

本次课的主要内容

第1章 计算机网络和因特网

1.1 什么是因特网

1.2 网络边缘

1.3 网络核心（重点）

1.4 分组交换网中的时延、丢包和吞吐量（难点）

1.5 协议层次及其服务模型（重点）

1.6 面对攻击的网络

1.7 计算机网络和因特网的历史

1.8 小结

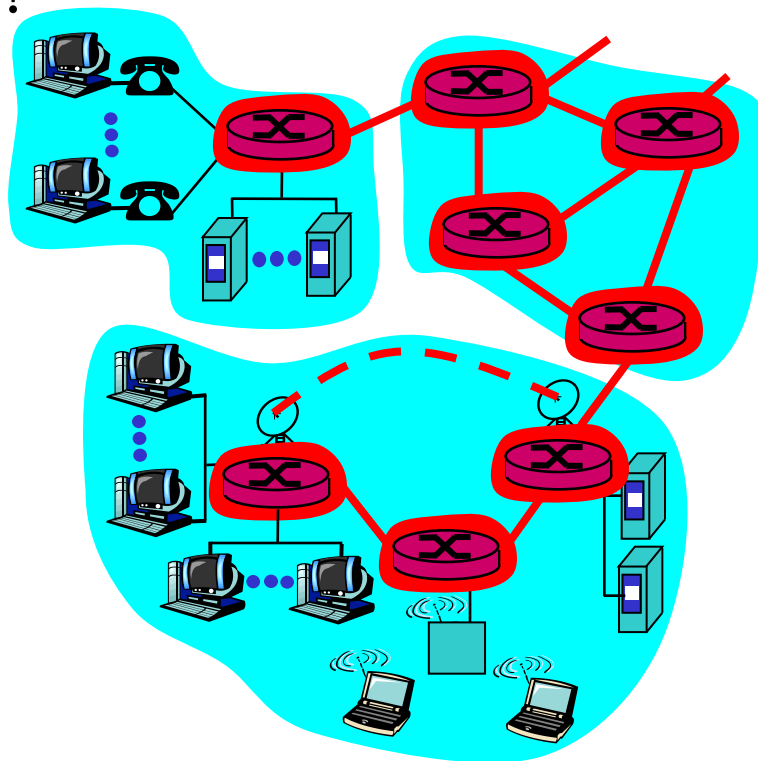
1.3 网络核心

- ❑ 网络内部：连接端系统的**分组交换机和链路形成的网状网络**
- ❑ 基本问题：数据如何通过网络传送？
 - **分组交换(packet-switching)**

1.3.1 分组交换

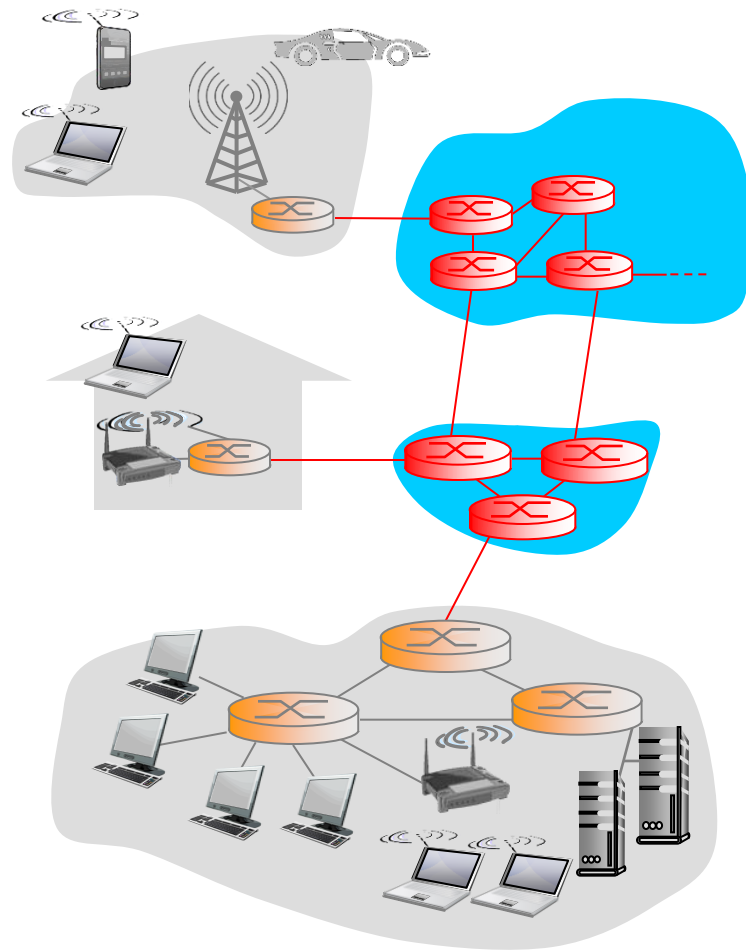
1.3.2 电路交换

1.3.3 网络的网络

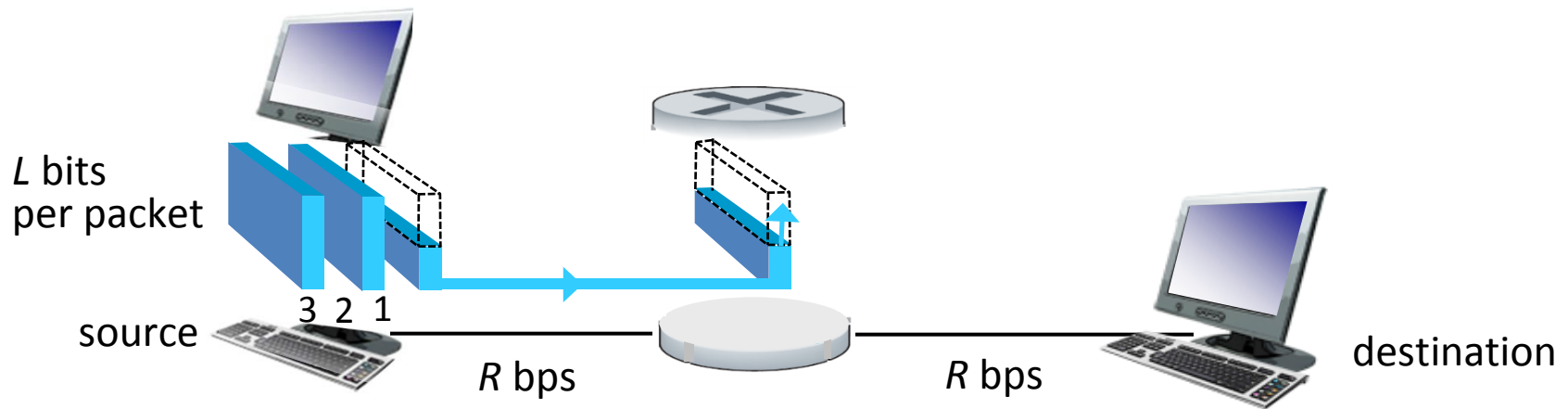


1.3.1 分组交换

- 分组交换(packet-switching): 主机将应用层消息(messages)分割为若干分组(packets)
 - 沿着从源主机到目的主机的传输路径通过每个路由器逐跳的转发分组
 - 每个分组以链路的最大传输速率传输



分组交换：存储转发机制



- ❖ takes L/R seconds to transmit (push out) L -bit packet into link at R bps
- ❖ *store and forward*: entire packet must arrive at router before it can be transmitted on next link
- ❖ end-end delay = $2L/R$ (assuming zero propagation delay)

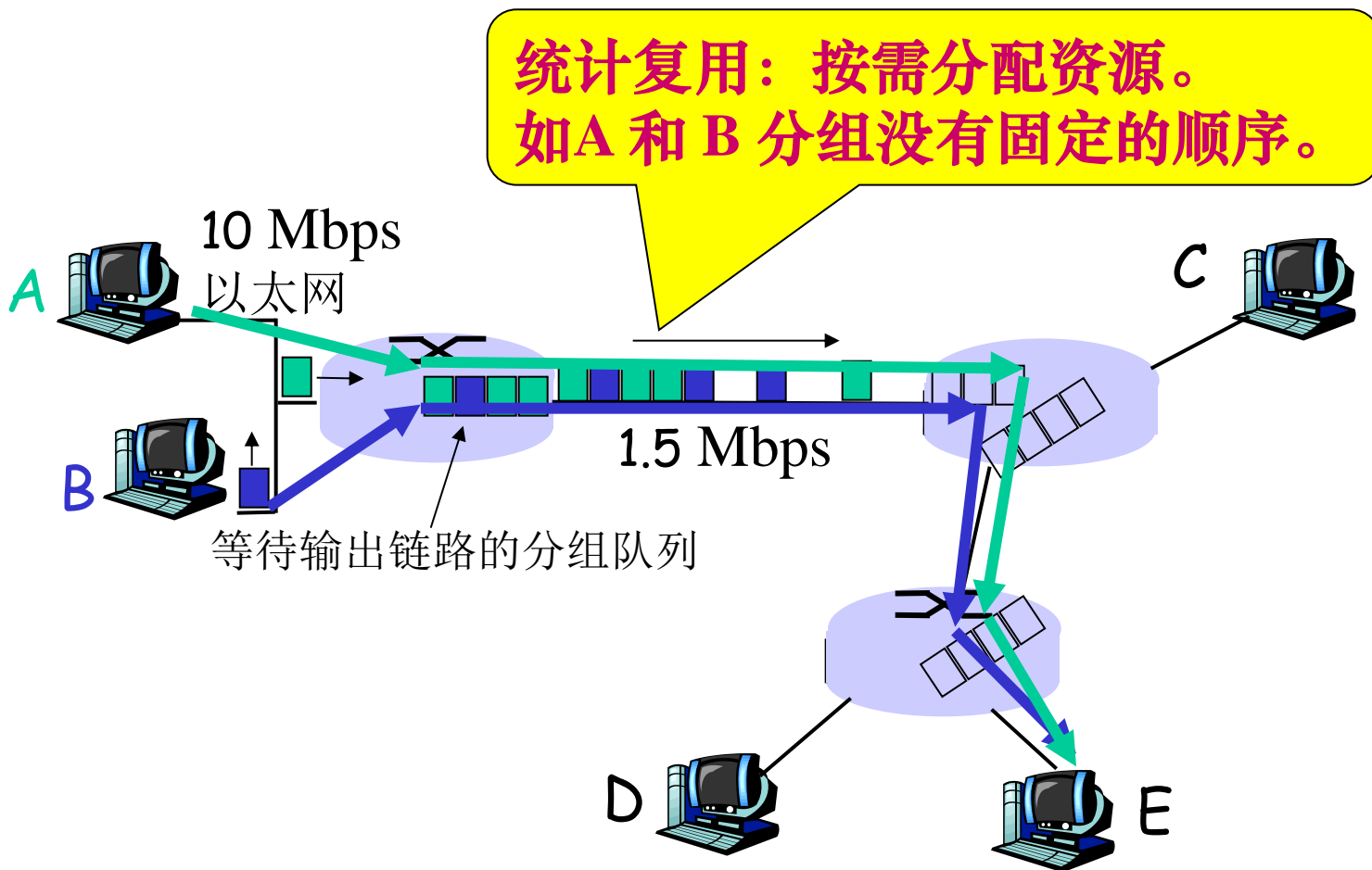
one-hop numerical example:

- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- one-hop transmission delay = 5 sec

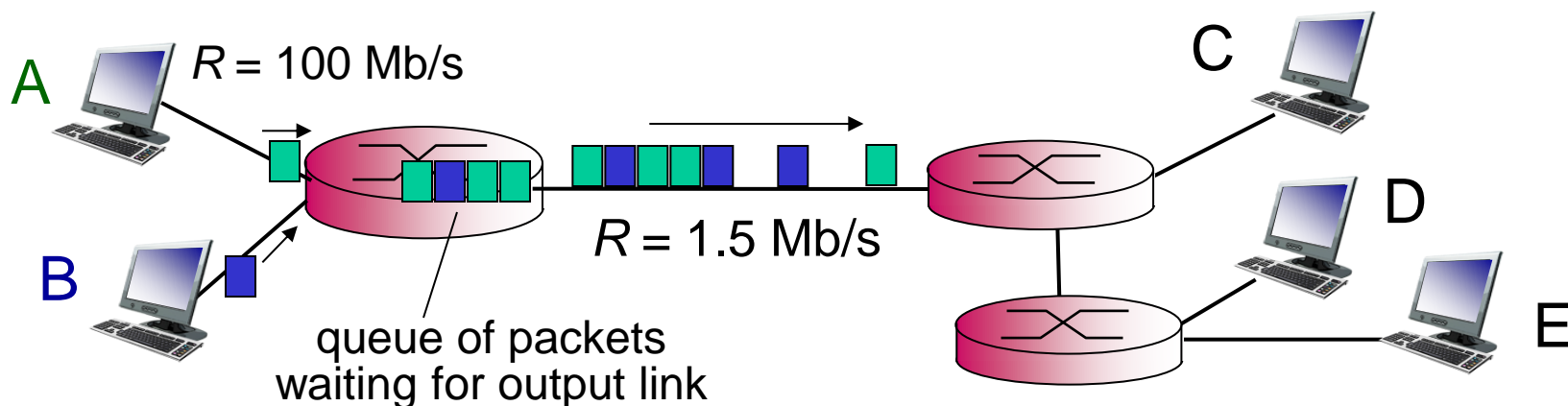
} more on delay shortly ...

例：一个简单的分组交换网络

两对主机通信： $A \rightarrow E$ 、 $B \rightarrow E$ 。所有分组长度相同。



分组交换：排队延时和丢包



排队(queueing)和丢包(loss):

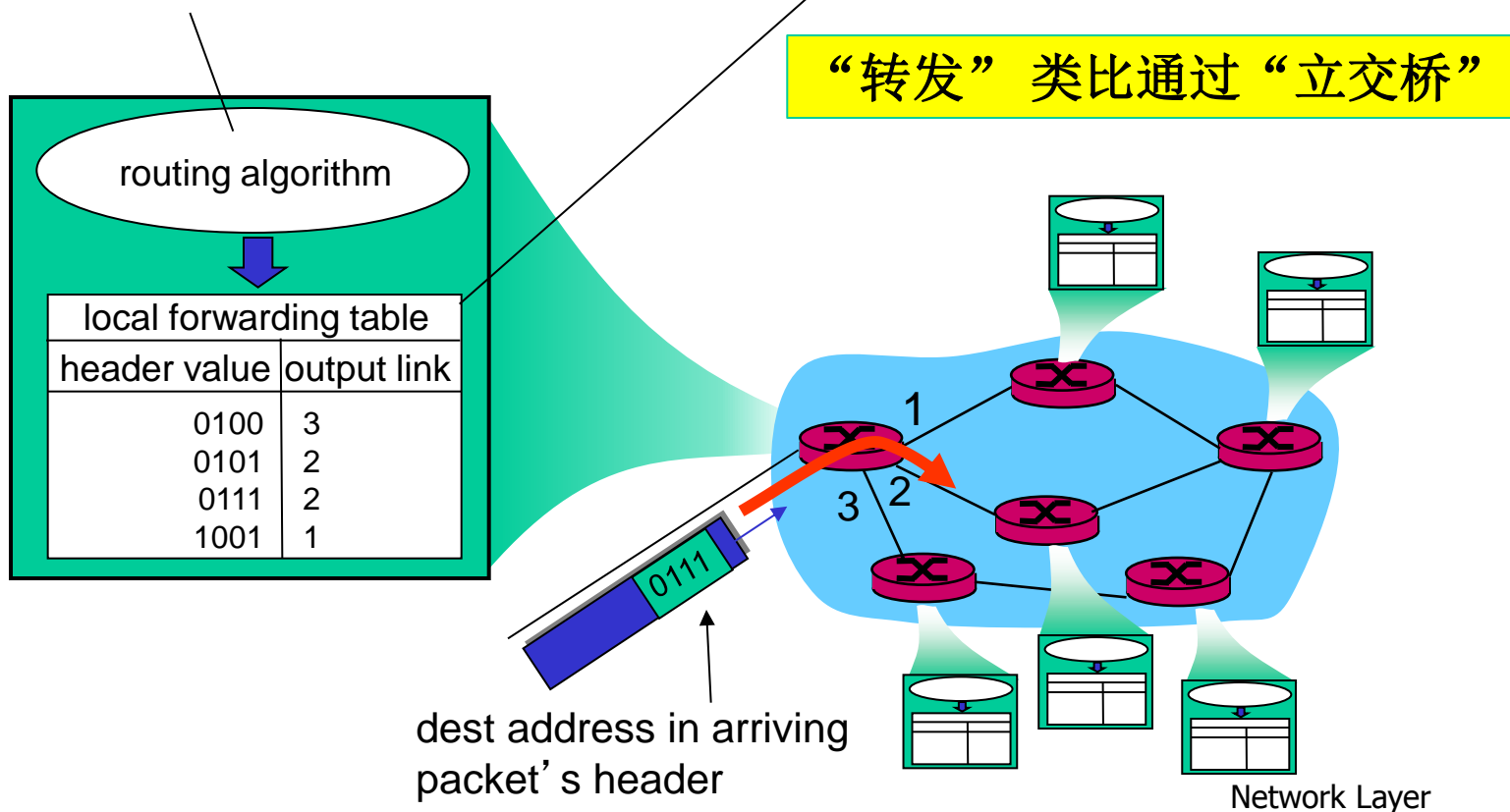
- ❖ 如果在一段时间内，分组到达链路速率超过了链路的传输速率：
 - 分组将排队，在输出链路上等待传输；
 - 如果缓存溢出，则会出现丢包。

两个关键的网络核心功能

路由(routing): 确定分组从源-目的主机的路径

- *routing algorithms*

转发(forwarding): 将分组从路由器的输入端口转到适当的路由器的输出端口



转发表和路由协议

□ 回顾路由器的作用：

- 接收分组，然后根据分组接收终端的地址信息，将分组转发到相应的输出链路上，最终能使分组到达接收终端。

如何选择输出链路？

- 在因特网中，每个分组都会包含接收终端的IP地址，路由器根据IP地址，在自己的转发表中查找这个IP地址对应的输出链路。

如何构建转发表？

每个路由器通过路由选择协议，自动的对自己维护的转发表进行设置。一个基本准则是：从本路由器到达目的终端的距离最短。

1.3.2 电路交换

❑ 电路交换 (circuit switching)

- **预留端到端资源**：预留端系统之间通信路径上所需要的资源 (缓存，链路带宽)，建立连接。
- 发送方**以恒定速率**向接收方**传送数据**。如，电话网络

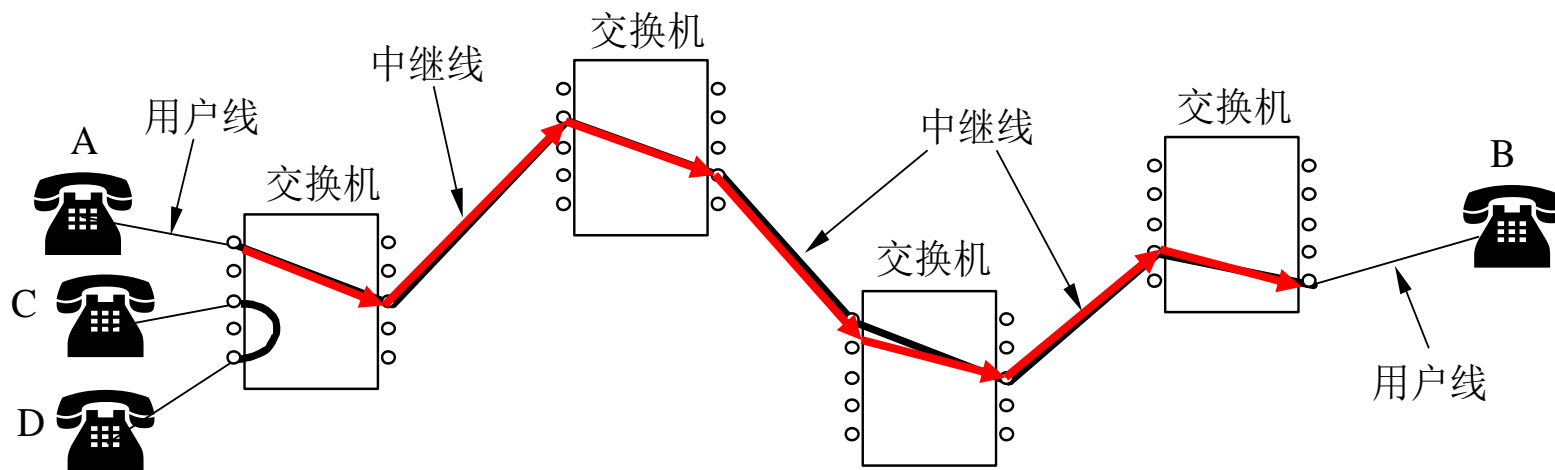
❑ 分组交换(packet switching)

- **不需要资源预留**；
- **按需使用资源，可能要排队等待**：同时有其它分组发送。如，因特网。

电路交换的工作过程

通信双方必须先建立一个**专用的连接（电路）**，一直维持，直到通信结束。如，电话网络。

通话过程：拨号 → 接通 → 通信 → 挂机



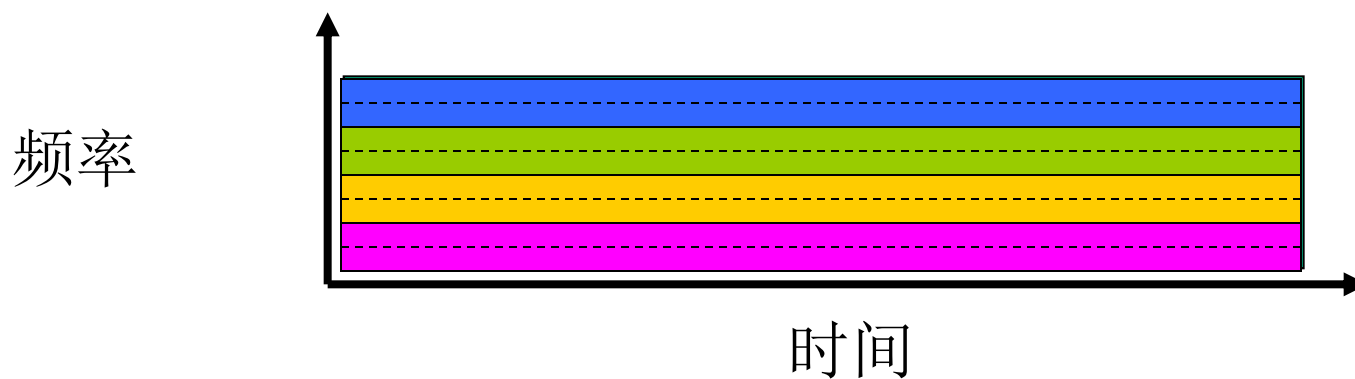
电路交换中的复用技术

- ❑ **多路复用**：在一条传输链路上同时**建立多条连接**，传输数据。
- ❑ **频分多路复用FDM** (frequency-division multiplexing)
 - 按频率划分若干**频段**，每个频段专用于一个连接。
- ❑ **时分多路复用TDM** (time-division multiplexing)
 - 时间划分为固定区间的**帧**，每帧再划分为固定数量的**时隙**，每一个时隙专用于一个连接，用于传输数据。

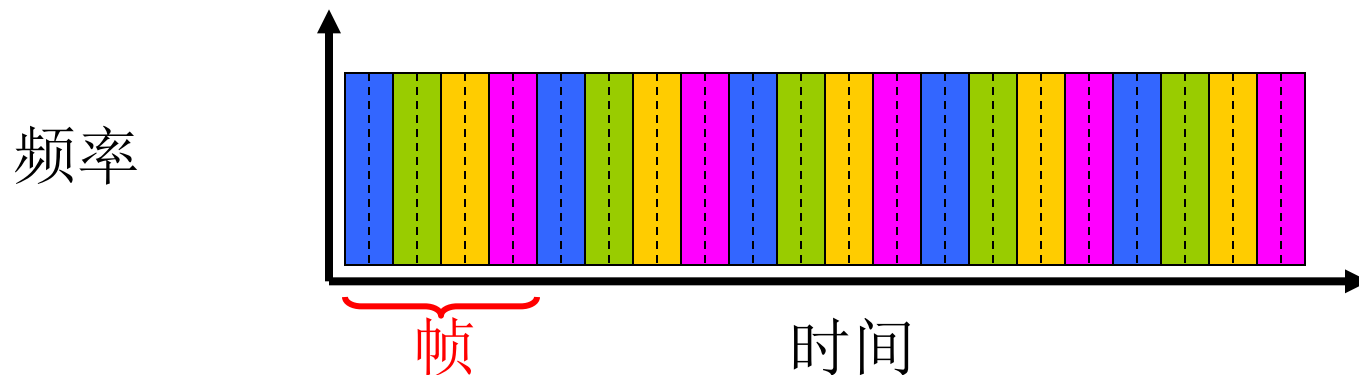
例： 4个用户复用



FDM: 划分4个频段



TDM: 每个帧划分4个时隙



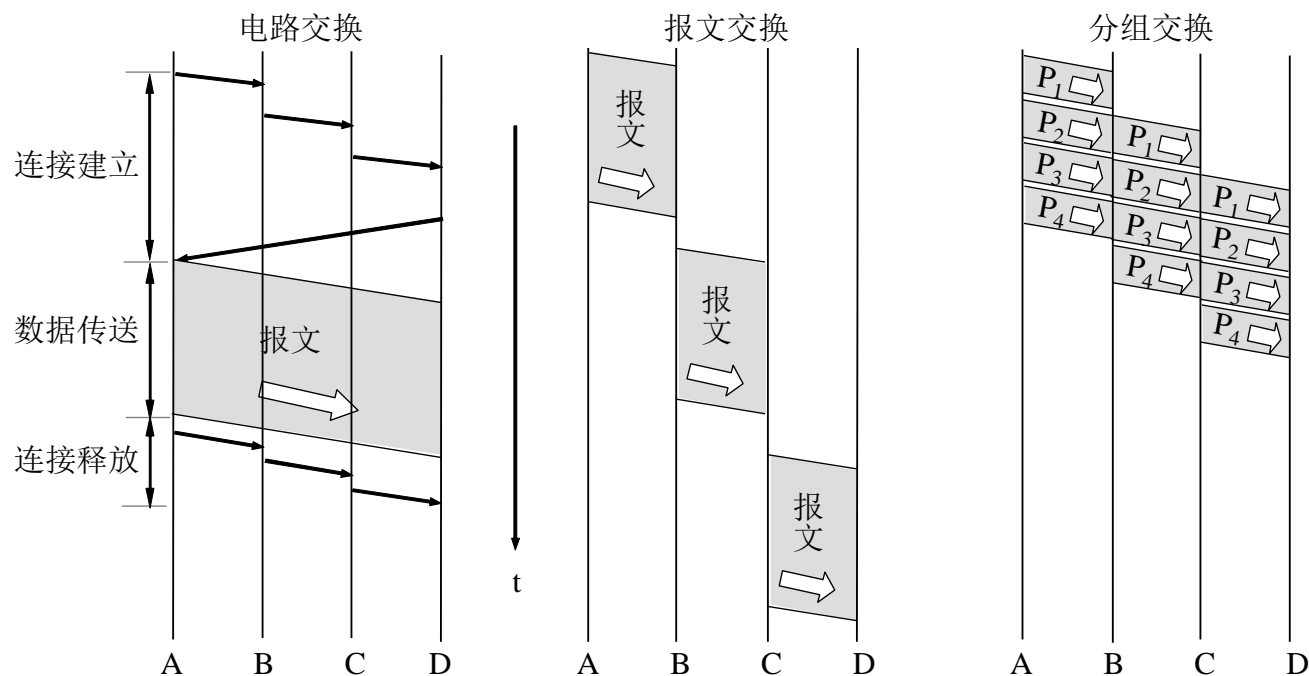
电路交换中的问题

- 效率较低
 - 静默期（无数据传输）专用电路空闲，网络资源被浪费
- 创建端到端电路及预留端到端带宽的过程复杂

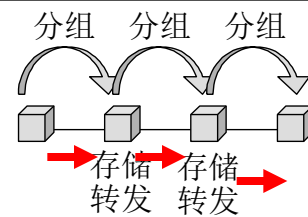
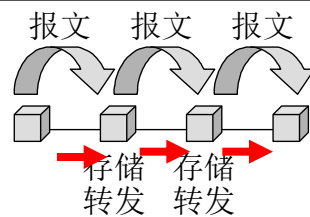
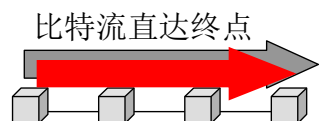
几种交换技术对比

- 电路交换：整个报文的比特流连续地从源点直达终点，好像在一个管道中传送。
- 报文交换：整个报文先传送到相邻结点，全部存储下来后，再转发到下一个结点。
- 分组交换：单个分组（只是整个报文的一部分）传送到相邻结点，存储下来后，再转发到下一个结点。

几种交换技术对比



数据传送
的特点



例：发送一个文件的时间

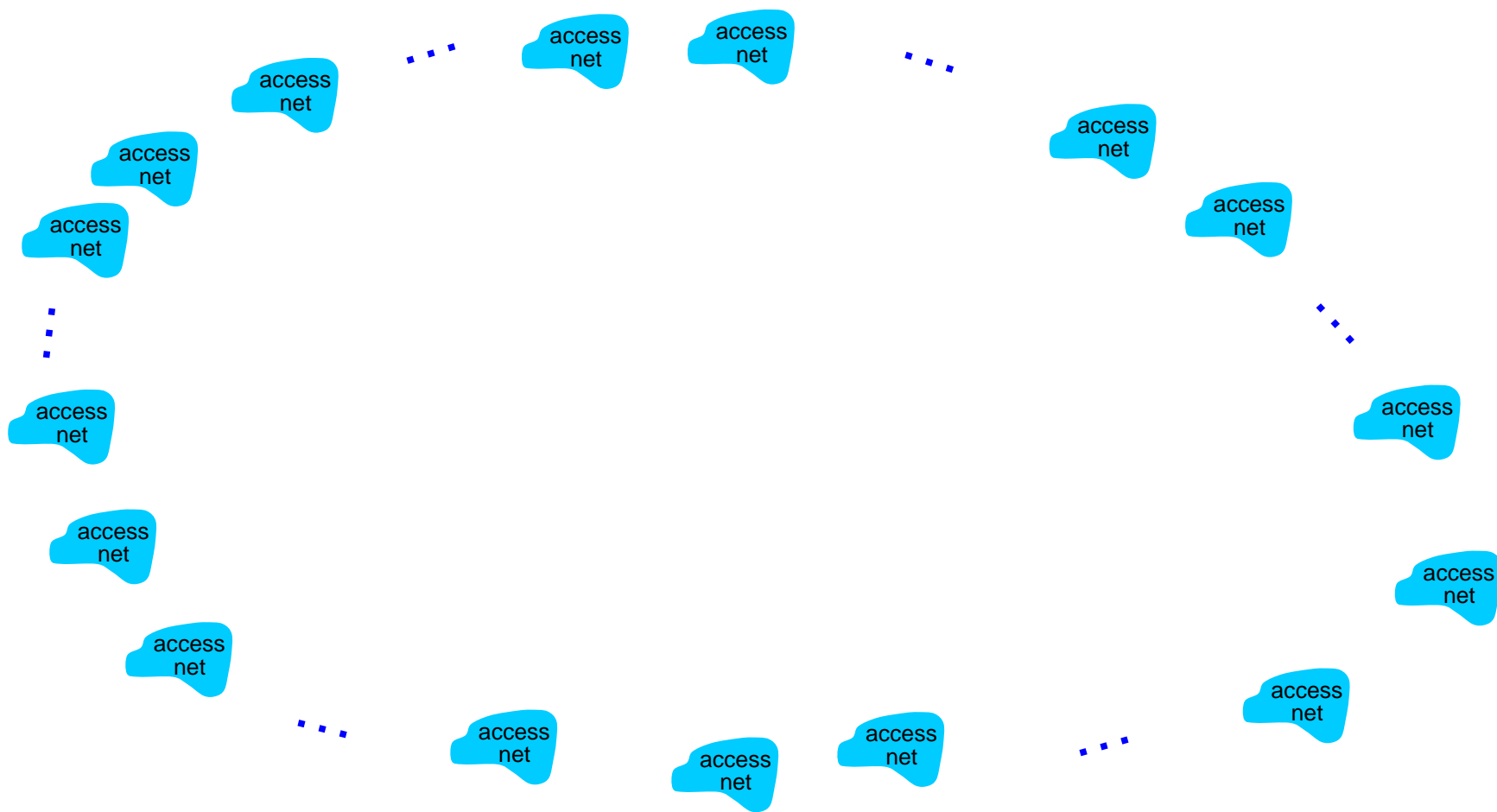
- 从主机A到主机B。已知：
 - 文件长640kb，链路传输速率是1.536 Mb/s
 - 每条链路使用具有24个时隙的TDM
 - 创建端到端电路需500ms
- 发送一个文件时间=创建电路时间+文件传输时间
 - ✓ 文件传输时间：文件长/电路的传输速率
 - ✓ 每条电路的传输速率：链路传输速率/时隙数
$$(1.536 \text{ Mb/s}) / 24 = 64 \text{ kb/s}$$
 - ✓ 文件传输时间： $640 \text{ kb} / (64 \text{ kb/s}) = 10 \text{ s}$
 - ✓ A到B的总发送时间： $0.5 \text{ s} + 10 \text{ s} = 10.5 \text{ s}$

1.3.3 网络的网络

- ❑ 终端系统通过ISPs(因特网服务提供商)与因特网相连
 - 住宅的、公司的、大学的ISPs
- ❑ 接入服务提供商必须相互连接
 - 才能使任意两个终端能够相互发送分组
- ❑ **最终形成非常复杂的网络的网络—因特网**
 - 演进是由经济和政治策略驱动的
- ❑ 让我们逐步地描述当前因特网的结构

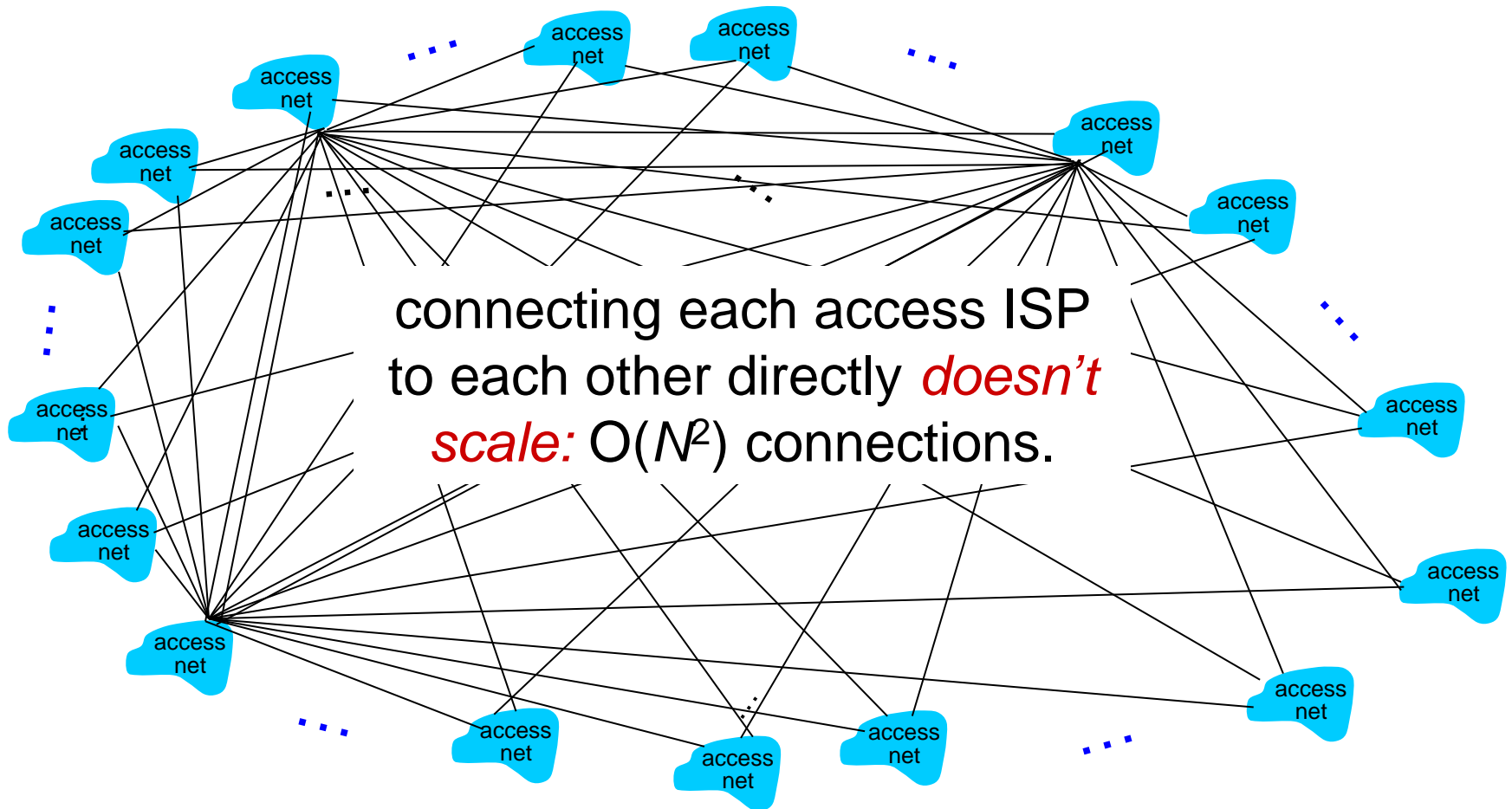
1.3.3 网络的网络

问题：如果有数百万个接入ISPs，应该怎么互联这些ISPs？



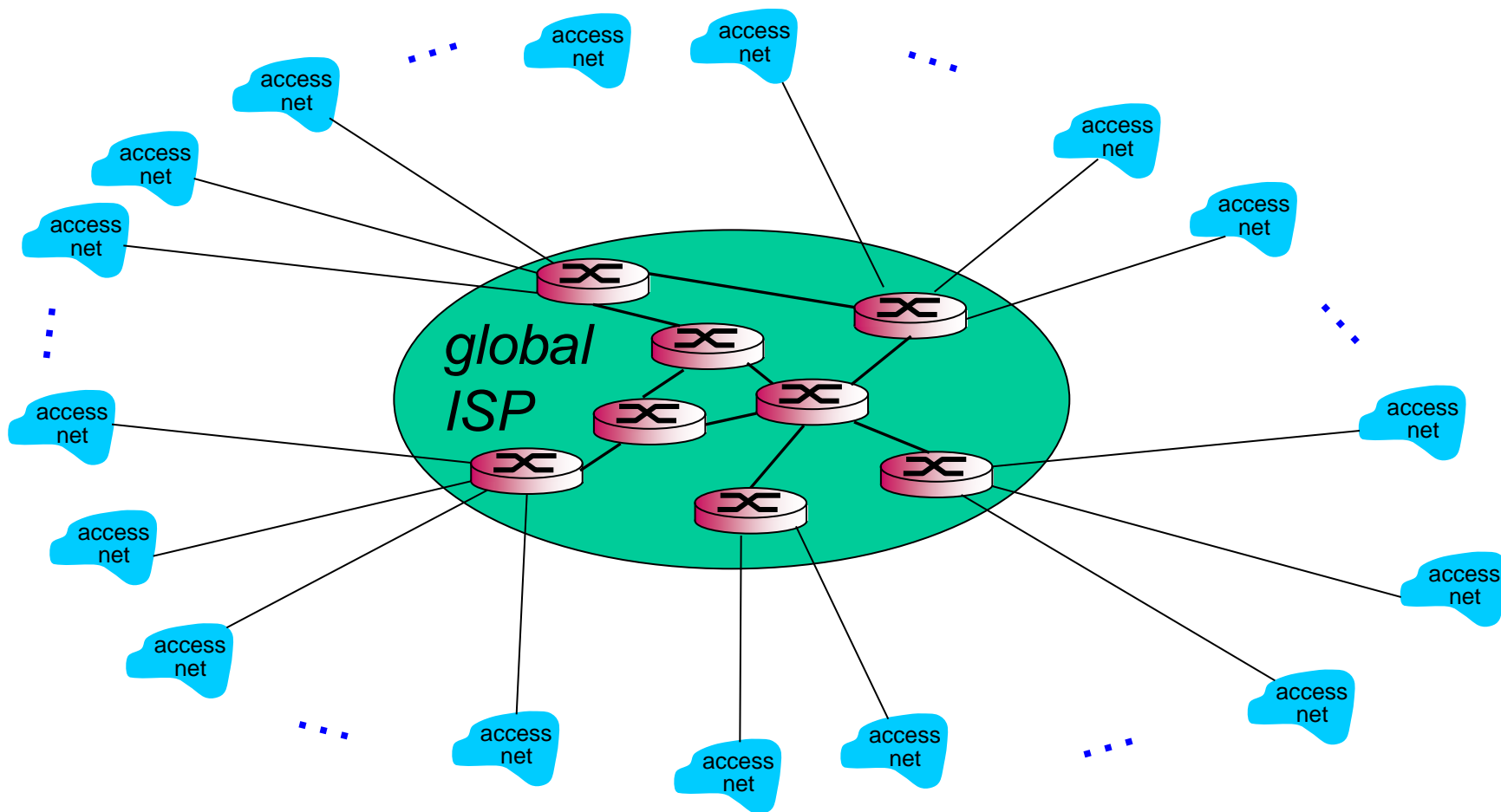
1.3.3 网络的网络

选项： 两两互连所有的ISPs？ 复杂度太高



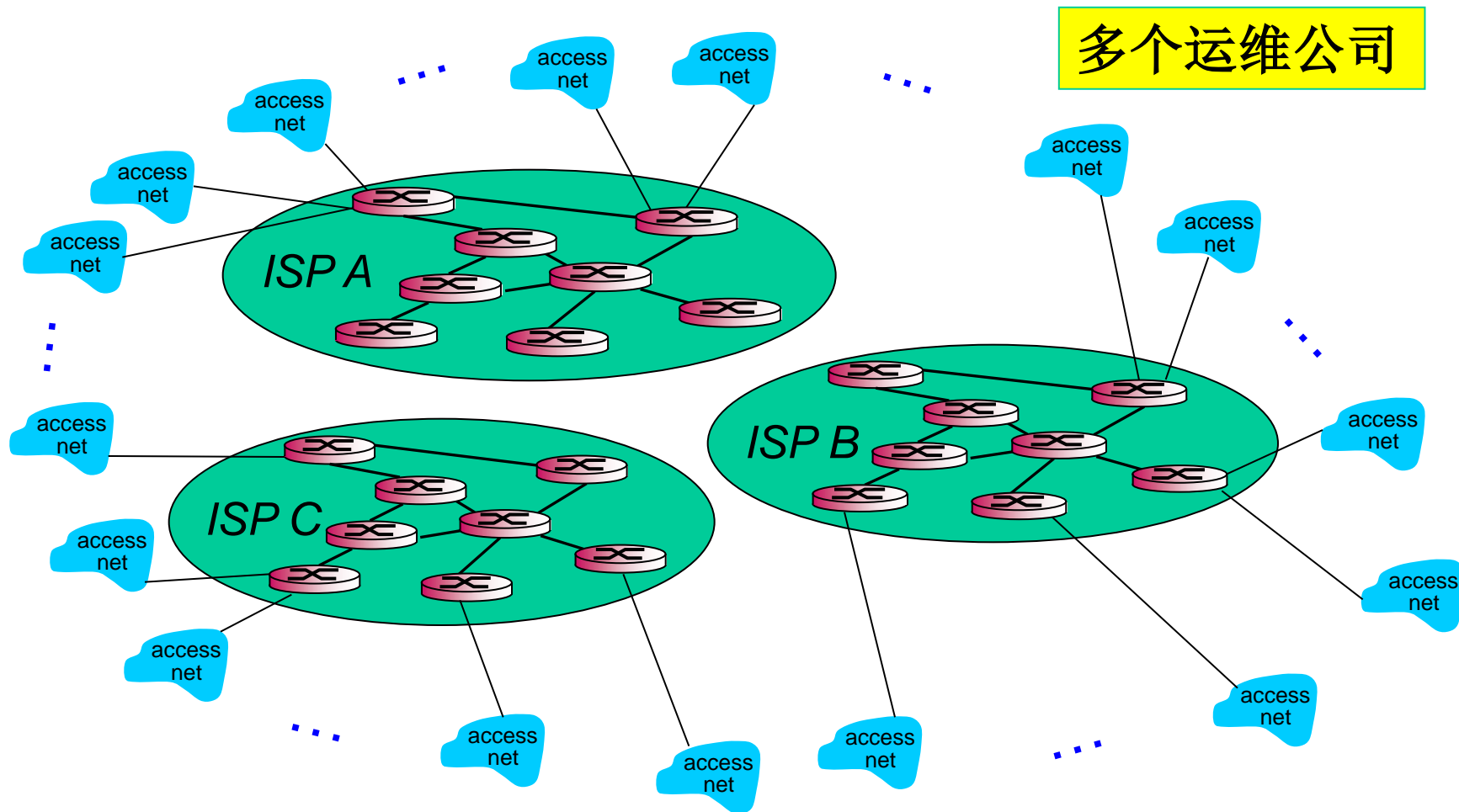
1.3.3 网络的网络

选项：将每个接入ISP连接到一个全局的转发(*transit*)ISP中？
客户方ISP和提供方ISP必须有经济合同。



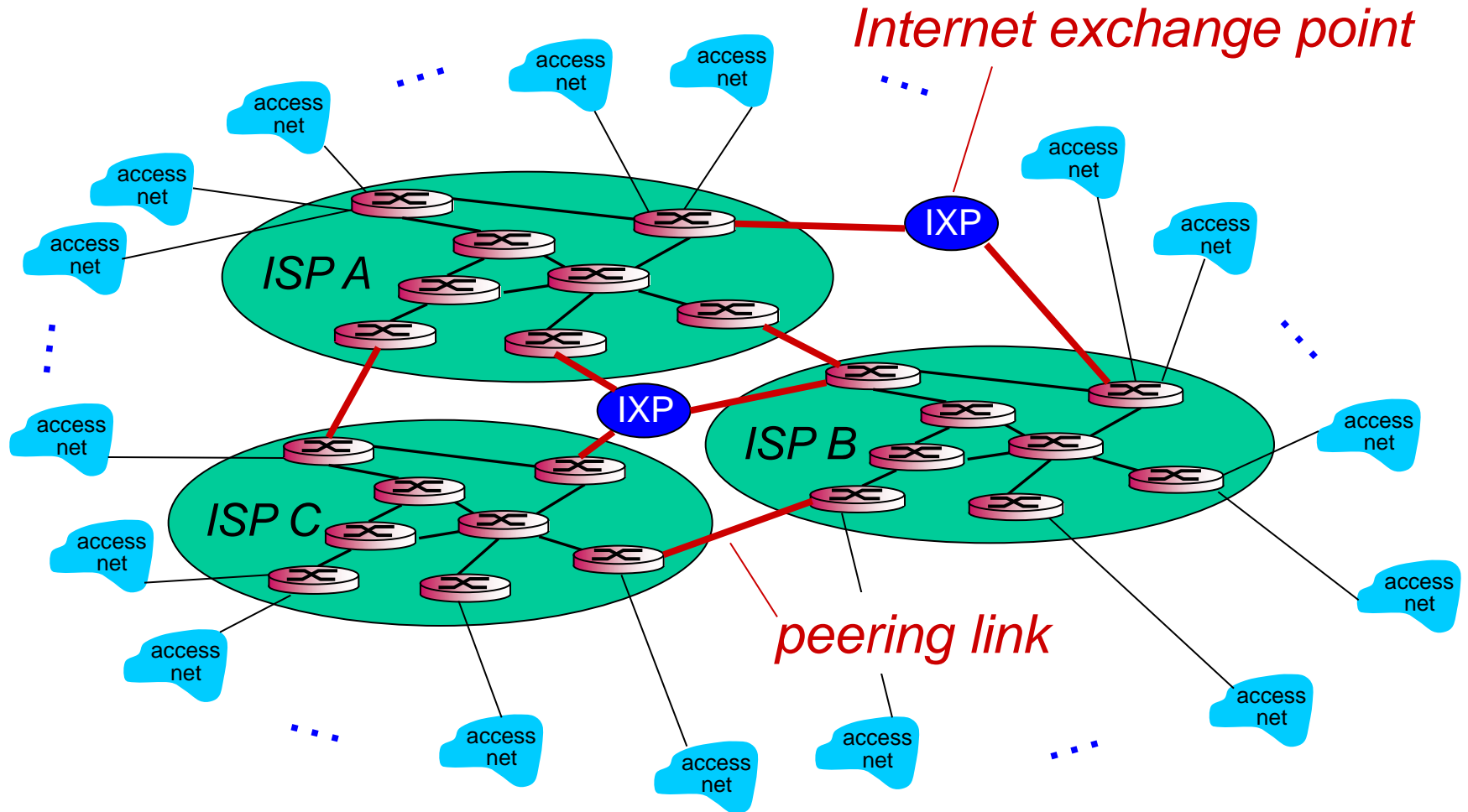
1.3.3 网络的网络

政治因素和经济上的竞争，导致不可能只有一个全球的ISP



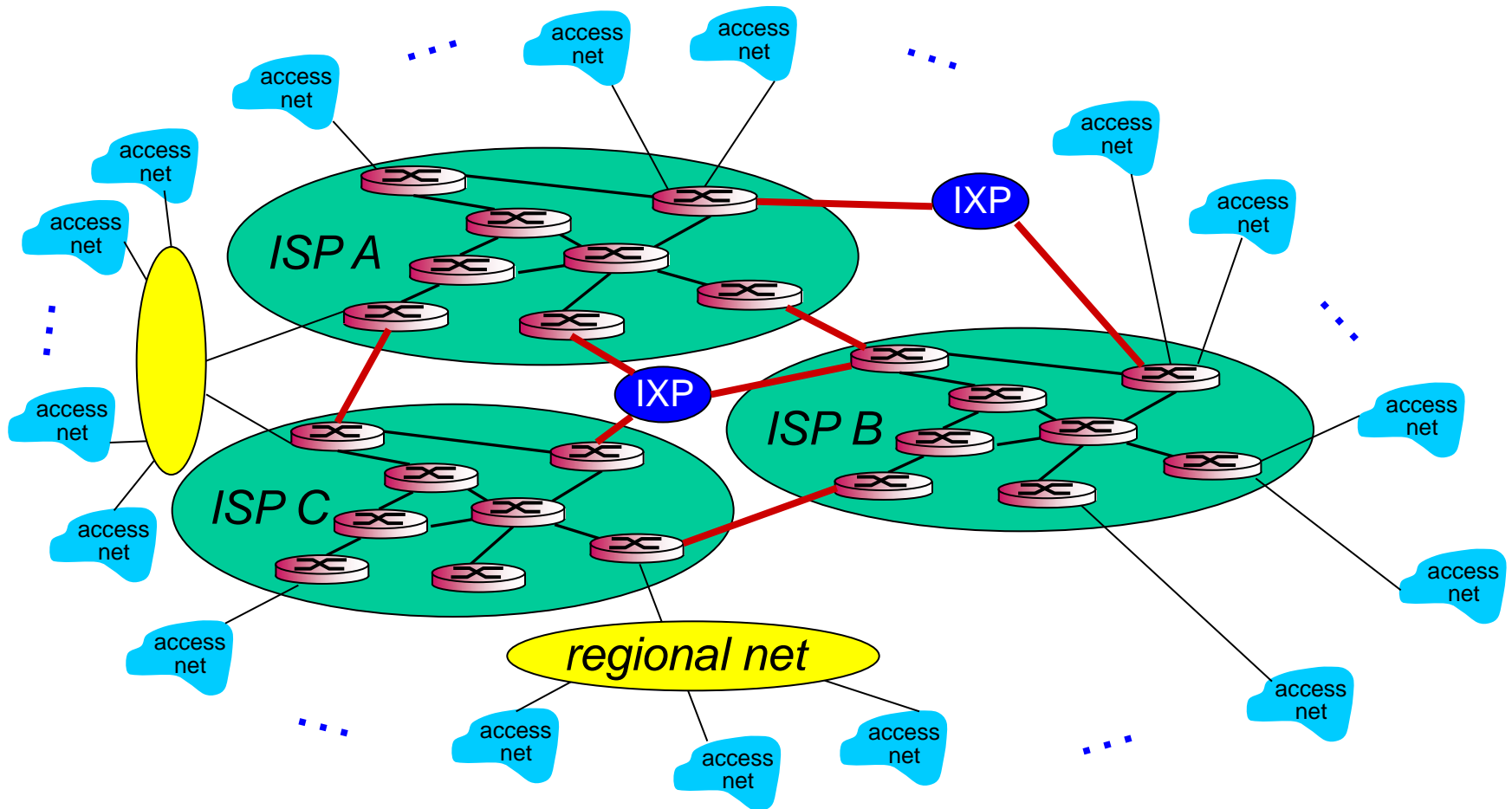
1.3.3 网络的网络

这些全球的ISPs需要相互连接



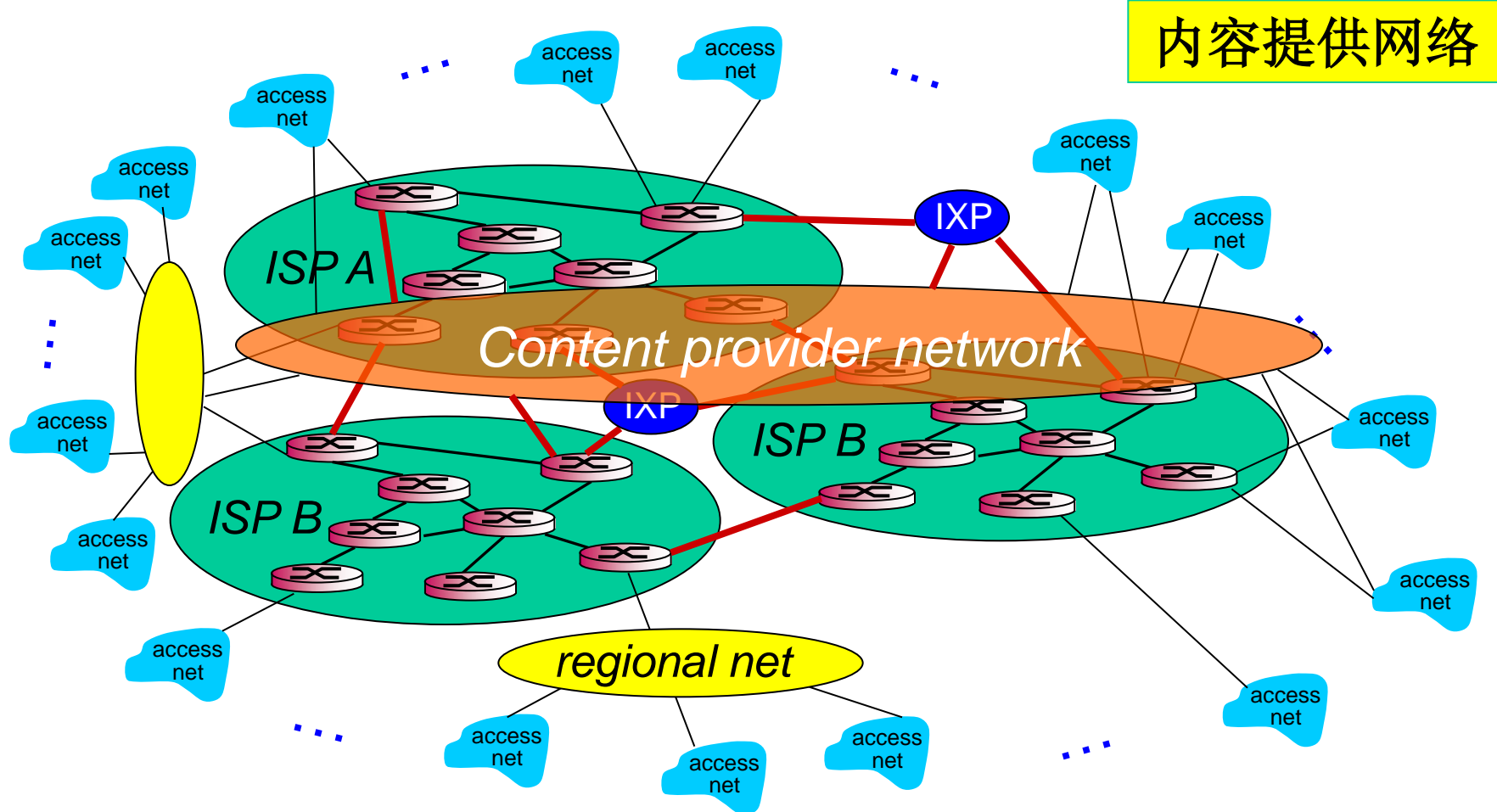
1.3.3 网络的网络

可能会出现区域网络，将接入网连接到全球ISPs

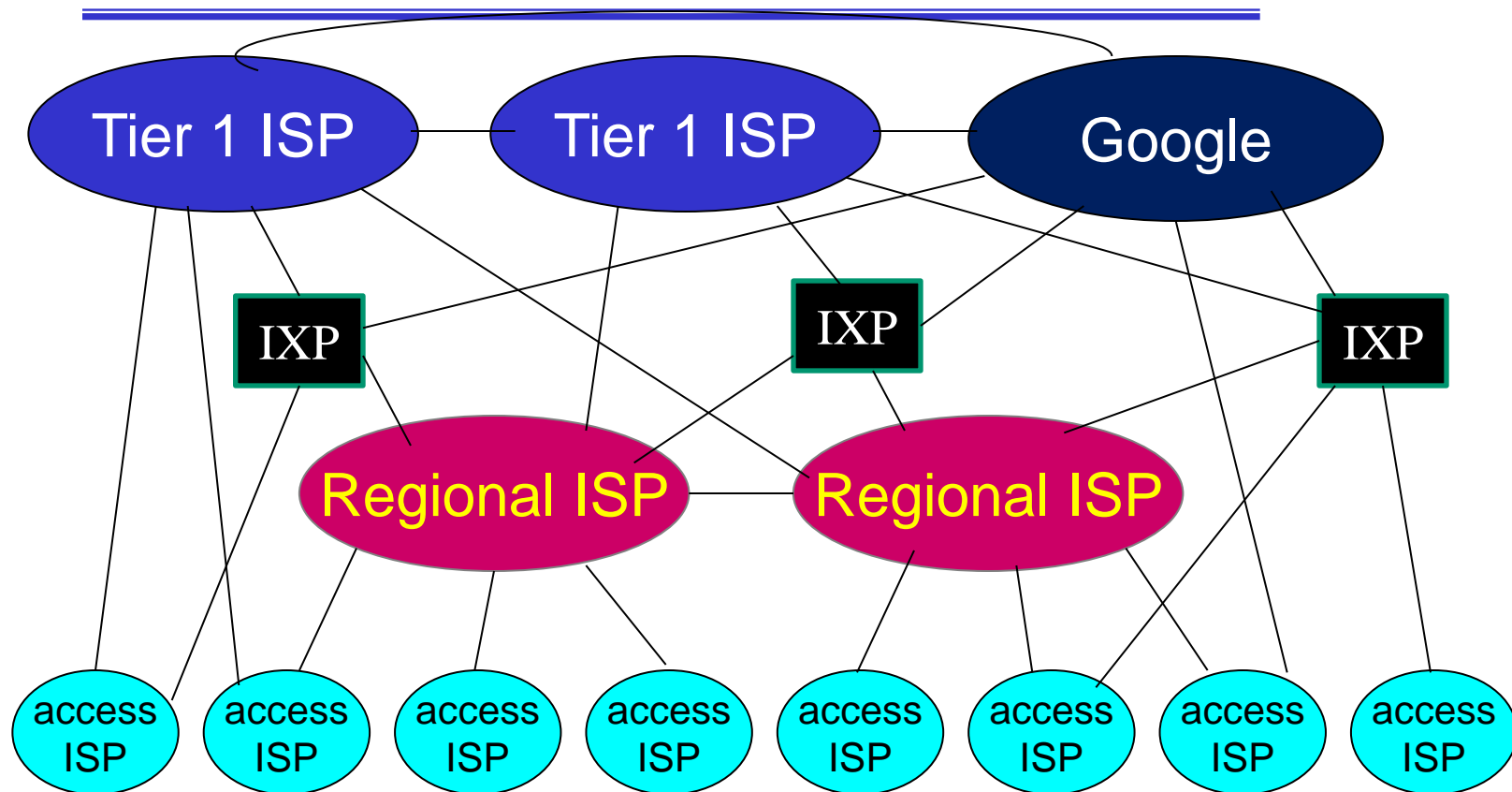


1.3.3 网络的网络

…内容提供商网络(e.g., Google, Microsoft, Akamai)可以运行自己的网络, 将服务、内容靠近最终用户



1.3.3 网络的网络



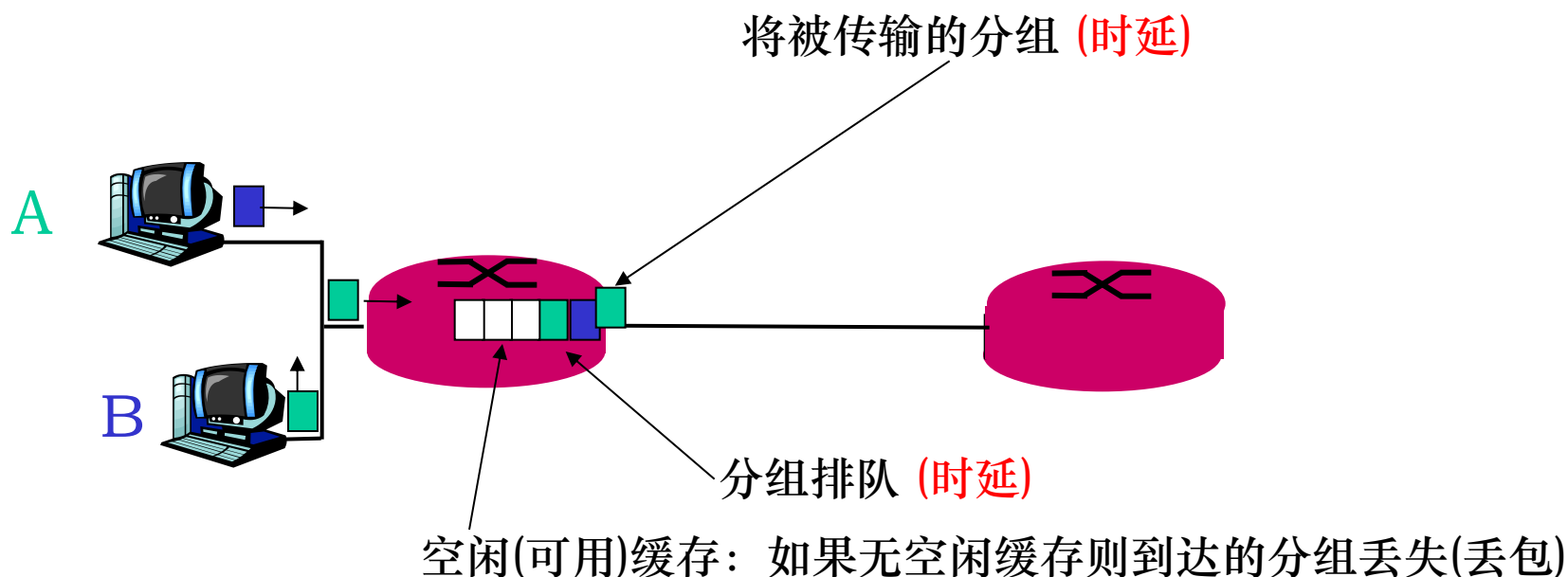
□ 中心：少数几个连通性很好的大型网络

- “**tier-1**” **商业 ISPs** (e.g., Level 3, Sprint, AT&T, NTT), 全国或国际范围的覆盖
- **内容提供网络** (e.g., Google): 私有网络能够将其数据中心与因特网互连, 通常可以绕过 tier-1 和 regional ISPs

1.4 分组交换网中的时延、丢包和吞吐量

时延和丢包是怎样出现的？

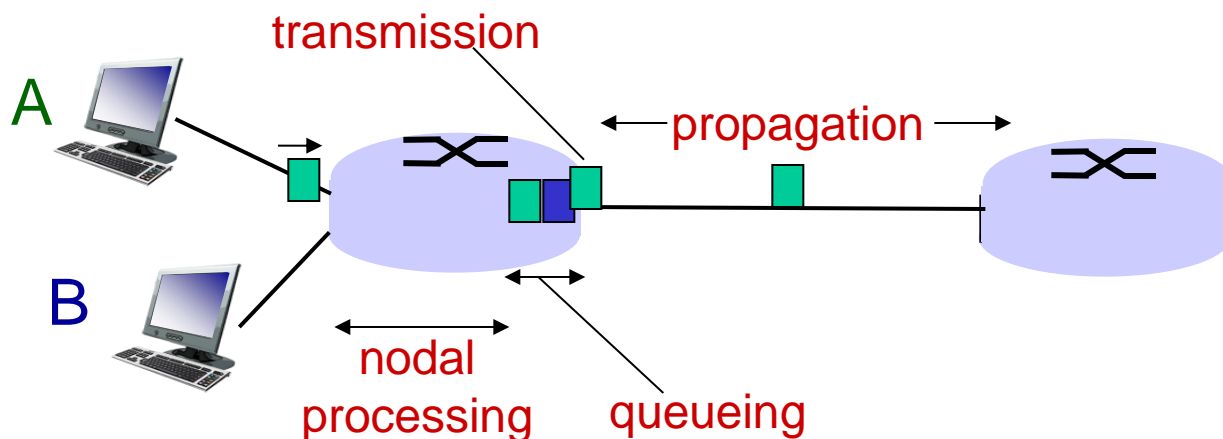
- 分组在每个路由器中先存储，再转发
 - 缓存中排队，等待交换 —— 时延
 - 分组到达链路的速率超过输出链路能力 —— 丢包



1.4.1 分组交换网中的时延概述

- 分组传输过程：从源主机出发，通过一系列路由器传输，最后到目的主机。
- 产生四种时延：节点处理时延、排队时延、传输时延和传播时延等。
- 节点总时延：各时延累加。

四种分组时延(delay)



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

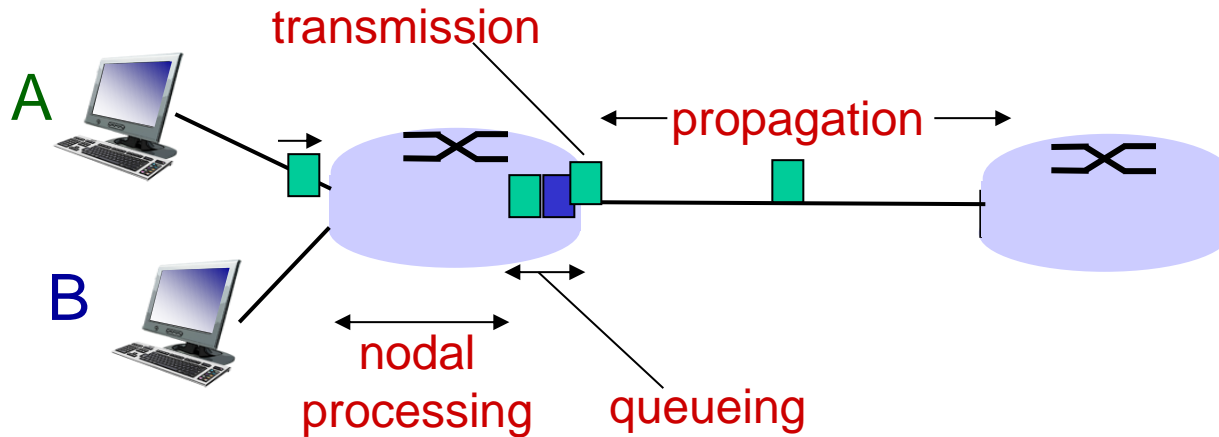
d_{proc} : 节点处理时延

- 检测比特错误
- 确定输出链路
- 通常时延小于毫秒级别

d_{queue} : 排队时延

- 在输出端口等待传输的时间
- 依赖于路由器的拥塞等级

四种分组时延(delay)



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

d_{trans} : 传输时延

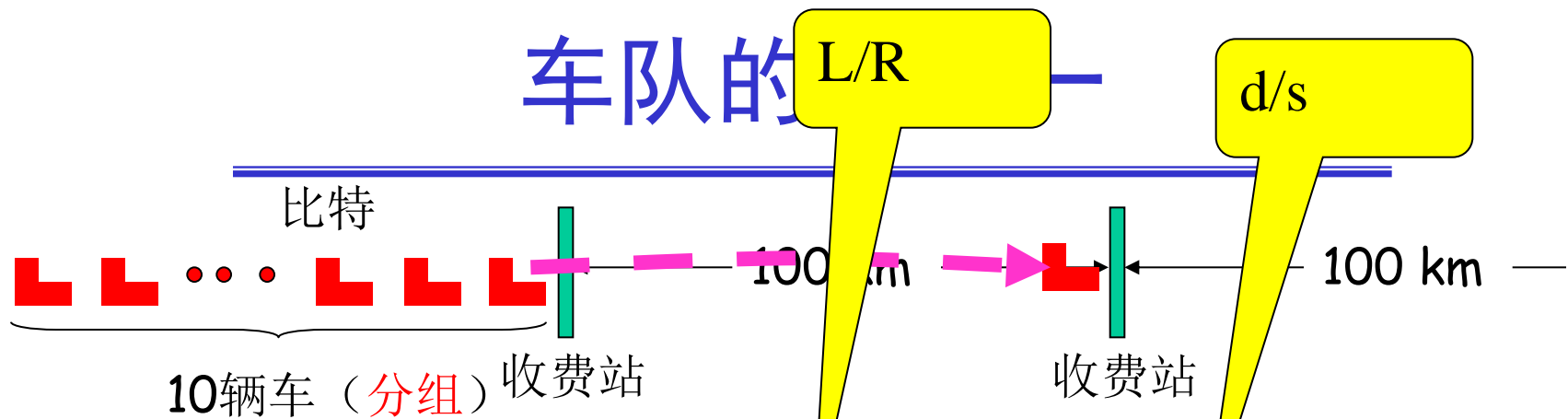
- L : packet length (bits)
- R : link bandwidth (bps)
- $d_{\text{trans}} = L/R$

d_{prop} : 传播时延

- d : length of physical link
- s : propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- $d_{\text{prop}} = d/s$

d_{trans} and d_{prop}
very different

* Check out the Java applet for an interactive animation on trans vs. prop delay

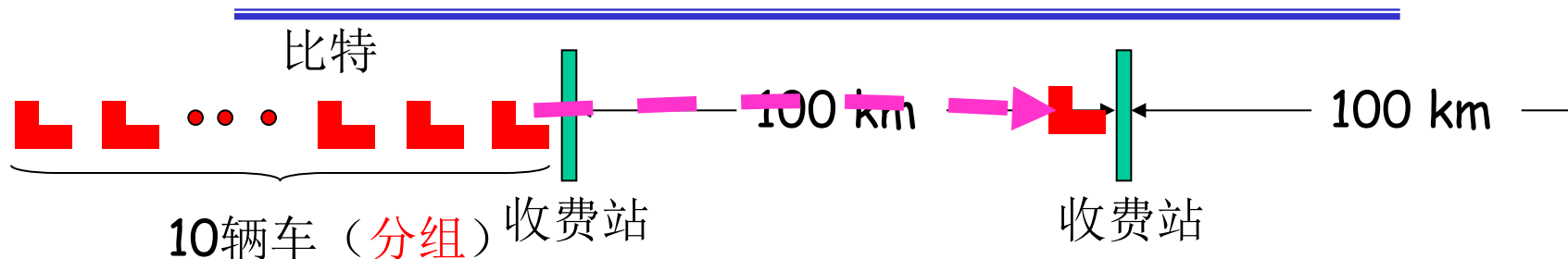


车队从第一个收费站出发，到达第二个收费站的时间？

“传输时延”和“传播时延”总和

- 设行驶速度（传播速率）100km/h，收费站12 sec服务一辆车（传输时间）
 - 链路传输速率 R ：60/12=5辆车/分钟
 - 传输时延：收费站将整个车队推向公路所需要的时间
 $(10\text{辆车})/(5\text{辆车/分钟})=2\text{分钟}$ 或 $12 \times 10 = 120\text{ sec}$
 - 传播时延：100km/(100km/h)=1小时
 - 总时间：62分钟

车队的类比二（传输时延大于传播时延）



□ 设行驶速度（**传播速率**）1000km/h，收费站一分钟服务一辆车(**传输时间**)

○ **传播时延**：100km/(1000km/h)=6分钟

○ **传输时延**：收费站将整个车队推向公路需要10分钟

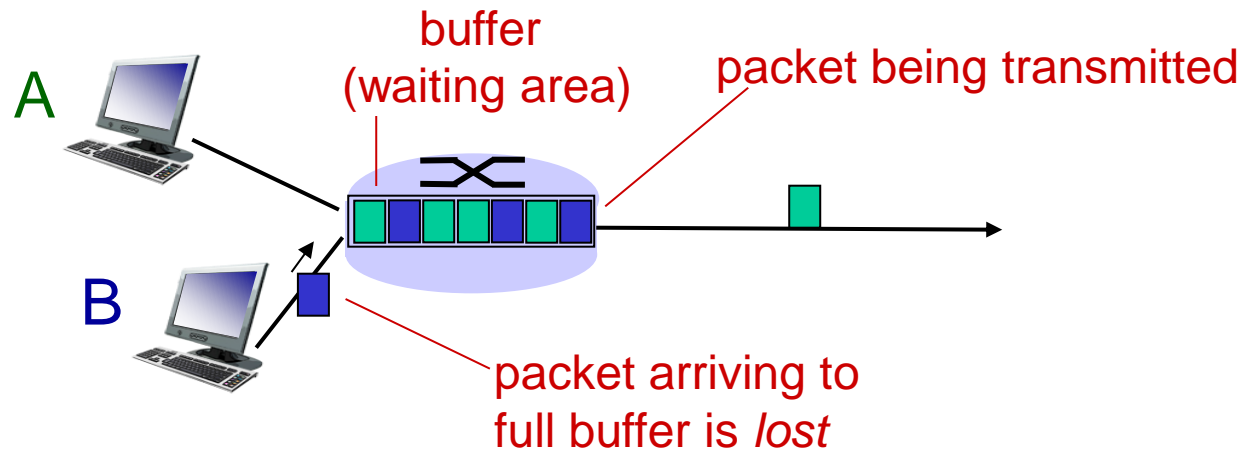
即收费站服务车队的时间大于汽车行驶的时间。

7分钟后，第一辆车到达第二个收费站，后面几辆车仍在第一个收费站。

传输时延大于传播时延时，一个分组中的前几个比特已到达下一路由器，而余下的仍然在前面的路由器中等待传输。

1.4.2 分组丢包

- ❑ 链路的缓冲区(buffer)的容量是有限的
- ❑ 分组到达一个排满的队列时会产生丢包(lost)
- ❑ 丢失的分组可以由前一个节点重传,也可以由源终端系统重传,或根据不需要重传(这里可以请同学们讨论应该用什么方式?)



* Check out the Java applet for an interactive animation on queuing and loss

1.4.3 端到端时延

□ 从源到目的地的时延。

- 设源主机和目的主机之间有 $N-1$ 个路由器(N 条链路)
- 网络无拥塞, 忽略排队时延
- 路由器和源主机的处理时延是 d_{proc}
- 路由器和源主机的输出速率是 $R\text{bit/s}$
- 每条链路的传播时延是 d_{prop}
- 每个节点时延: $d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}}$
- 端到端时延:

$$d_{\text{end-end}} = N(d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}})$$

$$d_{\text{trans}} = L/R$$

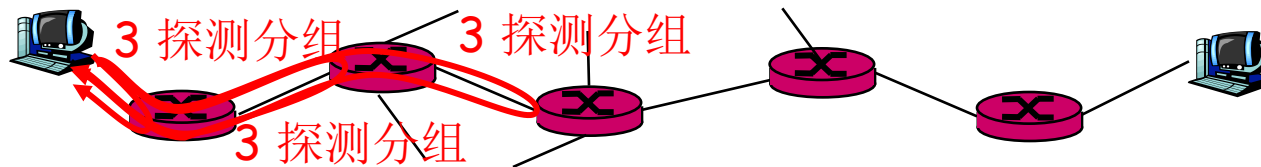
测量真实的因特网时延和转发路径

Traceroute诊断程序，在任何因特网主机上运行。`ping/ipconfig`

- Traceroute程序可追踪源和目的之间经过的路由。
- 当用户指定一个目的主机时，源主机中的该程序朝目的地发送多个特殊的分组，中间通过一系列路由器。

对所有路由器 i :


- ✓ 发送3个分组，该分组在朝着目的地的路径上到达路由器 i
- ✓ 路由器 i 将向发送方返回分组
- ✓ 发送方度量传输和响应间的时间间隔



“Real” Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from
gaia.cs.umass.edu to cs-gw.cs.umass.edu



```
1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms
```

trans-oceanic link

* means no response (probe lost, router not replying)

尝试：tracert命令

```
C:\Users\Administrator>tracert www.uestc.edu.cn
```

通过最多 30 个跃点跟踪
到 www.uestc.edu.cn [202.112.14.178] 的路由：

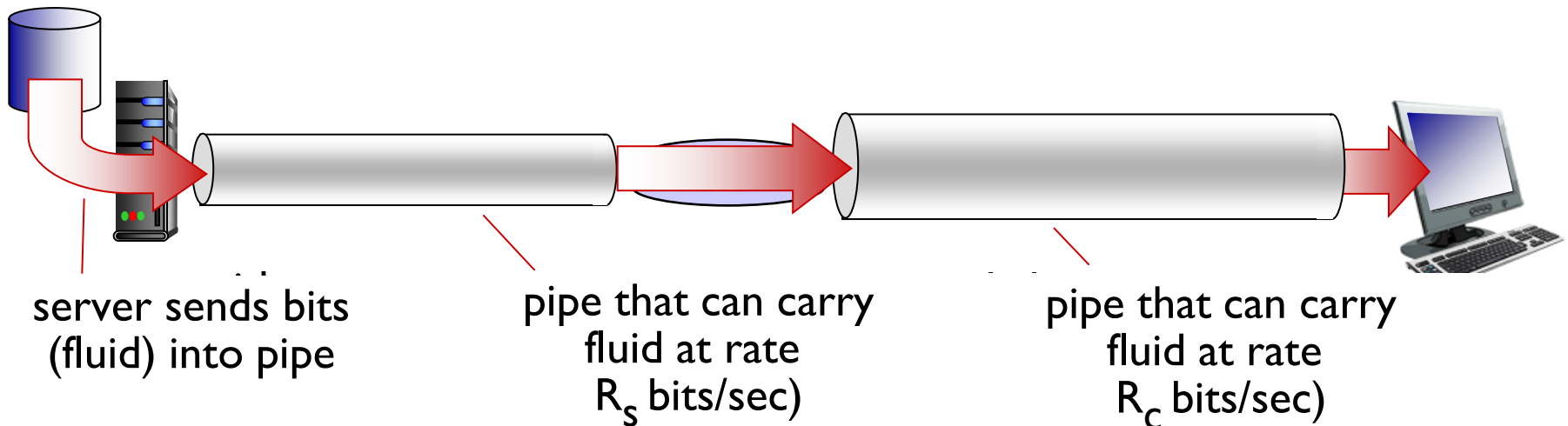
1	3 ms	4 ms	1 ms	192.168.1.1
2	4 ms	2 ms	1 ms	192.168.0.1
3	8 ms	3 ms	2 ms	210.41.108.1
4	3 ms	2 ms	2 ms	202.115.0.5
5	×	×	×	请求超时。
6	8 ms	7 ms	10 ms	202.112.14.178

跟踪完成。

第1列是经过路由器的编号，第2-4列是往返时延，最后一列是经过的路由器的到达端口的IP地址

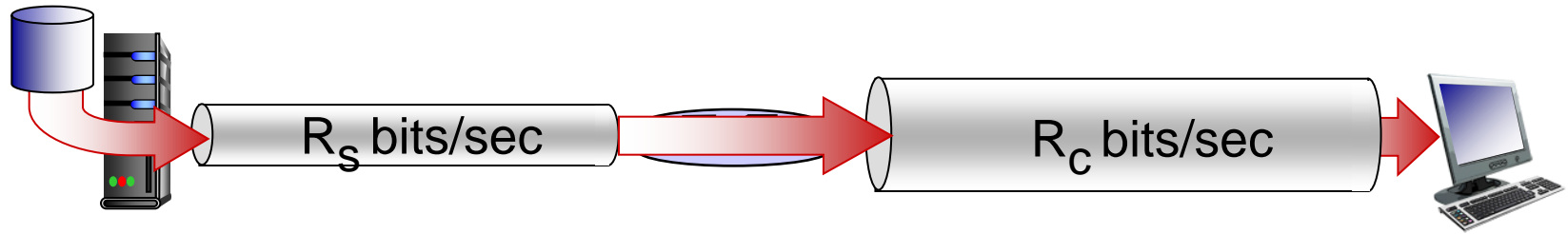
1.4.4 吞吐量 (Throughput)

- *throughput*: rate (bits/time unit) at which bits transferred between sender/receiver
 - *instantaneous*: rate at given point in time
 - *average*: rate over longer period of time

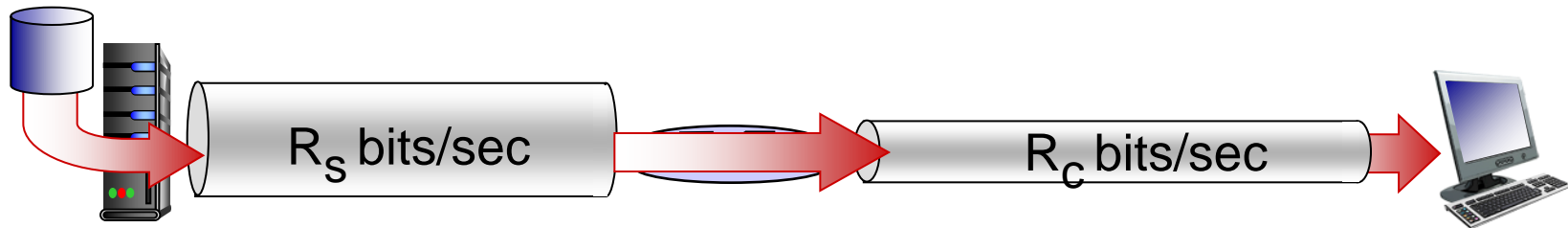


1.4.4 吞吐量 (Throughput)

❖ $R_s < R_c$ What is average end-end throughput?



❖ $R_s > R_c$ What is average end-end throughput?

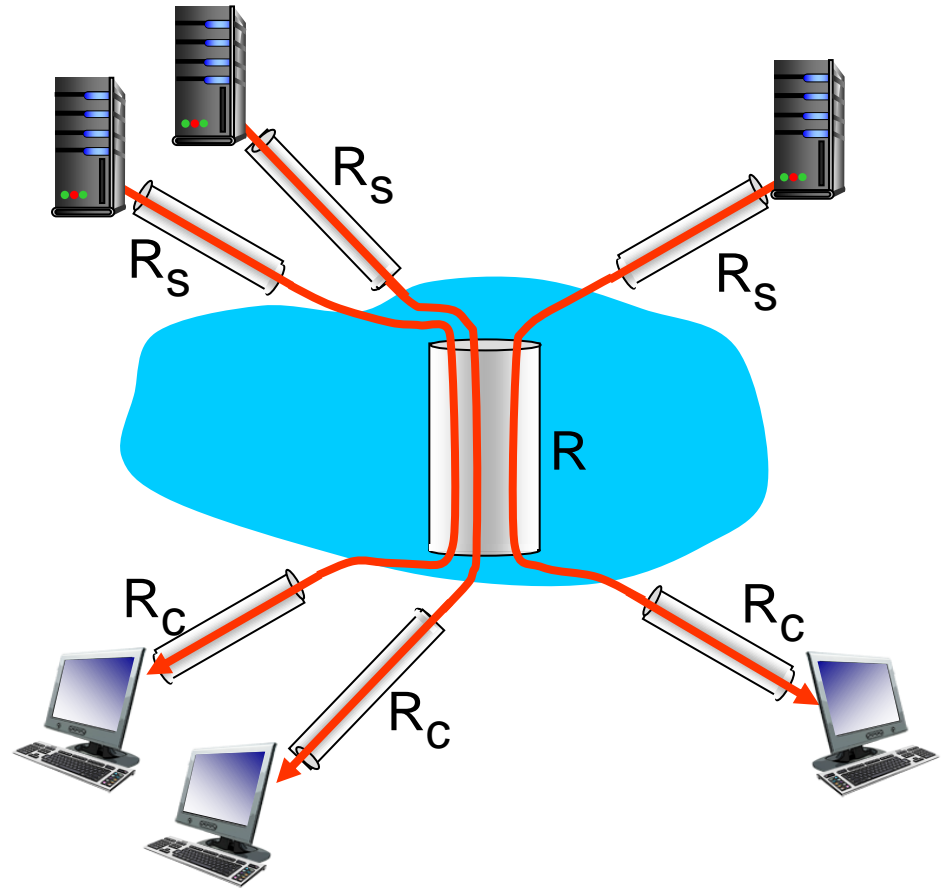


bottleneck link

link on end-end path that constrains end-end throughput

1.4.4 吞吐量：因特网的场景

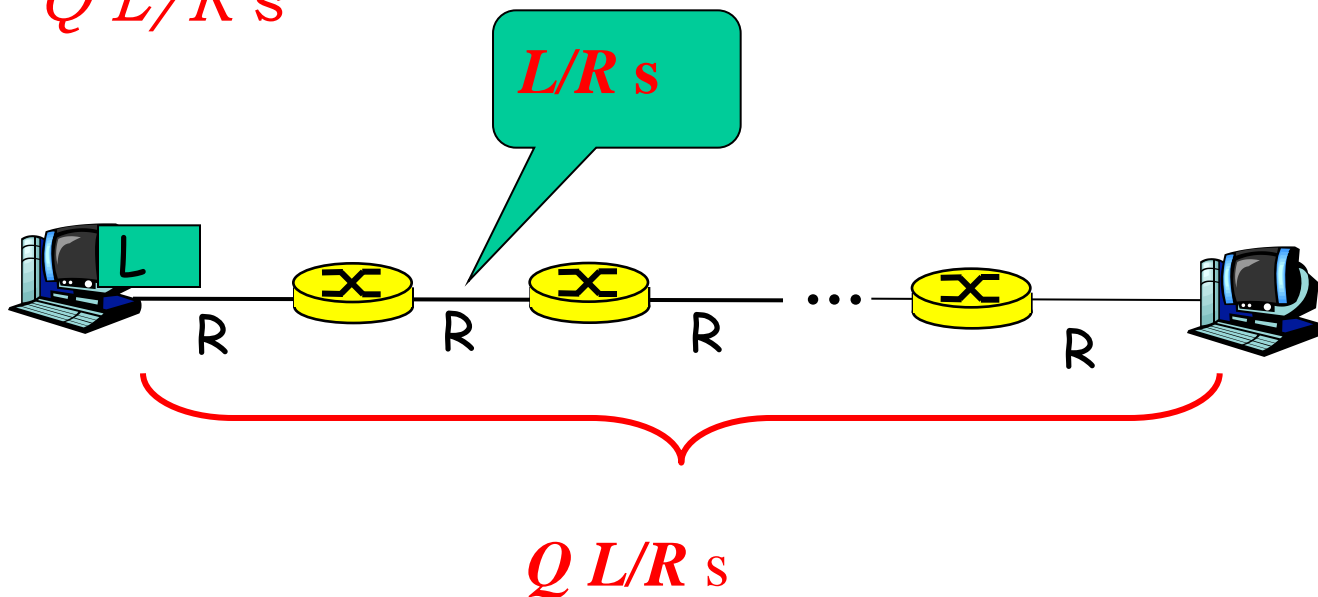
- ❖ per-connection end-end throughput:
 $\min(R_c, R_s, R/10)$
- ❖ in practice: R_c or R_s is often bottleneck



10 connections (fairly) share
backbone bottleneck link R bits/sec

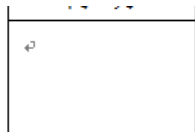
例：发送一个文件的时间

- ❖ 设两台主机之间有 Q 段链路，每条速率是 R bit/s，分组长 L bit，忽略排队时延和端到端传播时延。
- ❖ 每条链路传输时间： L/R s
- ❖ 经过中间路由器存储转发 $Q-1$ 次到目的地。
- ❖ 总时延： $Q L/R$ s



课堂练习

P48-P6



四、假设将从源主机向目的主机发送 40Mbit 的 Word 文档。在源和目的主机之间的路径中所有链路的传输速率都是 10Mbps。假设传播速率是

$2 \times 10^8 \text{ m/s}$ ，请回答以下问题：(16 分)

- (1) 假设源和目的端之间单独使用一条距离是 10^7 m 的物理链路连接，使用报文交换，该报文包括了完整的该 Word 文档，且忽略首部等封装用的字节的长度，忽略处理时延，分别计算传输时延、传播时延和端到端时延。(3 分)
- (2) 假设源和目的端之间单独使用一条距离是 10^7 m 的物理链路连接，使用 TDM 方式的电路交换，每帧划分为 10 个时隙，该 Word 文档使用其中一条电路进行发送，且发送时连接已经建立，分别计算传输时延、传播时延和端到端时延。(3 分)
- (3) 假设源和目的端之间的路径是由一台路由器连接的 2 条链路所组成，每条链路长度为 $5 \times 10^6 \text{ m}$ ，使用报文交换，该报文包括了完整的该 Word 文档，且忽略首部等封装用的字节的长度，忽略处理时延和排队时延，计算从发送方开始发送到接收方接受到完整的 Word 文档所花费的时间。(5 分)
- (4) 假设源和目的端之间的路径是由一台路由器连接的 2 条链路所组成，每条链路长度为 $5 \times 10^6 \text{ m}$ ，使用分组交换，该 Word 文档分为 4 个分组，每个分组长度为 10Mb，且忽略首部等封装用的字节的长度，忽略处理时延和排队时延，4 个分组连续发送，计算从发送方开始发送到接收方接受到完整的 Word 文档所花费的时间。(5 分)

1.5 协议层次及其服务模型

1.5.1 分层的体系结构

□ 网络协议为什么要进行分层？

- 使复杂系统简化：将一个大而复杂系统划分为若干个明确、特定的部分，分别讨论研究。
- 易于维护、系统的更新：某层功能变化，不会影响系统其余部分。

□ 分层的特点

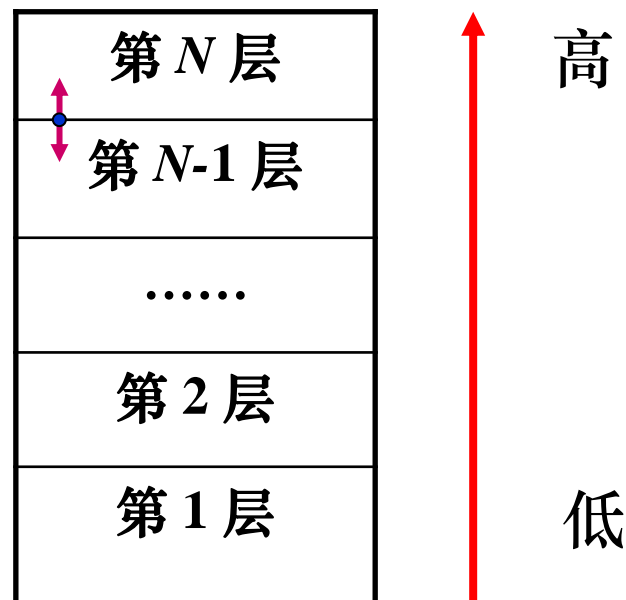
- 每个层次实现一个相对单一的服务
- 每层自己执行一些动作来完成服务
- **每层可以使用相邻下一层提供的服务**

1.5.1 分层的体系结构

上层调用下层提供的服务

下层为上层提供服务

- 第N层只能调用第N-1层提供的服务；而不能调用N-2层提供的服务；
- 第N-1层不能调用第N层提供的服务

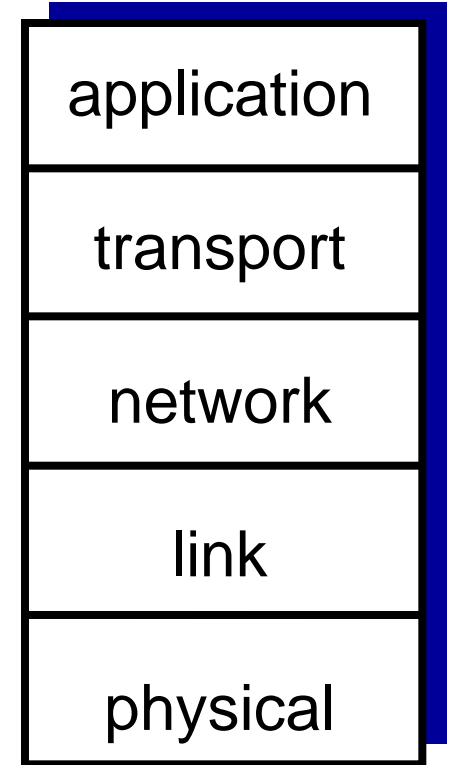


请同学们思考分层协议的缺点？

- 有些功能可能在不同层重复出现：如基于链路和基于端到端传输的差错恢复；
- 某层的功能可能需要仅存在其他某层的信息。

TCP/IP协议栈

- ❖ *application*: 支持网络应用
 - FTP, SMTP, HTTP
- ❖ *transport*: 完成(process-process)的数据传输
 - TCP, UDP
- ❖ *network*: routing of datagrams from source to destination
 - IP, routing protocols
- ❖ *link*: data transfer between neighboring network elements
 - Ethernet, 802.11 (WiFi), PPP
- ❖ *physical*: bits “on the wire”

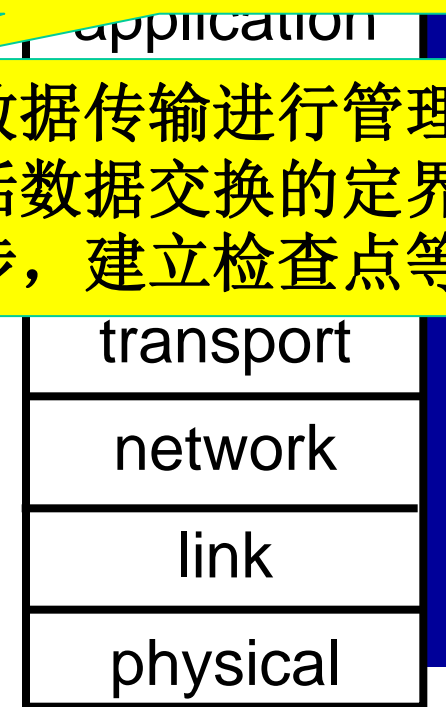


OSI 参考模型 (7层)

- ❖ *presentation*: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- ❖ *session*: synchronization, checkpointing, recovery of data exchange
- ❖ 因特网的协议栈不包含这两层
 - these services, *if needed*, must be implemented in application
 - needed?

加解密、数据压缩、格式转换

对数据传输进行管理，包括数据交换的定界、同步，建立检查点等。



在不同网络终端设备的层次

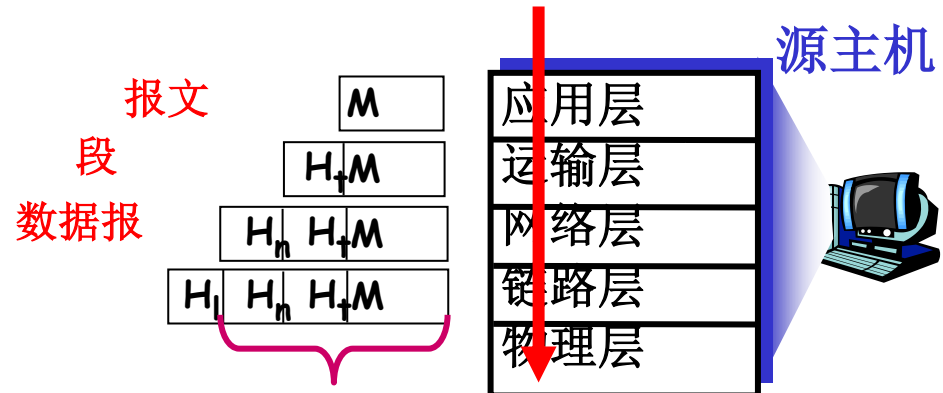
- 与端系统类似，路由器和链路层交换机以分层方式组织网络硬件和软件，通常只实现低几层。
 - **路由器**：实现第一层到第三层，能够实现IP协议；
 - **链路层交换机**：实现第一层和第二层，能够识别第二层地址，如以太网地址。
 - **主机**：实现所有5个层次。

1.5.2 封装(Encapsulation)

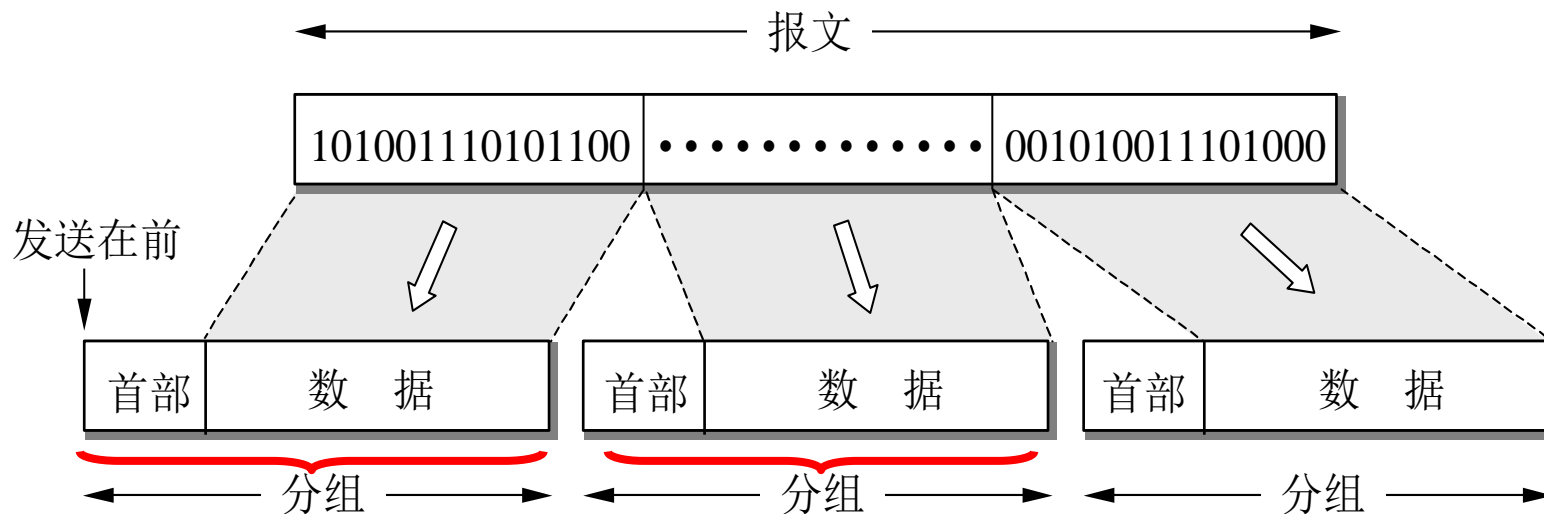
❖ 源主机：由高层向低层逐层传递数据（封装）

- 应用层报文(message)M传递到运输层，附加上运输层首部信息 H_t （运输层**报文段(segment)**）；
- 报文段传递到网络层，附加上网络层首部信息 H_n （网络层**数据报(datagram)**）；
- 数据报传递到链路层，附加上链路层首部信息 H_l （链路层**帧(frame)**）
- 传到物理层(**比特流(bits)**)，送入网络传输。

- 每层传递的数据分为首部字段和有效载荷字段两部分。
- 有效载荷是相邻上层传下来的数据。



1.5.2 封装(Encapsulation)



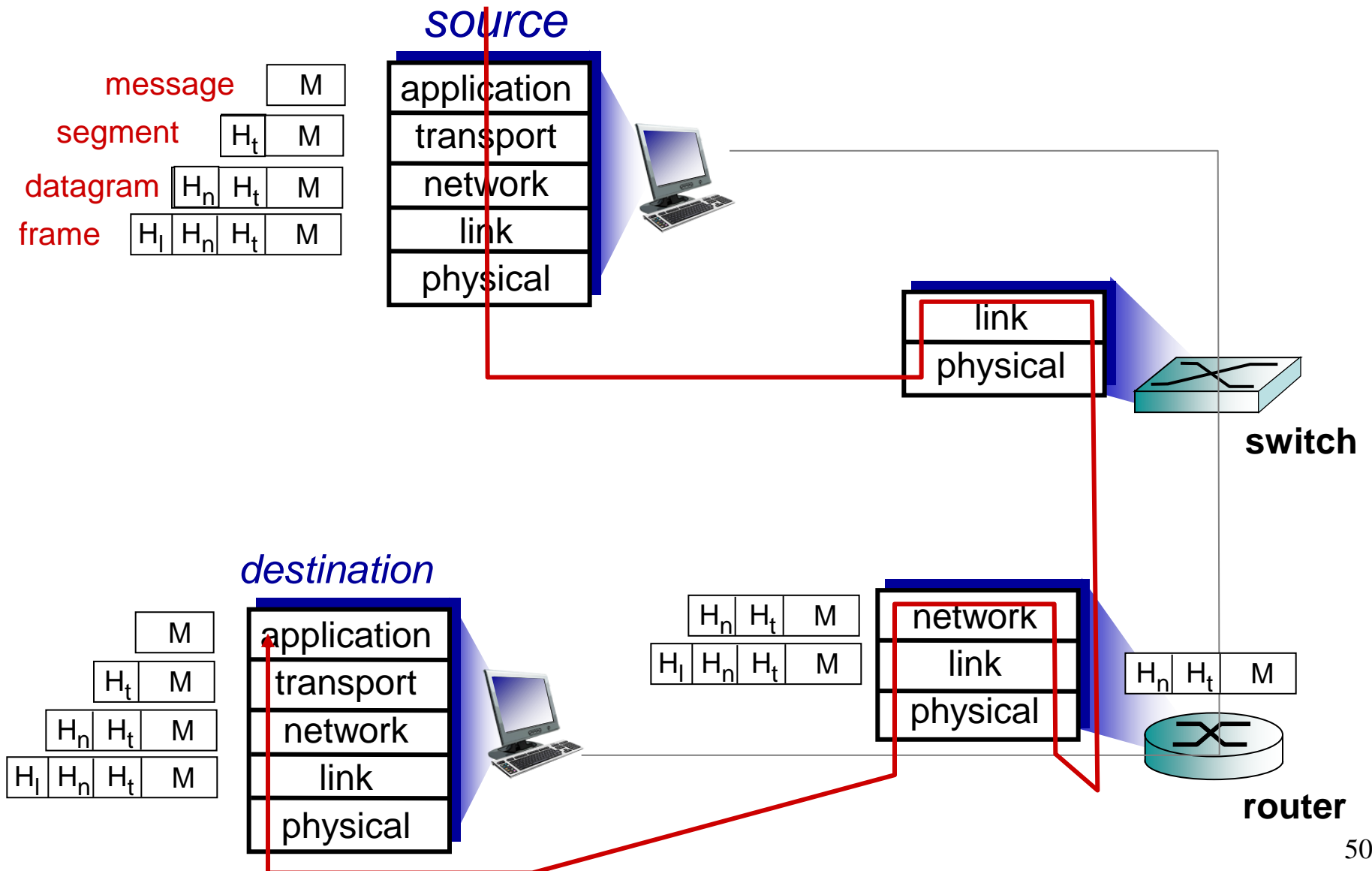
1.5.2 封装(Encapsulation)

- ❑ 目的主机：由低层向高层逐层传递（解封装）
 - 物理层接收，并将其沿协议栈逐层向上传递，每层去除对应的首部，恢复原报文。

如果报文很长，传输时，可先分成多个报文段，每个报文段在网络层再分成多个数据报。



1.5.2 封装(Encapsulation)



1.6 面对攻击的网络

□ 网络安全的领域

- 坏人攻击计算机网络的手段有哪些
- 我们抵抗网络攻击的手段有哪些
- 如何设计好的网络架构使不被攻击

□ 因特网最初的设计没有考虑安全问题

- 初始想法：一组相互可信的用户连接到一个透明的网络中
- 因特网协议设计者正在玩“追赶游戏”
- 应该在所有层次上考虑安全问题！

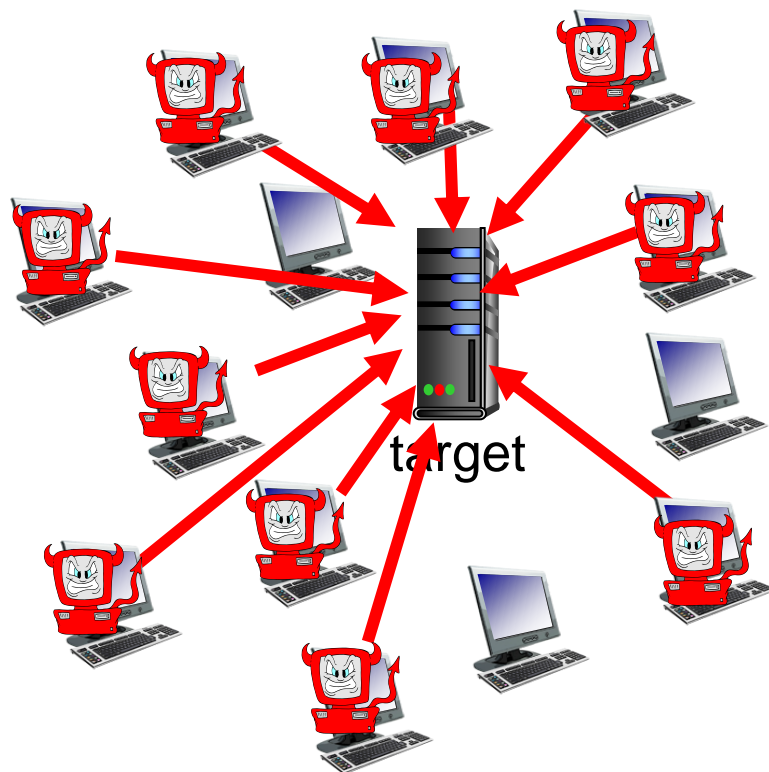
坏家伙能够经因特网将有害程序放入你的计算机中

- ❑ **恶意代码**(Malware)可以从以下方式进入主机:
 - **Virus(病毒)**: 通过接收/执行对象自我复制的方式感染(例如, e-mail附件的方式)
 - **Worm(蠕虫)**: 通过接收能够自己执行的对象实现自我复制感染
- ❑ **间谍软件**(spyware malware)可以记录击键、访问的网站, 将信息上载到收集网站
- ❑ 被感染的主机可以在僵尸网络(botnet)中注册, 参与垃圾邮件(spam)、DDoS攻击中

坏家伙能够攻击服务器和基础设施

- ❑ **Denial of Service (DoS):** 攻击者使资源(server/bandwidth)对于合法的流量不可用，通过使用虚假的流量来耗费资源

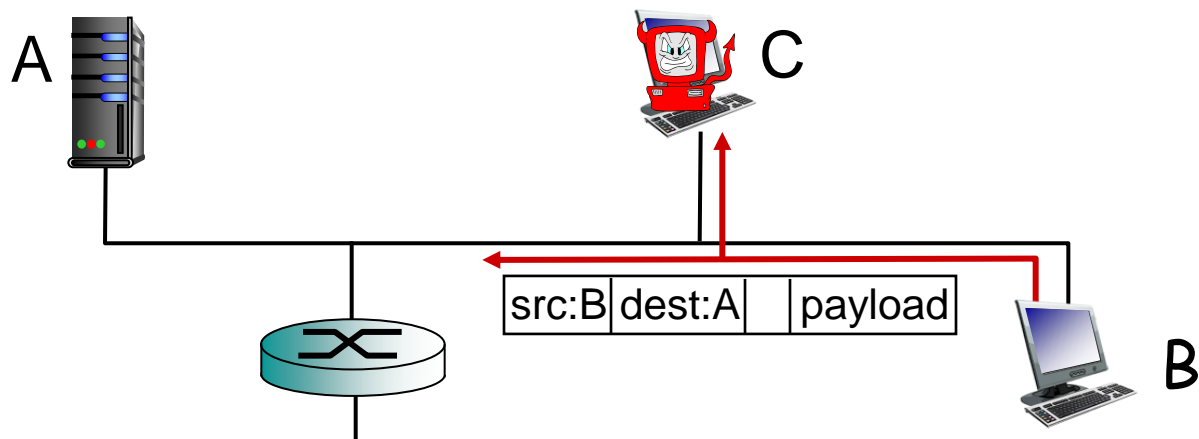
1. 选择目标
2. 感染若干主机
3. 从受感染的主机发送分组到目标设备



坏家伙能够嗅探(sniff)分组

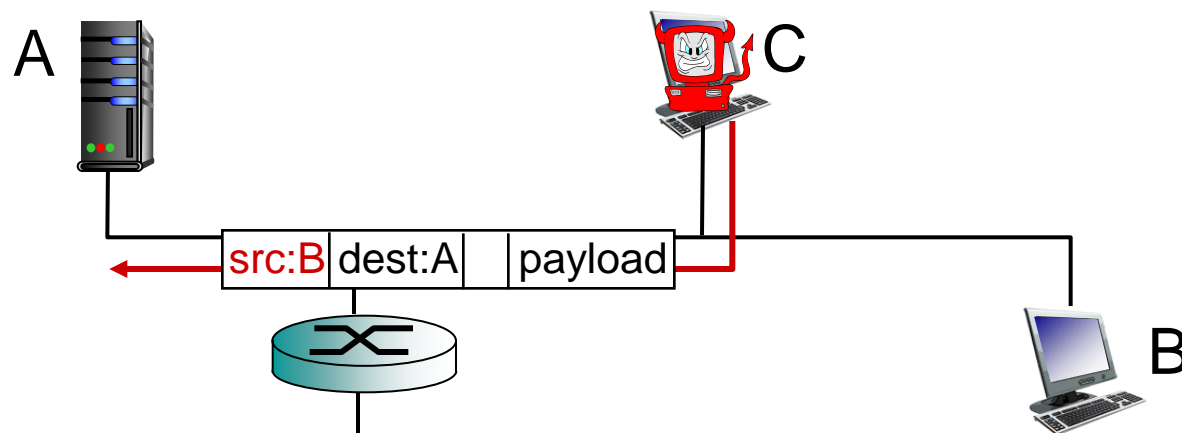
□ 分组嗅探

- 广播信道(shared Ethernet, wireless)
- 将网络接口设置为混杂模式，可以读取/记录所有它能收到的数据包（例如，密码！）
- Wireshark软件是一个免费的分组嗅探器



坏家伙能够伪装成你信任的人

- IP哄骗(IP spoofing): 采用伪装的源IP地址发送分组

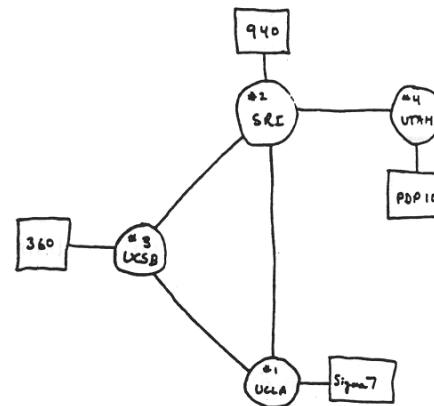


... lots more on security (throughout, Chapter 8)

1.7 因特网的历史

1961-1972: Early packet-switching principles

- ❖ 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- ❖ 1964: Baran - packet-switching in military nets
- ❖ 1967: ARPAnet conceived by Advanced Research Projects Agency
- ❖ 1969: first ARPAnet node operational
- ❖ 1972:
 - ARPAnet public demo
 - NCP (Network Control Protocol) first host-host protocol
 - first e-mail program
 - ARPAnet has 15 nodes



THE ARPA NETWORK

1.7 因特网的历史

1972-1980: Internetworking, new and proprietary nets

- ❖ 1970: ALOHAnet satellite network in Hawaii
- ❖ 1974: Cerf and Kahn - architecture for interconnecting networks
- ❖ 1976: Ethernet at Xerox PARC
- ❖ late70' s: proprietary architectures: DECnet, SNA, XNA
- ❖ late 70' s: switching fixed length packets (ATM precursor)
- ❖ 1979: ARPAnet has 200 nodes

Cerf and Kahn' s internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today' s Internet
architecture

1.7 因特网的历史

1980-1990: new protocols, a proliferation of networks

- ❖ 1983: deployment of TCP/IP
- ❖ 1982: smtp e-mail protocol defined
- ❖ 1983: DNS defined for name-to-IP-address translation
- ❖ 1985: ftp protocol defined
- ❖ 1988: TCP congestion control
- ❖ new national networks: Csnet, BITnet, NSFnet, Minitel
- ❖ 100,000 hosts connected to confederation of networks

1.7 因特网的历史

1990, 2000's: commercialization, the Web, new apps

- ❖ early 1990's: ARPAnet decommissioned
- ❖ 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- ❖ early 1990s: Web
 - hypertext [Bush 1945, Nelson 1960's]
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, later Netscape
 - late 1990's: commercialization of the Web
- late 1990's – 2000's:
 - ❖ more killer apps: instant messaging, P2P file sharing
 - ❖ network security to forefront
 - ❖ est. 50 million host, 100 million+ users
 - ❖ backbone links running at Gbps

1.7 因特网的历史

2005-present

- ❖ ~750 million hosts
 - Smartphones and tablets
- ❖ Aggressive deployment of broadband access
- ❖ Increasing ubiquity of high-speed wireless access
- ❖ Emergence of online social networks:
 - Facebook: soon one billion users
- ❖ Service providers (Google, Microsoft) create their own networks
 - Bypass Internet, providing “instantaneous” access to search, email, etc.
- ❖ E-commerce, universities, enterprises running their services in “cloud” (eg, Amazon EC2)

1.8 本章小结

- ❑ 构成网络的各种软硬件：
 - 网络边缘：端系统和应用程序，及提供的运输服务；
 - 协议：
 - 网络核心：分组交换和电路交换；
 - 链路层技术和物理媒体：
 - 因特网的等级结构：由较高层和较低层ISP组成。
- ❑ 分组交换网的时延和分组丢失：
- 介绍传输、传播和排队时延的简单定量模型。
- ❑ 协议分层和服务模型、体系结构：
 - 分层特点、数据传递方法、因特网分层。

重点掌握术语、概念、基本方法和原理

课堂练习

- 在协议分层服务模型中，服务的定义为（ ）。
A、各层向下提供的一组原语操作 B、各层间对等实体间通信的功能实现
C、各层通过其SAP向上层提供的一组功能 D、和协议的含义是一样的
- 当一台计算机从FTP服务器下载文件时，在该FTP服务器上对数据进行封装的五个转换步骤是()。
A、比特、帧、数据报、报文段、报文 B、报文、报文段、数据报、帧、比特
C、数据报、报文段、报文、比特、帧 D、报文段、数据报、帧、比特、报文
- 因特网协议栈分为几个层次？分别是哪几个层次(按自顶向下的顺序描述)？简单描述每个层次的主要功能。

安装Wireshark工具包

开发商：The Wireshark community

软件官网：<http://www.wireshark.org/>

主要功能：Wireshark可用来检测网络问题，检查信息安全相关问题，可用于学习和分析网络协议，也可以用于测试新的通讯协议。仔细分析Wireshark抓取的数据包能够帮助使用者对于网络行为有更清楚的了解。Wireshark不会修改网络数据包的内容，它只会反映出目前通信中的数据包信息。Wireshark本身也不会向网络发送数据包。

- 支持UNIX和Windows平台
- 在接口实时捕捉包
- 能详细显示包的详细协议信息
- 可以打开/保存捕捉的包
- 可以导入导出其他捕捉程序支持的包数据格式
- 可以通过多种方式过滤包
- 多种方式查找包
- 通过过滤以多种色彩显示包
- 创建多种统计分析

Wireshark工具包

