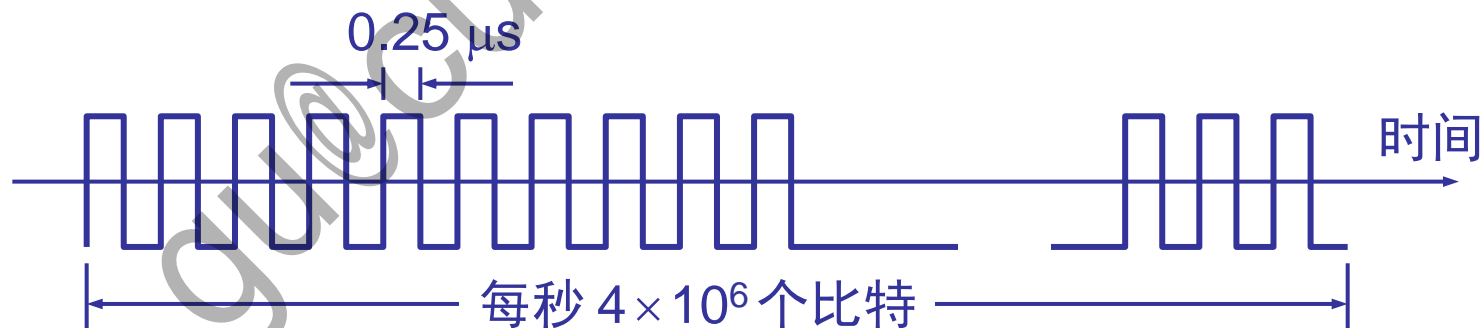
信号宽度和带宽的关系

- 在时间轴上信号的宽度随带宽的增大而变窄。

带宽为
1 Mb/s



带宽为
4 Mb/s





比喻：汽车运货

宽带线路

窄带线路

宽带和窄带线路：车速一样

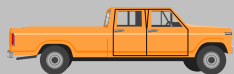
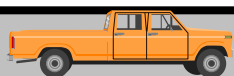
宽带线路：车距缩短

- 通常所说的“**光纤信道的传播速率高**”是指光纤可以提供**很高的信号带宽**，从而可以用更高的速率向光纤信道发送数据。
- 其实，光纤信道的传播速率（约为每秒20.5万千米）实际上比铜线的传播速率（5类线约为每秒23.1万公里）还要略低一点。





另一种错误概念 ——“宽带”相当于“多车道”



多车道公路是并行传输

通信线路上通常都是串行传输

.....100101110100100111010001011010





3. 吞吐量

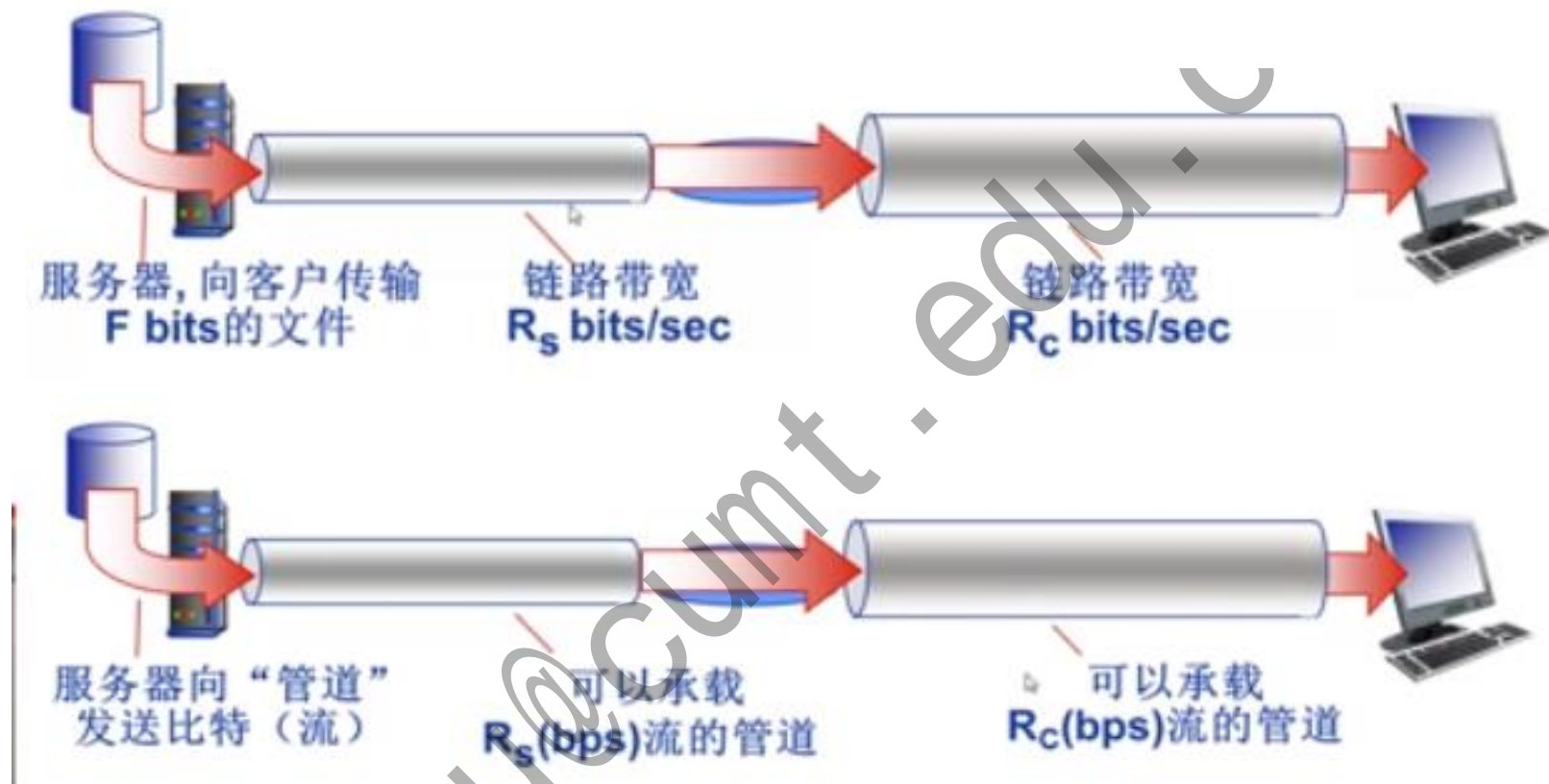
- 吞吐量(throughput)表示在单位时间内通过某个网络（或信道、接口）的实际的数据量。
- 吞吐量作为一种速率，表示在发送端与接收端之间传送数据的速率(比特/秒)，有时还可用于每秒传送的字节数或帧数来表示。
 - 即时吞吐量：给定时刻的速率
 - 平均吞吐量：一段时间内的平均速率





- 吞吐量经常用于对现实世界中网络的一种**测量**，以便知道**实际上**到底有多少数据量能够通过网络。
- 吞吐量受网络的**带宽或额定速率**的限制。
 - 例如，1Gbit/s的以太网的实际吞吐量可能只有100Mbit/s，甚至更低。





若带宽 $R_s < R_c$, 则端到端吞吐量取决于 R_s ;
反之, 取决于 R_c 。

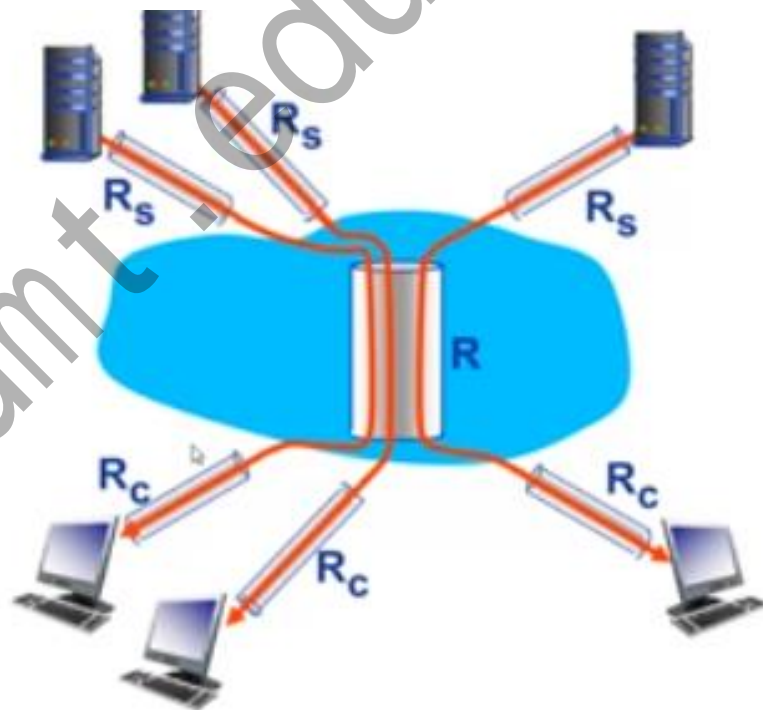




吞吐量的Internet场景

- 瓶颈链路：端到端路径上，限制端到端吞吐量的链路。

- ❖ 每条“连接”的端到端吞吐量： $\min(R_c, R_s, R/10)$
- ❖ 实际网络： R_c 或 R_s 通常是瓶颈



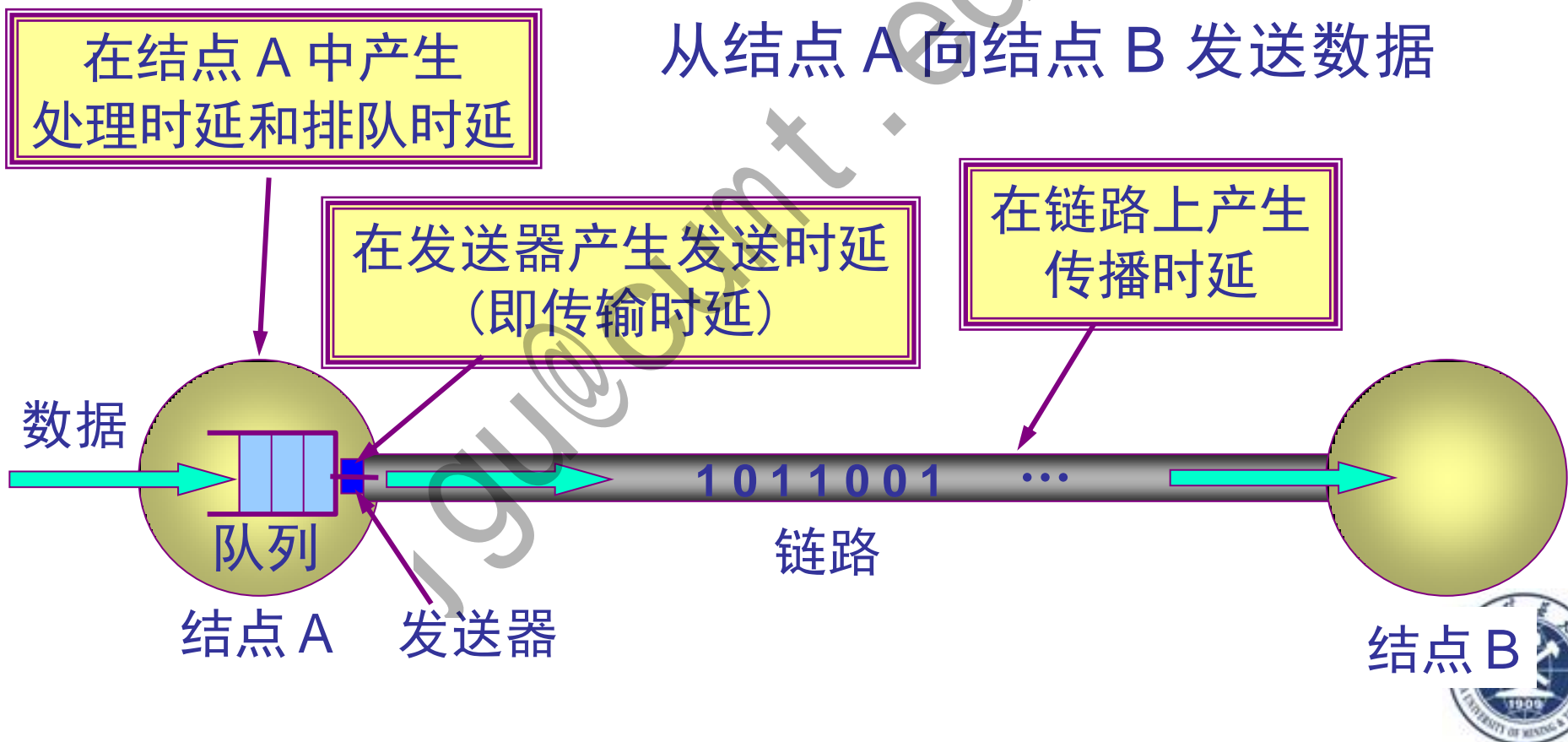
10 条“连接”共享
主干网瓶颈链路 R bits/sec





4. 时延(delay 或 latency)

- 时延是数据(一个报文或分组, 甚至比特)从网络(或链路)的一个端传送到另一端所需的时间。





处理时延(processing delay)

- **处理时延**——交换结点为存储转发而进行一些必要的处理所花费的时间。
 - ✓ 例如，分析分组的首部、从分组中提取数据部分、进行差错检验或者查找适当的路由等，这就产生了处理时延。
 - ✓ 现在一般忽略，因为交换结点处理能力较强





排队时延(queueing delay)

- **排队时延**——结点缓存队列中分组排队所经历的时延。
 - 分组在经过网络传输时，要经过许多路由器，但分组在进入路由器后要先进入输入队列中排队等待处理。
 - 在路由器确定了转发接口后，还要在输出队列中排队等待转发。
 - 排队时延的长短往往取决于网络中**当时的通信量**。当网络的通信量很大时会发生队列溢出，使分组丢失。





发送时延(transmission delay)

- **发送时延(传输时延)**——发送数据时，数据块从结点进入到传输媒体所需要的时间。
 - 也就是从发送数据帧的第一个比特算起，到该帧的最后一个比特发送完毕所需的时间。

$$\text{发送时延}(d_{\text{trans}}) = \frac{\text{数据块长度 } L \text{ (比特)}}{\text{发送速率 } R \text{ (比特/秒)}}$$

- ◆ **发送时延**是主机或路由器发送数据帧所需要的时间，发生在机器内部的发送器中，与传输信道的长度（或信号传送的距离）没有任何关系。
- ◆ 提高数据的**发送速率**可以减小数据的**发送时延**。





传播时延(propagation delay)

- **传播时延**——电磁波在信道中需要传播一定的距离而花费的时间。

$$\text{传播时延} = \frac{\text{信道长度 (米)}}{\text{信号在信道上的传播速率 (米/秒)}}$$

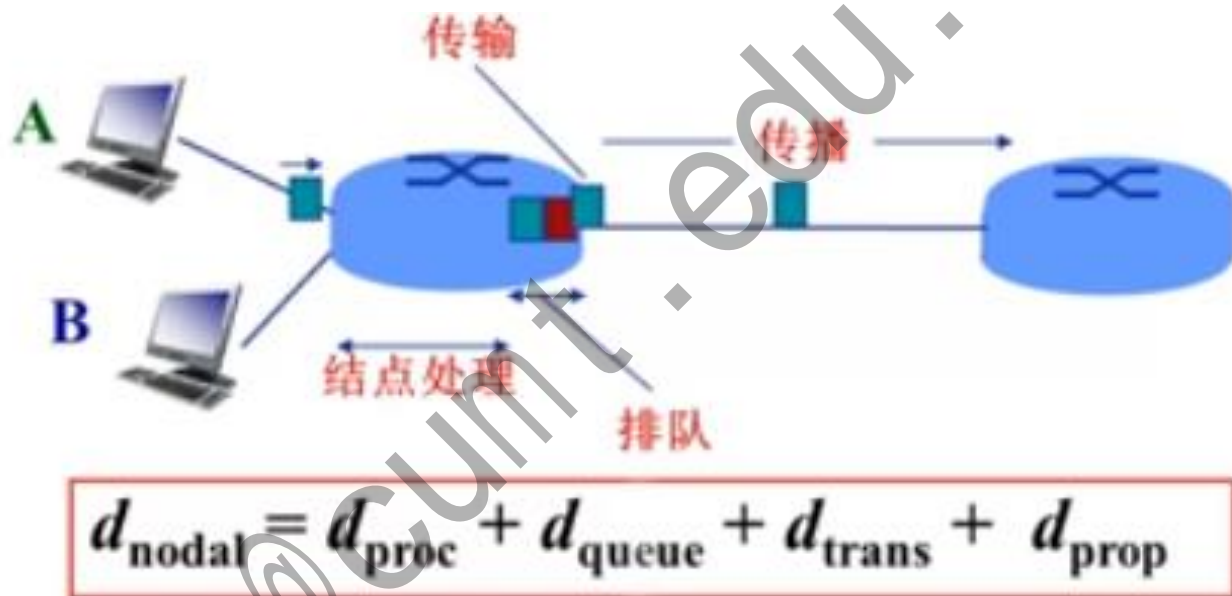
- 传播时延发生在机器外部的传输信道媒体上，与信号的发送速率无关。
- 信号的传输距离越远，传播时延就越大。
- 对于高速网络链路，提高的仅仅是数据的发送速率而不是比特在链路上的传播速率。





总时延(total delay)

总时延 = 处理时延 + 排队时延 + 发送时延 + 传播时延



在总时延中，究竟是哪一种时延占主导地位，必须具体分析。





例1:

一个长度为100MB的数据块在带宽为1 Mbit/s的信道上连续发送，求发送时延。

- 解：数据长度为 $100 \times 2^{20} \times 8 = 838,860,800$ bit
发送速率为 10^6 bit/s
发送时延 = $100 \times 2^{20} \times 8 \div 10^6 = 838.9$ s
- 如果把这个数据块用光纤传送到1000 km远的计算机，其传播时延约为 $10^6 \text{ m} \div (2.05 \times 10^8) \approx 5 \text{ ms}$
- 则总时延 = $838.9 + 0.005 \approx 839.9$ s
- 如果带宽提高到100 Mbit/s，那么总时延 = $8.389 + 0.005 = 8.394$ ，缩小到原有数值的1/100。





例2:



- 报文交换：
 - 报文长度为 M bits
 - 链路带宽（数据传输率）为 R bps
 - 每次传输报文需要 M / R 秒
- 分组交换
 - 报文被拆分为多个分组
 - 每个分组长度为 L bits，拆分的组数为 $G = M / L$
 - 每个分组传输时延为 L / R 秒





报文交换的报文交付时间



- 假设中间的中转节点个数为 n ，则经过的链路个数为 $(n+1)$ ；
- 假设报文交换不考虑传输时延，而且只有一个消息需要传送，则传送时间为

$$(n+1) * M / R$$





分组交换的报文交付时间



- 假设中间的中转路由器个数为 n ，则经过的链路个数为 $n+1$ ；
- 分组交换所分的组数为 $G = M / L$ ，不考虑分组报头；
- 考虑最后一个分组需要等待的时间为 $(G-1) * L / R$ ，最后一个分组传送的时间为 $(n+1) * L / R$ ；
- 分组交换的总时间为 $(G-1) * L / R + (n+1) * L / R$
 $\Rightarrow (G + n) * L / R$
 $\Rightarrow (M / L + n) * L / R \Rightarrow (M + n * L) / R$
 $\Rightarrow (1 + n / G) * M / R$





报文交付时间举例



传输方式	传输时间	缓存
报文交换	$(n+1)*M/R$	M
分组交换	$(1+n/G)*M/R$	M/G

报文长度：
 $M = 7.5 \text{ Mbits}$
 $L = 1500 \text{ bits}$
 $G = M/L = 5000 \text{ 个}$
链路带宽：
 $R = 1.5 \text{ Mbps}$

传输方式	传输时间	缓存
报文交换	15 s	$7.5*10^6 \text{ bits}$
分组交换	$(2/5000+1)*(7.5/1.5)$ $= 5.002 \text{ s}$	$1.5*10^3 \text{ bits}$





5. 往返时间 RTT

- 互联网上的信息不仅单方向传输，还要双向交互。因此，有时需要知道双向交互一次所需的时间。
- 例如，A向B发送数据。如果数据长度是100MB，发送速率是100 Mbit/s，那么

$$\text{发送时间} = \frac{\text{数据长度(b)}}{\text{发送速率(b/s)}} = \frac{100 \times 2^{20} \times 8}{100 \times 2^6} \approx 8.39 \text{ s}$$

- 但是，A不可能连续以100Mbit/s的速率发送数据。
- 假定A只有在收到B的确认信息后，才能继续向B发送数据，并且，如果B正确接收完100 MB的数据后，就立即向A发送确认。
- 这样，A在两次发送数据之间需要等待一段时间。





- 这段等待时间称为往返时间，表示从发送方发送数据开始，到发送方收到来自接收方的确认，总共经历的时间。
 - 这里假设确认信息很短，可忽略B发送确认的时间。
- 如果RTT = 2 s，那么可以算出A向B发送数据的有效数据率。

$$\text{有效数据率} = \frac{\text{数据长度}}{\text{发送时间} + \text{RTT}} = \frac{100 \times 2^{20} \times 8}{8.39 + 2} \approx 80.7 \text{ Mbit/s}$$

- 确实比原来A的数据率100Mbit/s小不少。





- 在互联网中，往返时间还包括各中间结点的处理时延、排队时延以及转发数据时的发送时延。
- 当使用卫星通信时，往返时间 RTT 相对较长，是很重要的一个性能指标。





6. 时延带宽积

- 链路的时延带宽积（BDP）又称为以比特为单位的链路长度。

$$\text{时延带宽积} = \text{传播时延(s)} \times \text{带宽(bit/s)}$$



链路像一条空心管道，时延带宽积表示这个管道的**体积**

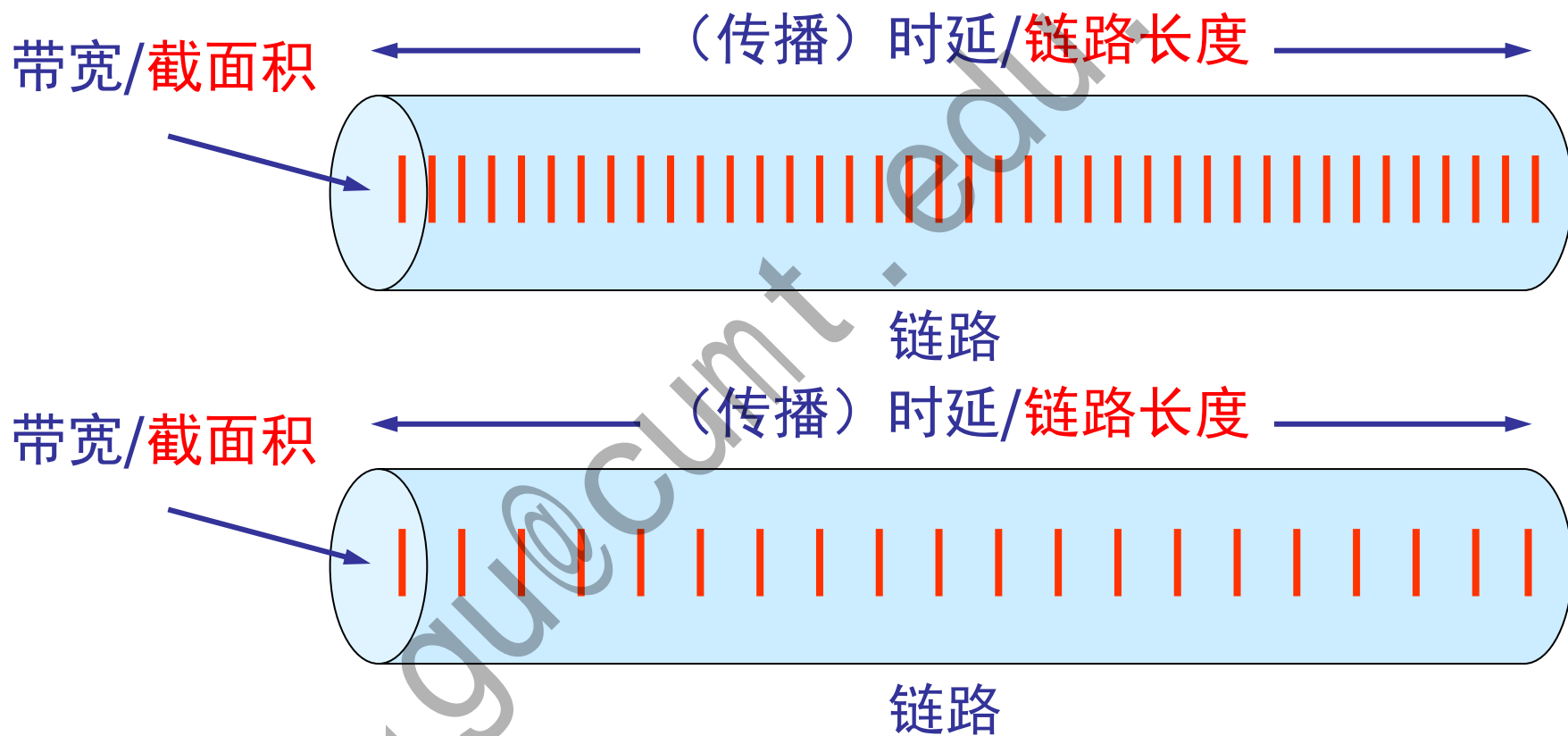
时延带宽积表示当第一个比特到达链路的末端时，链路上存在（容纳了）多少个比特





时延带宽积的物理含义

$$\text{时延带宽积} = \text{传播时延(s)} \times \text{带宽(bit/s)}$$



时延带宽积代表这条管道（链路）的传输能力，只有在代表链路的管道都充满比特时，链路才得到了充分利用。





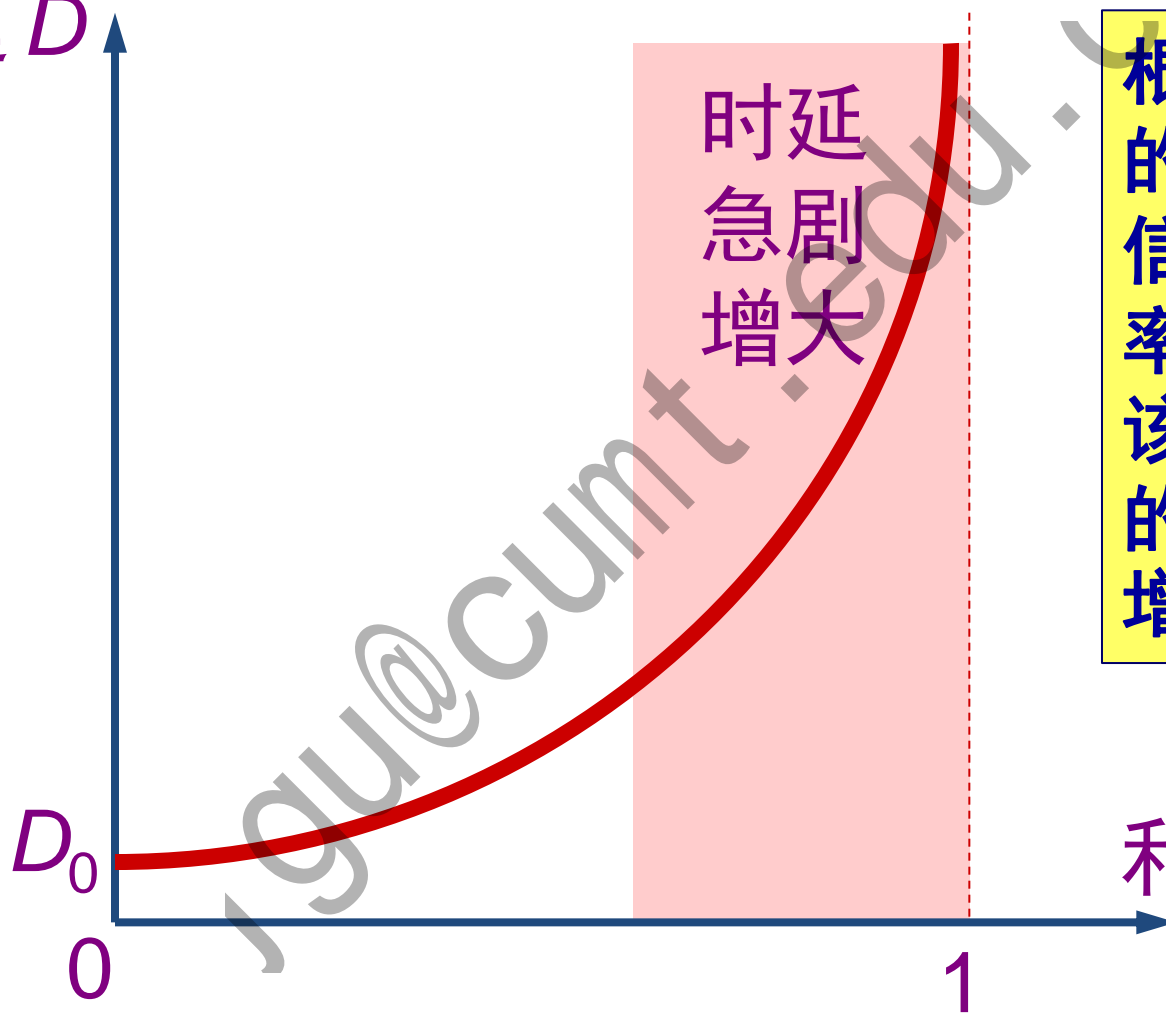
7. 利用率

- 分为信道利用率和网络利用率。
- 信道利用率指出某信道有百分之几的时间是被利用的（有数据通过）。
 - 完全空闲的信道的利用率是零。
- 网络利用率则是全网络的信道利用率的加权平均值。
- 信道利用率要提高，但并非越高越好。
 - 当某信道的利用率增大时，该信道引起的时延也就迅速增加。





时延 D



根据排队论的理论，当信道的利用率增大时，该信道引起的时延迅速增加。





时延与网络利用率的关系

- 若令 D_0 表示网络空闲时的时延， D 表示网络当前的时延，则在适当的假定条件下，可以用下面的简单公式表示 D 和 D_0 之间的关系：

$$D = \frac{D_0}{1 - U}$$

U 是网络的利用率，数值在 0 到 1 之间。





Q19: 计算机网络的非性能特征 ?

- 一些非性能特征也很重要。它们与前面介绍的性能指标有很大的关系。主要包括：
 - 费用
 - 质量
 - 标准化
 - 可靠性
 - 可扩展性和可升级性
 - 易于管理和维护





Q20: 互联网的标准化工作?

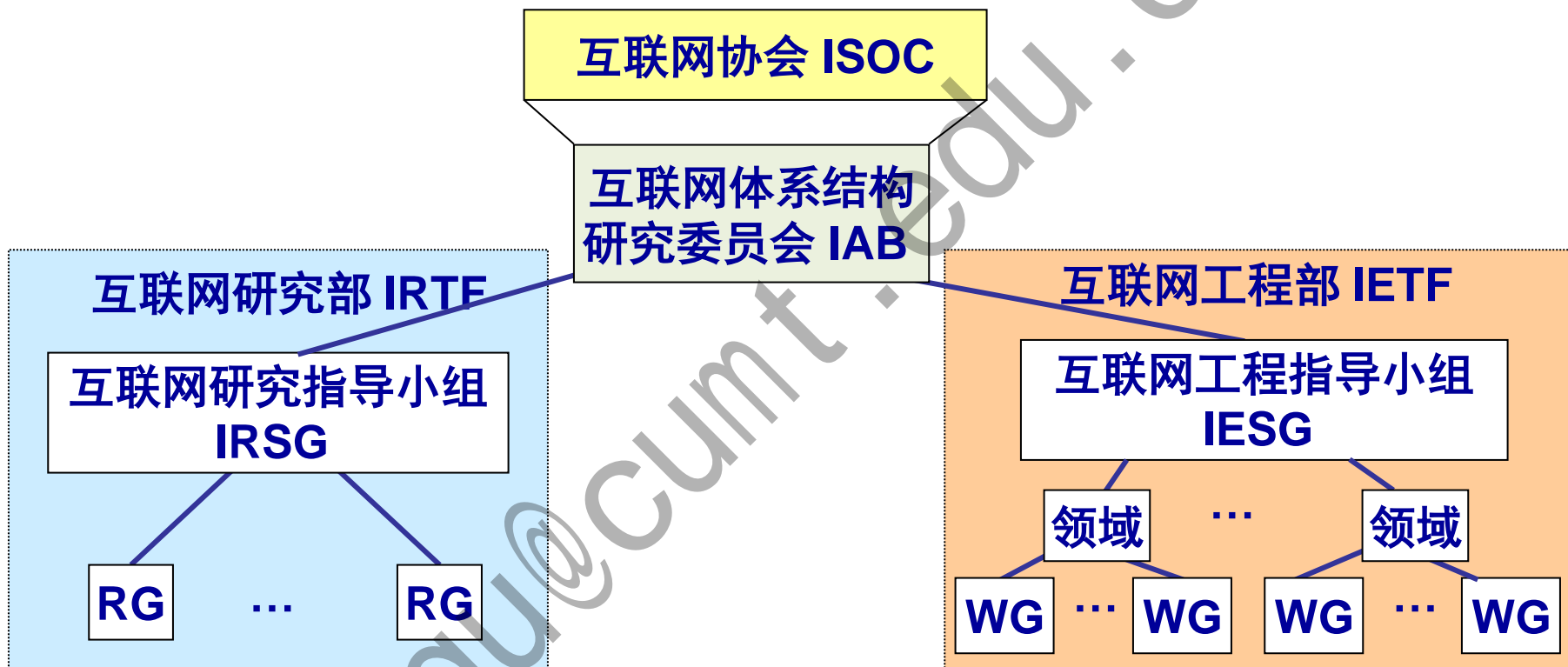
互联网的标准化工作对互联网的发展起到了非常重要的作用。

一流企业定标准
二流企业做品牌
三流企业卖技术
四流企业做产品





互联网协会 ISOC





成为互联网正式标准要经过三个阶段

所有互联网标准都以 RFC 的形式在互联网上发表。

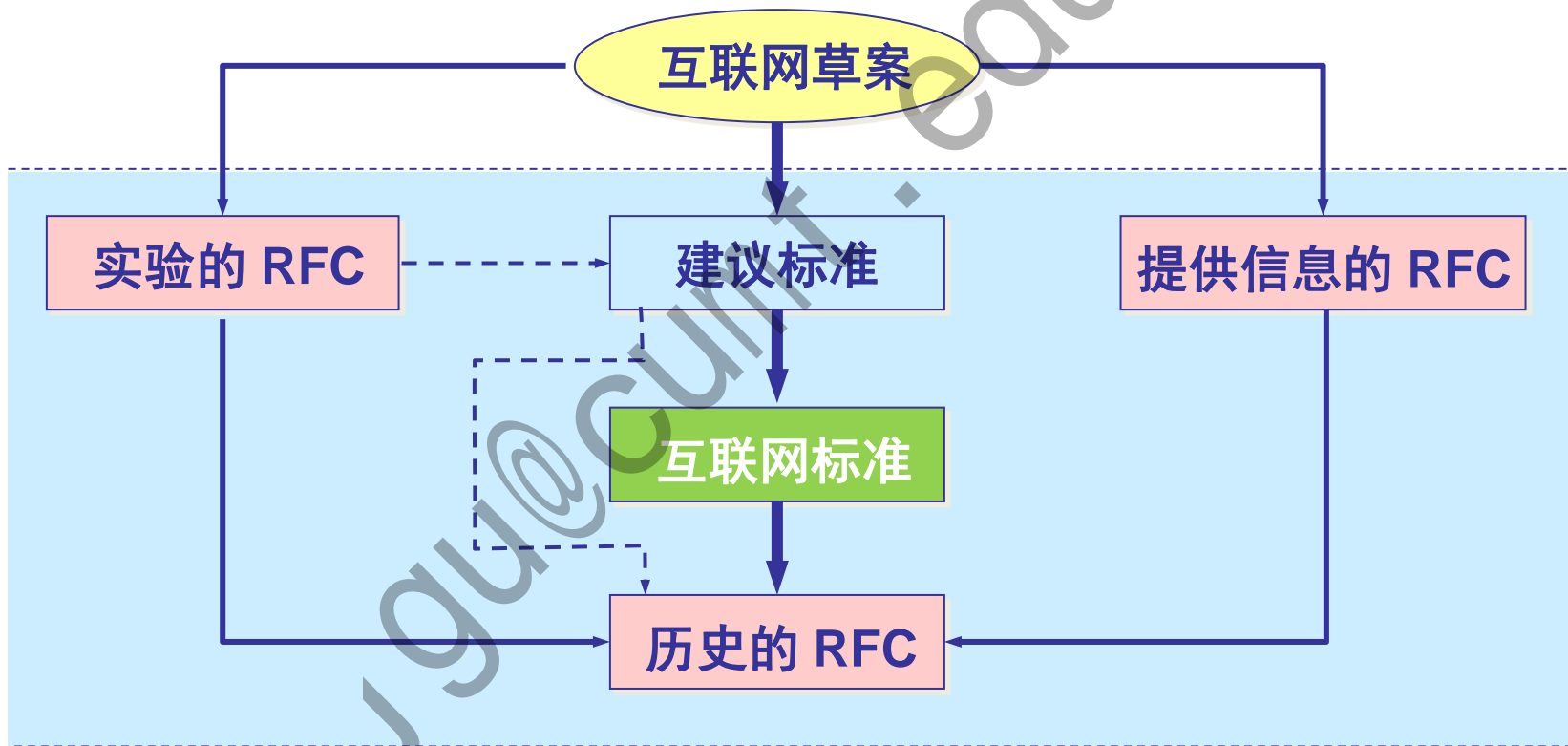
- 互联网草案 (Internet Draft) ——有效期只有六个月。在这个阶段还**不是** RFC 文档。
- 建议标准 (Proposed Standard) ——从这个阶段开始就成为 RFC 文档。
- 互联网标准 (Internet Standard) ——达到正式标准后，每个标准就分配到一个编号 STD xx。一个标准可以和多个 RFC 文档关联。





各种 RFC 之间的关系

除了建议标准和互联网标准这两种 RFC 文档外，还有三种 RFC 文档，即历史的、实验的和提供信息的 RFC 文档。





Q21: IT行业四大定律?

贝尔定律告诉你上网的设备越来越便宜，体积也会越来越小，因此越来越多的用户可以使用各种智能设备上网。

1

麦特卡夫定律说因为有更多的用户上网，所以这个“网”创造的价值越来越大；由于价值越来越大，又有更多的内容和用户上网，网络流量越来越多。

2

3

四大定律

吉尔德定律说没有关系，我发展得更快，主干网的带宽将每6个月增加一倍，所以可以传输这些信息。

4

摩尔定律说传输过来的内容我可以存储、处理和控制。





摩尔定律

- ◆ **摩尔定律**主要描述硬件处理能力的发展规律
 - 1965年，英特尔创始人戈登·摩尔经过长期观察集成芯片(IC)上可容纳的晶体管数目，约每隔18个月便会增加一倍，性能也将提升一倍。
 - ✓ 这个定律是30年前提出来的，在过去的30年里都相当的准确。业内认为摩尔定律在以后的十年或二十年里会继续有效，就是说整个存储，还有计算的发展还会按照指数性的方式前进。
 - 新摩尔定律：全球数据总量每18个月翻番。
 - ✓ 掀起大数据的巨浪





贝尔定律

- ◆ **贝尔定律**讲的是器件价格和体积的规律
 - 如果保持性能不变，每18个月，器件的价格或者体积就会减半，就是说东西越来越便宜。





麦特卡夫定律

◆ 麦特卡夫定律讨论计算和带宽的效应。

- 当我打电话的时候，一个人和另外一个人沟通，这个时候的价值是 **1**；当我做广播的时候，一个对多个，价值是 **N**；当我把 **N** 个用户和 **N** 个装备连接起来的时候，这个价值是按照 **N^2** 的速率增长。
- 由于我有 **N** 个用户，在这个系统中得到 **N^2** 的价值，那么 **N** 就会不断的增长，所以我得到的价值很大。由于 **N** 在增加， **N^2** 会越来越大，**N** 就反馈增长，这就是正向的循环，就是**网络效应**。
- “网络价值同网络用户数量的平方成正比，即 **N** 个联结能创造 **N** 的2次方效益。”





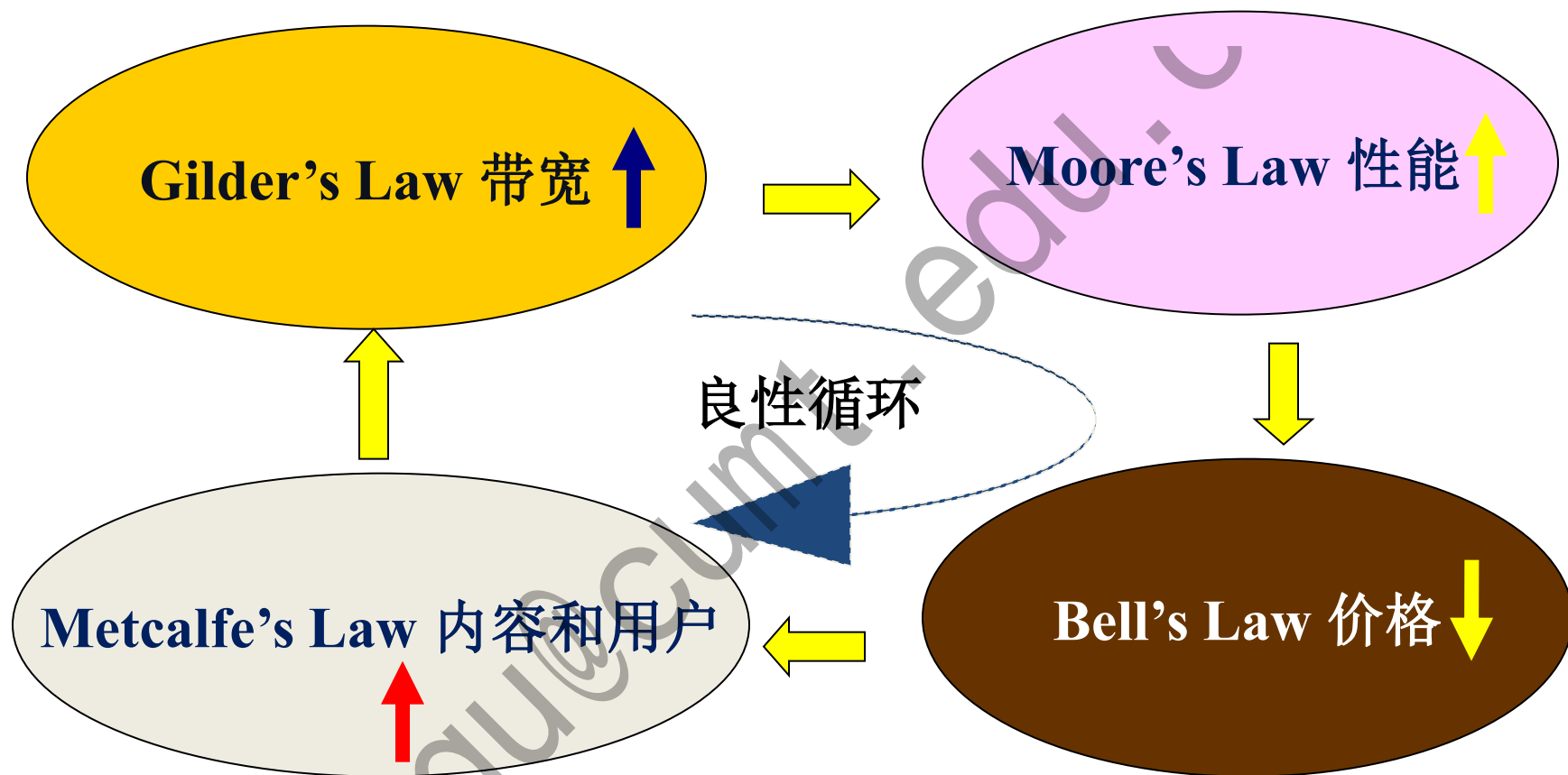
吉尔德定律

- ◆ **吉尔德定律**预测网络带宽的发展规律
 - 在未来25年，主干网的带宽将每6个月增加一倍，其增长速度超过摩尔定律预测的CPU增长速度的3倍。





网络发展遵循四大定律



这就变成一个正向的反馈，使得计算和网络会继续呈现非线性的、指数性的增长。



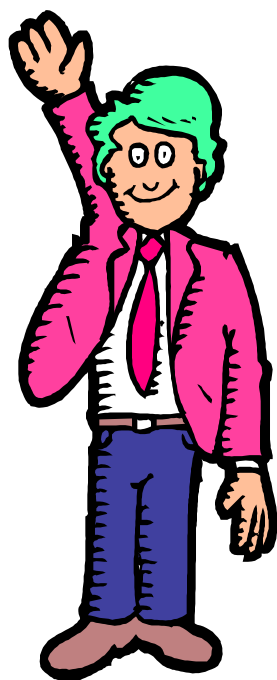


软件的作用不容忽视

- 还有一个最重要的因素，那就是软件的发展。
 - 软件方面本身并没有一个定律，人们也希望找到一个定律，但是一直找不出来。
 - 软件方面其实没有任何的限制
 - ◆ 因为软件是人写出来的，它的限制是人的IQ，是人的**创新能力**；
 - ◆ 它的功能是建立在硬件上，硬件有多快，软件的功能就会有多少，这样本身也是一个相辅相成的关系。

软件定义一切





**THANK
YOU!**

