

# 计算机网络:自顶向下方法

肖周芳

计算机学院 1教606

计算机网络:自顶向下方法（第7版）

J.F.Kurose, K.W.Ross著,陈鸣译,机械工业出版社,2018.

Computer Networking: A Top-Down Approach(Seventh Edition)

J.F.Kurose, K.W.Ross,2017.

本PPT改编自英文版教材附带的PPT。

# Chapter 1: roadmap

## 1.1 什么是 Internet?

## 1.2 网络边缘

- 端系统, 接入网, 链路

## 1.3 网络核心

- 电路交换, 数据报交换, 网络结构

## 1.4 包交换网络中的延迟, 丢包和吞吐率

## 1.5 协议层次, 服务模型

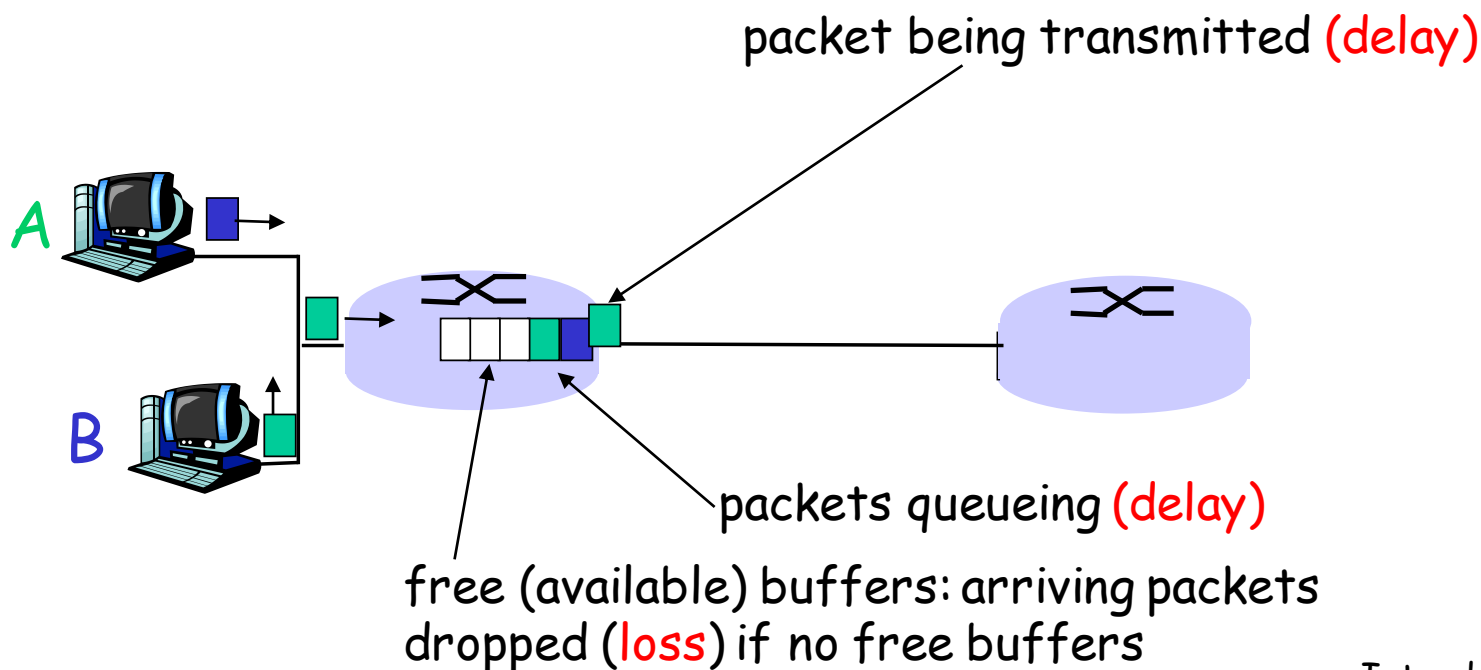
## 1.6 网络攻击: 安全性

## 1.7 历史

# 延迟和丢包的发生?

数据报在路由缓冲区中排队

- ❑ 数据报到达速率超过链路容量
- ❑ 数据报排队, 等待发送



# 数据报延迟的四个来源

## ❑ 1. 节点处理

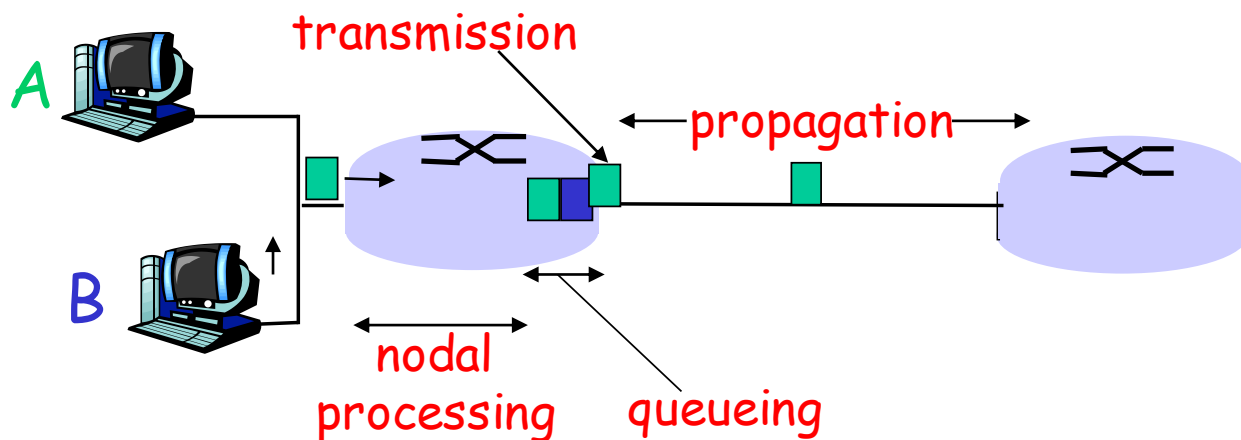
### ❑ (nodal processing)

- ❖ 检差比特错误
- ❖ 决定输出链路

## ❑ 2. 排队

### ❑ (queueing)

- ❖ 在输出链路等待传输的时间
- ❖ 依赖于路由器拥塞程度



# 数据报交换网路中的延迟

## 3. 传输延迟

### (Transmission delay)

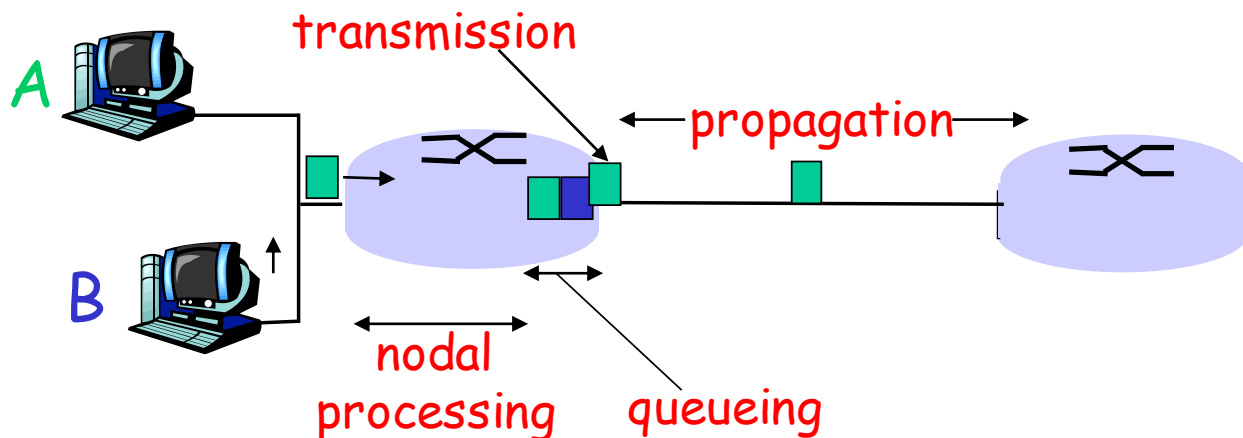
- $R$  = 链路带宽 (bps)
- $L$  = 数据报长度 (bits)
- 发送到链路的时间 =  $L/R$

## 4. 传播延迟

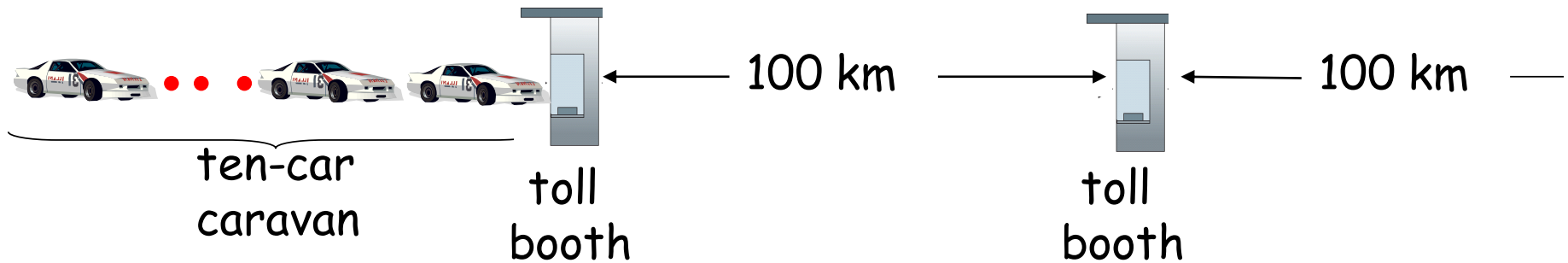
### (Propagation delay)

- $d$  = 物理介质长度
- $s$  = 介质传输速度  
( $\sim 2 \times 10^8$  m/sec)
- 传播延时 =  $d/s$

Note:  $R$ 、 $s$  是完全不同的概念! (动画)



# 类比



- ❑ 车辆移动速度  $100 \text{ km/hr}$
- ❑ 收费站  $12 \text{ sec}$  处理一辆车 (传输时间)
- ❑ 车  $\sim \text{bit}$ ; 车队  $\sim \text{packet}$
- ❑ **Q:** 一个车队多久抵达下一收费站?

- ❑ 车队通过收费站时间
- ❑  $= 12 * 10 = 120 \text{ sec}$
- ❑ 最后一辆车到达下一收费站的时间:  
 $100 \text{ km} / (100 \text{ km/hr}) = 1 \text{ hr}$
- ❑ **A: 62 minutes**

# 节点延迟

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

- $d_{\text{proc}}$  = 处理延时
  - ❖ 一般为 微秒 (microsecs)
- $d_{\text{queue}}$  = 排队延时
  - ❖ 决定于拥塞
- $d_{\text{trans}}$  = 传输延时
  - ❖  $= L/R$ , 低速链路明显
- $d_{\text{prop}}$  = 传播延时
  - ❖ 几微秒~几百微秒

# 排队延迟

- ❑  $R$ =带宽
- ❑ (link bandwidth, bps)
- ❑  $L$ =数据报长
- ❑ (packet length, bits)
- ❑  $a$ =平均到达速率
- ❑ (average packet arrival rate)

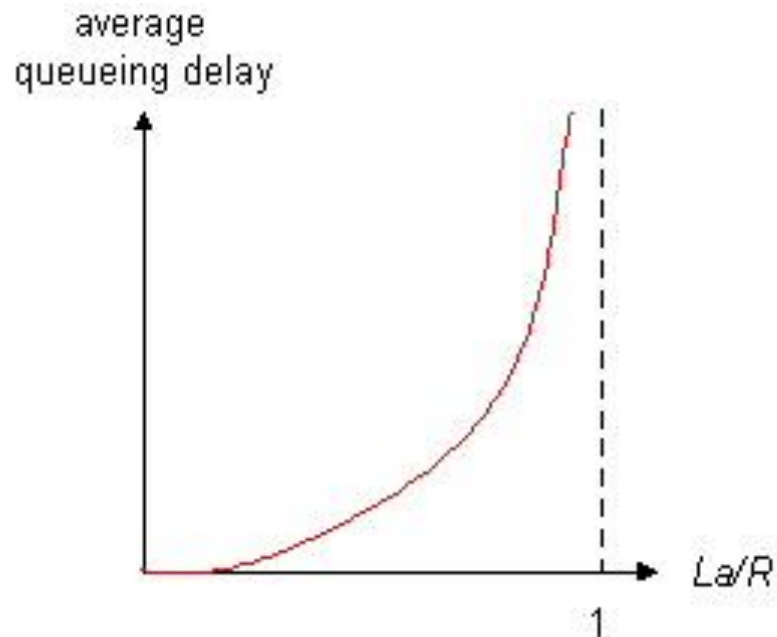
$$\text{流量强度} = La/R$$



# 排队延迟

- $R$ =带宽
- (link bandwidth, bps)
- $L$ =数据报长
- (packet length, bits)
- $\alpha$ =平均到达速率
- (average packet arrival rate)

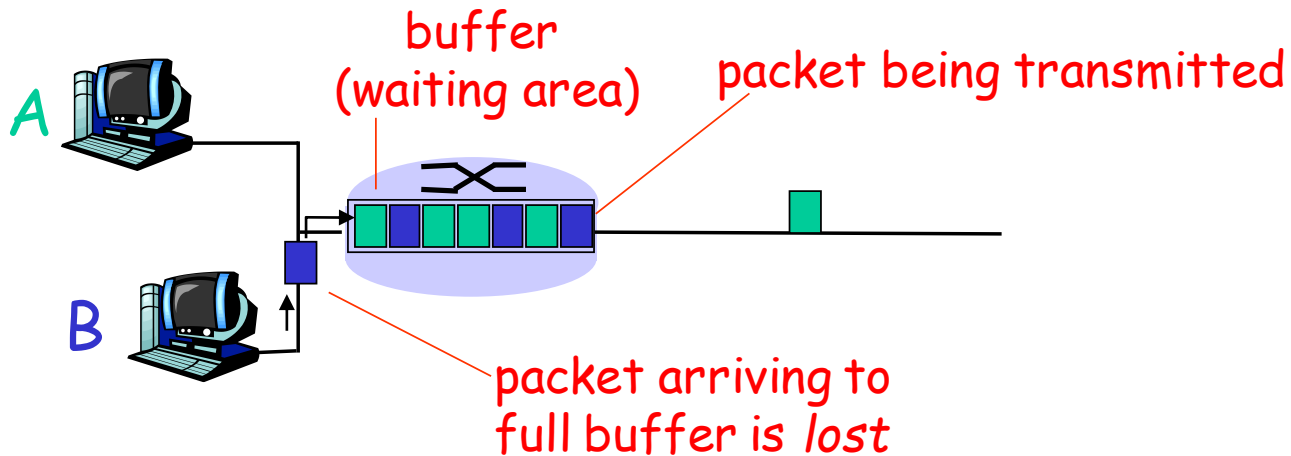
流量强度 =  $La/R$



- $La/R \sim 0$ : 平均排队延时较小
- $La/R \rightarrow 1$ : 延时变大
- $La/R > 1$ : 超过节点服务能力, 延时变的无限大!

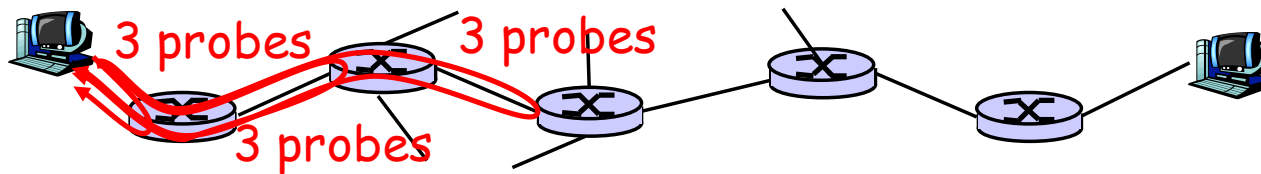
# 数据报丢失

- ❑ 队列所在的内存有限
- ❑ 队列满了以后到达的数据包会丢失(**lost**)
- ❑ 节点性能衡量：时延+丢包概率来度量
- ❑ 丢失的数据包会被重传, 或者根本不会被重传



# 真实的数据报延迟和路由


- ❑ Traceroute program (Tracert):
- ❑ 提供了源到目的地节点端到端路径、时延的检测功能：
  - ❖ 对途径的每个路由 $i$ 发送3个检测数据报；
  - ❖ 路由 $i$ 返回检测结果；
  - ❖ 发送、接收的时间间隔为时延，同时显示路由器 $i$ 的名字、地址。



# 真实的数据报延迟和路由


**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



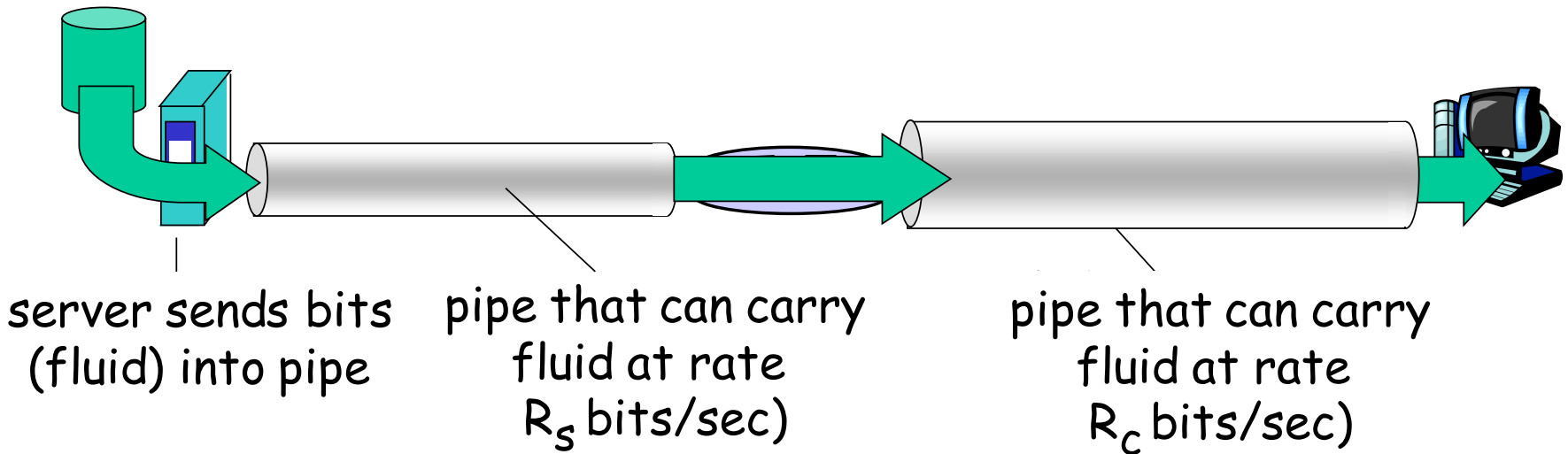
\* means no response (probe lost, router not replying)



# 吞吐率

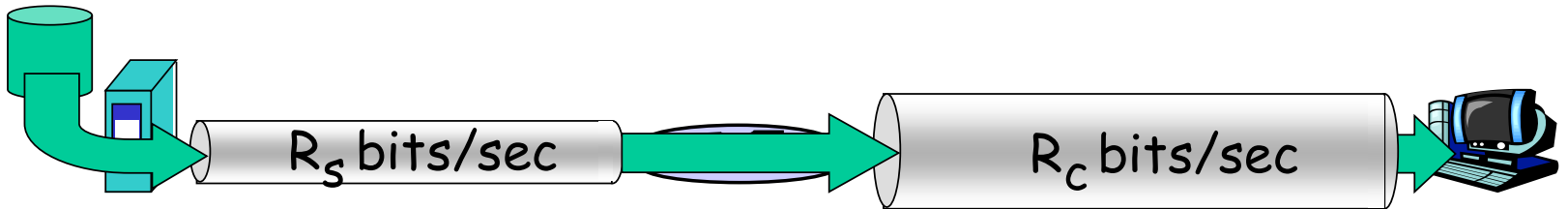
□ (*throughput*): 发送和接收者之间数据传输的速率 (bits/time unit)

- ❖ *即时*: 任何给定时间的速率
- ❖ *平均*: 较长时间内的平均速率

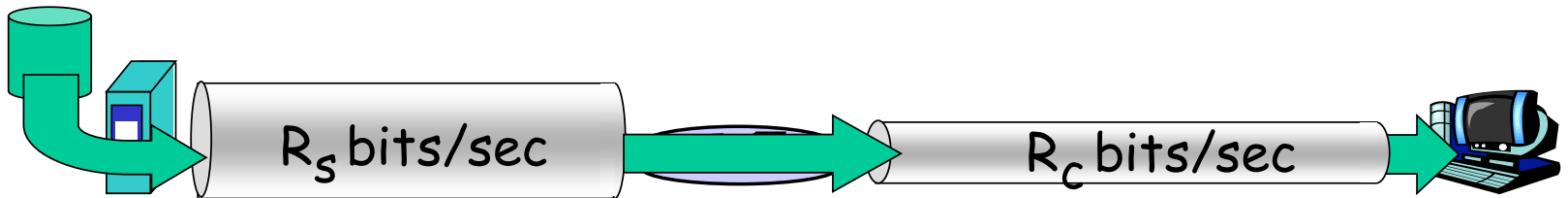


# 吞吐率(more)

□  $R_s < R_c$  哪个是端到端的平均速率?



□  $R_s > R_c$  哪个是端到端的平均速率?



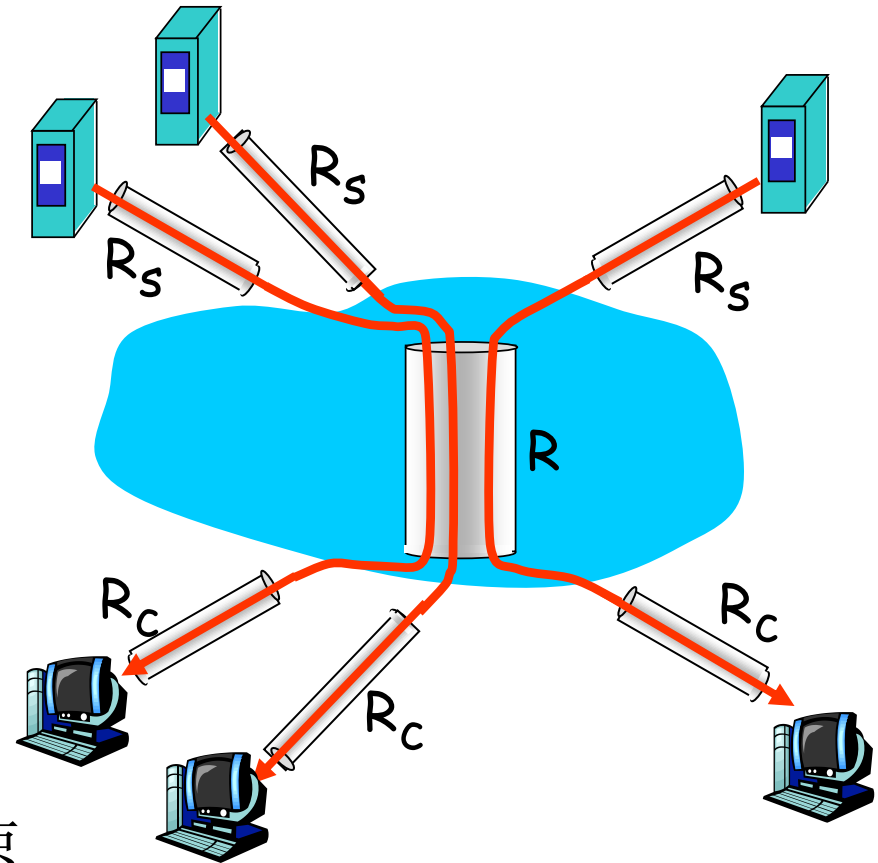
## 瓶颈链路

在路径上最终限制端到端吞吐率的链路

# Internet 中吞吐率:

- 单个连接的吞吐率:  
 $\min(R_c, R_s, R/10)$
- 现实生活中:
- $R_c$  or  $R_s$   
通常是瓶颈链接

取决于链路的最小传输速率、干扰流量



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

# Chapter 1: roadmap

## 1.1 什么是 Internet?

## 1.2 网络边缘

- 端系统, 接入网, 链路

## 1.3 网络核心

- 电路交换, 数据报交换, 网络结构

## 1.4 包交换网络中的延迟, 丢包和吞吐率

## 1.5 协议层次, 服务模型

## 1.6 网络攻击: 安全性

## 1.7 历史



# 协议层次

## 网络非常复杂!

- many "pieces":
  - ❖ hosts
  - ❖ routers
  - ❖ links of various media
  - ❖ applications
  - ❖ protocols
  - ❖ hardware, software

## Question:

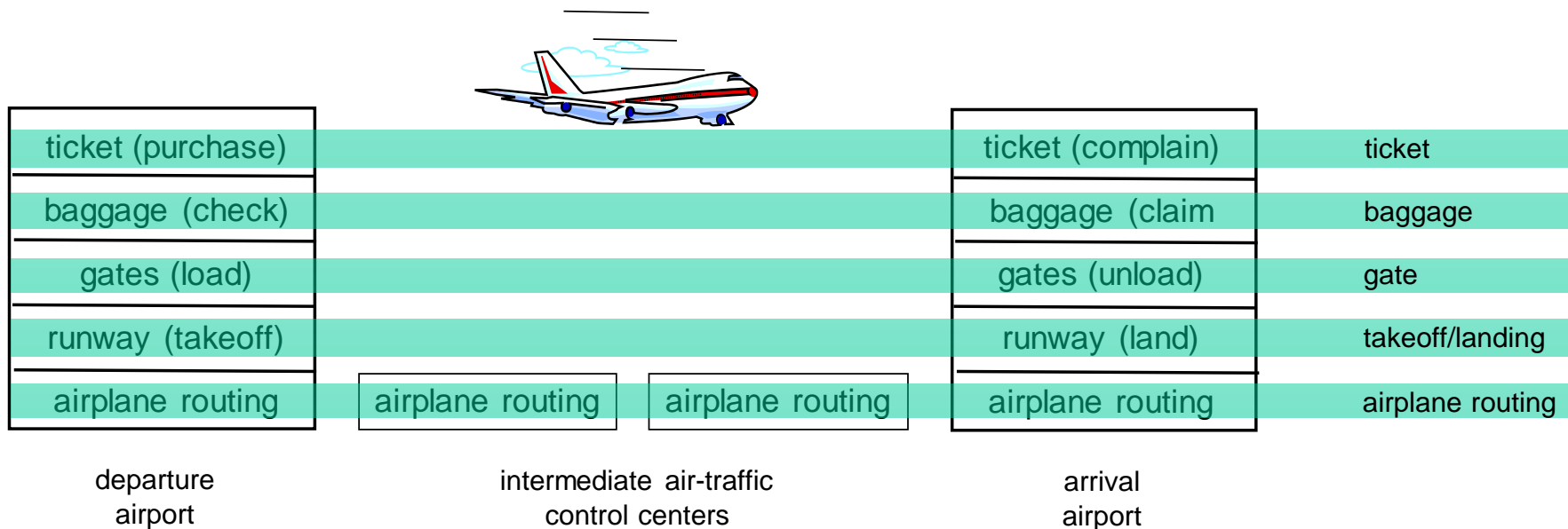
网络有一定的组织结构吗?  
至少在我们学习这门课的时候?

# 机场的登机顺序



## □ 一系列步骤

# 航空运输的层次



**Layers:** 每一层次都实现一定的功能

- ❖ 层次内部实现一定功能
- ❖ 依赖于其他层次提供的服务

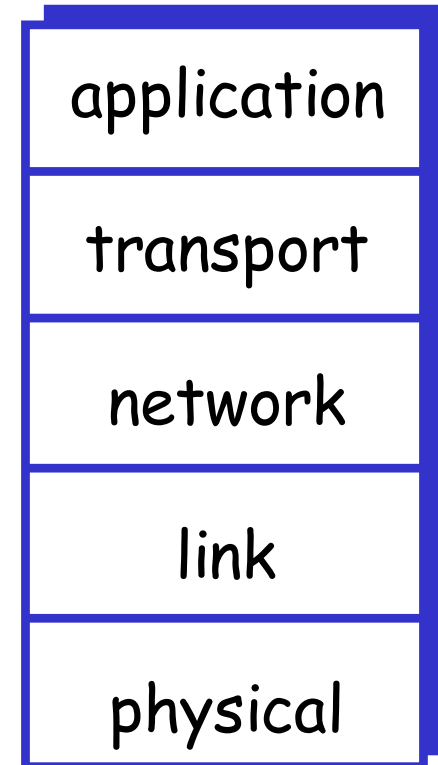
# 为什么分层?

处理复杂的问题:

- ❑ 简化处理:
- ❑ 准确的结构定义能够区分、界定复杂系统各个组成部门的功能和边界
  - ❖ 层次化参考模型(**reference model**) 提供讨论
- ❑ 便于维护:
- ❑ 模块化易于维护,更新系统
  - ❖ 某层服务功能的实施、更新独立于其它层次,提高结构稳定性
  - ❖ *e.g.*, 登机口的改变不会影响旅客接受的服务质量
- ❑ 好像也没其它坏处?

# Internet 协议栈

- 应用层(application): 报文
  - ❖ 支持网络应用
  - ❖ FTP, SMTP, HTTP
- 传输层(transport): 报文段
  - ❖ process-process 数据传输
  - ❖ TCP, UDP
- 网络层(network): 数据报
  - ❖ 从源到目的地路由数据包
  - ❖ IP, routing protocols
- 链路层(link): 帧
  - ❖ 相邻两个网络设备之间传输数据
  - ❖ PPP, Ethernet
- 物理层(physical):
  - ❖ bits "on the wire"



# ISO/OSI 参考模型

## □ 表示层(presentation):

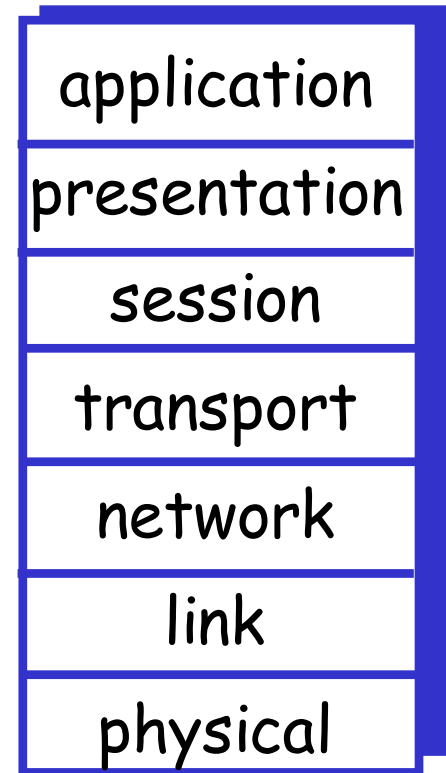
- ❖ 允许应用程序解释数据含义
- ❖ , e.g., 加密, 压缩, 特定编码

## □ 会话层(session):

- ❖ 同步, 数据恢复, 会话状态

## □ Internet 没有这两层!

- ❖ 认为这些服务应该在应用层中实现



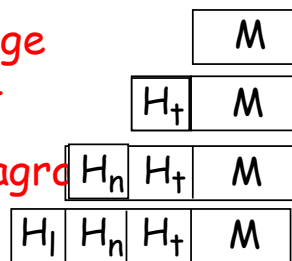
# 封装 Encapsulation

消息message

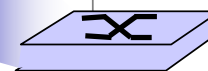
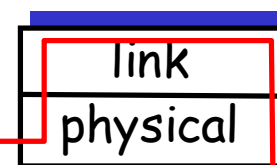
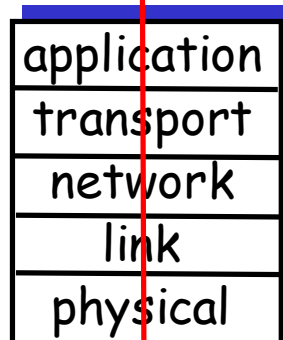
段segment

数据报datagram

帧frame

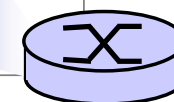
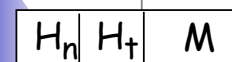
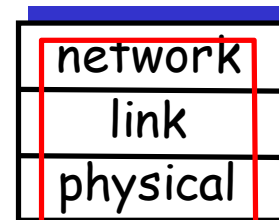
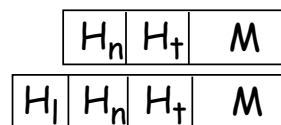
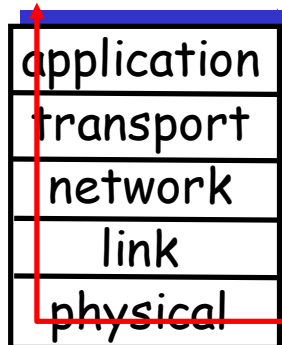


source



switch

destination

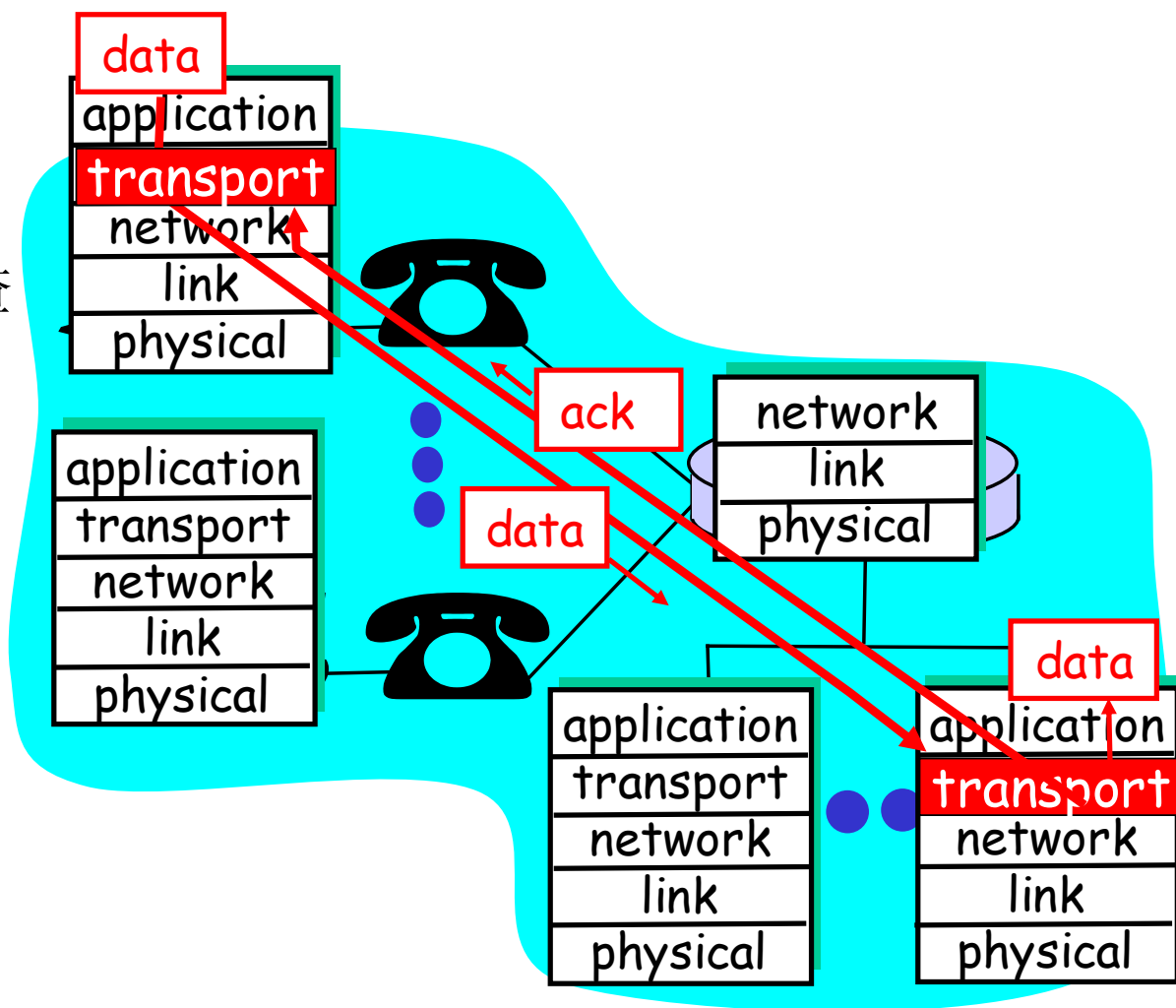


router

# 层次化逻辑通信

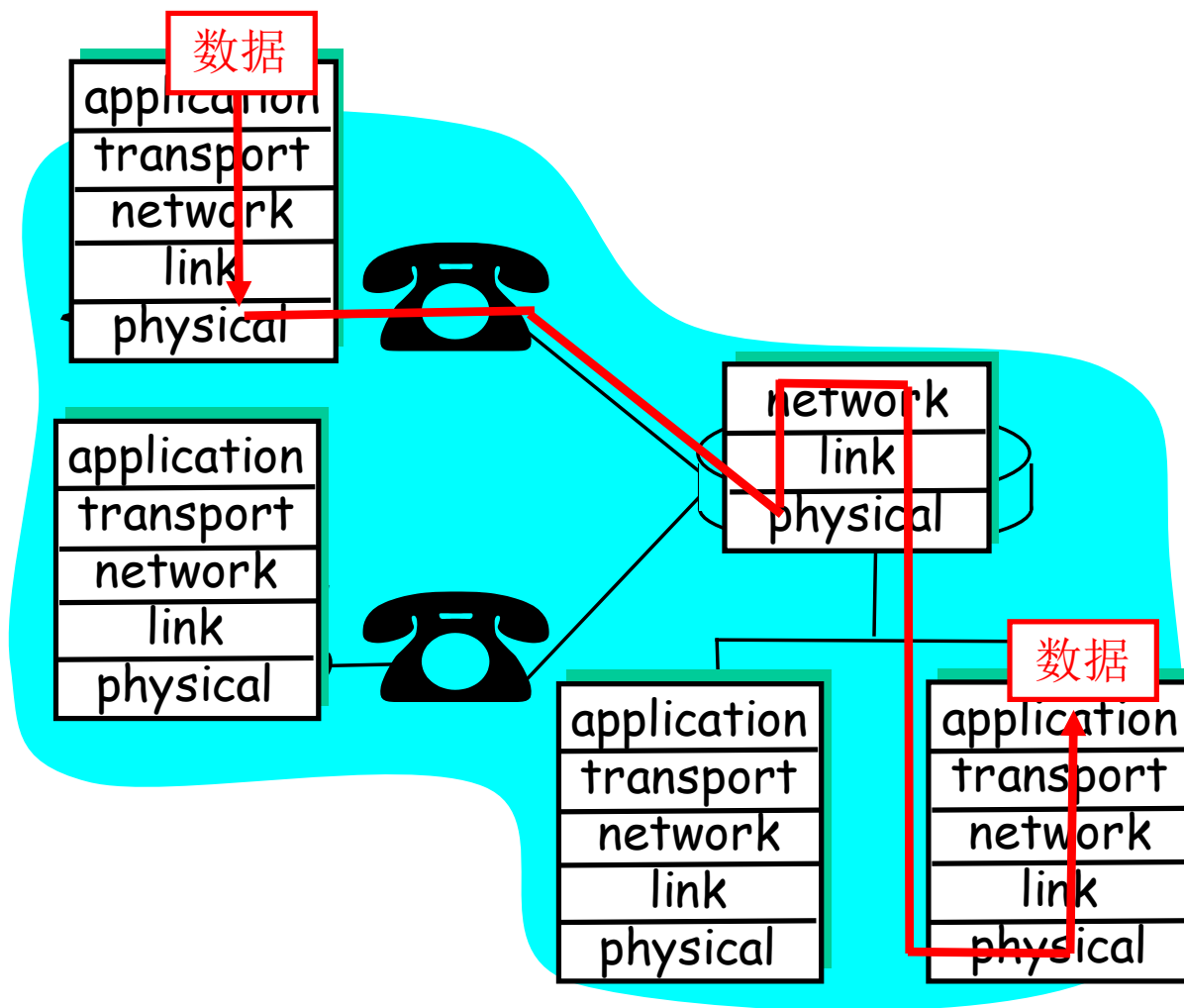
## E.g.: 传输层

- ❑ 从应用接收数据报
- ❑ 加上地址, 可靠性检查信息
- ❑ 发送报文段到对等端
- ❑ 等对等端的确认消息
- ❑ 类比: 邮局





# 层次化物理通信





# Chapter 1: roadmap

## 1.1 什么是 Internet?

## 1.2 网络边缘

- 端系统, 接入网, 链路

## 1.3 网络核心

- 电路交换, 数据报交换, 网络结构

## 1.4 包交换网络中的延迟, 丢包和吞吐率

## 1.5 协议层次, 服务模型

## 1.6 网络攻击: 安全性

## 1.7 历史

# 网络安全

## □ 网络安全问题是关于:

- ❖ 坏人如何攻击计算机网络
- ❖ 如何防卫
- ❖ 如何设计网络架构能够防御攻击

## □ Internet 设计时没有考虑安全问题

- ❖ 最初的设想是:
- ❖ “透明网络中一群互相信任的用户” ☺
- ❖ Internet 协议设计者采用“catch-up”的策略
- ❖ 每一个层次都有防御策略!

# 信息安全

## □ 保密性

- ❖ 具有一定保密程度的信息只能让有权读到或更改的人读到和更改。

## □ 完整性

- ❖ 在存储或传输信息的过程中，原始的信息不能允许被随意更改。

## □ 可用性

- ❖ 对于信息的合法拥有和使用用户，在他们需要这些信息的任何时候，都应该保障他们能够及时得到所需要的信息。

## □ 不可抵赖性

- ❖ 对自己行为的不可抵赖及对行为发生的时间的不可抵赖。

# 行业主管部门及监管体制

## □ 中央网信办：

- 统筹协调各个领域的网络安全和信息化重大问题，研究制定网络安全和信息化发展战略、宏观规划和重大政策，推动国家网络安全和信息化法治建设

## □ 国家发改委：

- 负责产业政策、产业规划的研究制定、行业的管理与规划等

## □ 工信部：

- 工信部下属信息化和软件服务业司负责与软件服务业相关的指导工作，信息安全协调司为电子认证服务业主管单位

## □ 公安部：

- 主管全国计算机信息系统安全保护工作，信息安全及等级保护的监督管理工作和计算机信息系统安全销售许可核准工作

# 行业主管部门及监管体制

## □ 国家密码管理局

- 主管全国商用密码管理工作，批准生产的商用密码产品品种和型号等

## □ 国家保密局

- 管理和指导保密技术工作，负责办公自动化和计算机信息系统的保密管理，指导保密技术产品的研制和开发应用，对从事涉密信息系统集成的企业资质进行认定

## □ 国家国防科工局

- 负责管理国防科技工业的行政管理机关,负责核、航天、航空、船舶、兵器、电子等领域武器装备科研生产重大事项的组织协调和军工核心能力建设

## □ 国家版权局

- 主管全国新闻出版事业与著作权管理工作，在公司所处的行业负责软件著作权的管理工作

# 行业的主要法律法规及政策

- 2005 年《中华人民共和国电子签名法》
- 2015 年《中华人民共和国国家安全法》
  - 提出要建设网络与信息安全保障体系，提升网络与信息安全保护能力，加强网络和信息技术的创新研究和开发应用，实现网络和信息核心技术、关键基础设施和重要领域信息系统及数据的安全可控
- 2017年《中华人民共和国网络安全法》
  - 明确了网络空间主权的原则、网络安全与信息化发展并重原则和共同治理原则、网络产品和服务提供者的安全义务、网络运营者的安全义务，进一步完善了个人信息保护规则，建立了关键信息基础设施安全保护制度以及关键信息基础设施重要数据跨境传输的规则。
- 2018年《网络安全等级保护条例（征求意见稿）》



# 恶意软件 通过 Internet 进入主机

- ❑ 恶意软件(Malware)

可能是 病毒(virus), 蠕虫(worm), 或木马(trojan horse).

- ❑ 间谍恶意软件(Spyware malware)

记录键盘, 访问过的网址, 上传用户信息.

- ❑ 僵尸网络(Botnet)

感染的主机组成僵尸网络, 发动spam 和 DDoS 攻击.

- ❑ 特点: 自我复制

- ❑ 恶意软件通常能够自我复制: 从感染的主机, 伺机进入其他有类似缺陷的主机

# 通过Internet将恶意软件注入主机

## □ Trojan horse

- ❖ 它是指用于控制另一台计算机的一段特定的程序（木马程序）
- ❖ 控制端；即被控制端
- ❖ 通常潜伏于网页  
(Active-X, plugin)

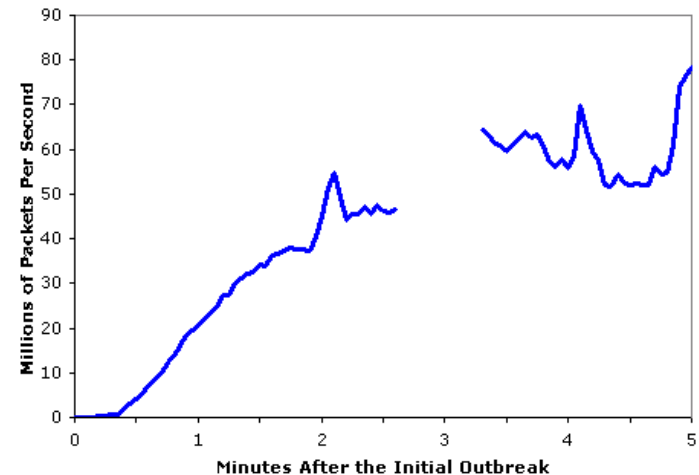
## □ Virus

- ❖ 破坏计算机功能或者数据的代码
- ❖ 自我复制的一组计算机指令或者程序代码

## □ Worm:

- ❖ 传播它自身功能的拷贝或某些部分到其他的计算机系统
- ❖ 网络或者系统漏洞进行传播
- ❖ 自我复制：感染其他用户、主机

Sapphire Worm: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)



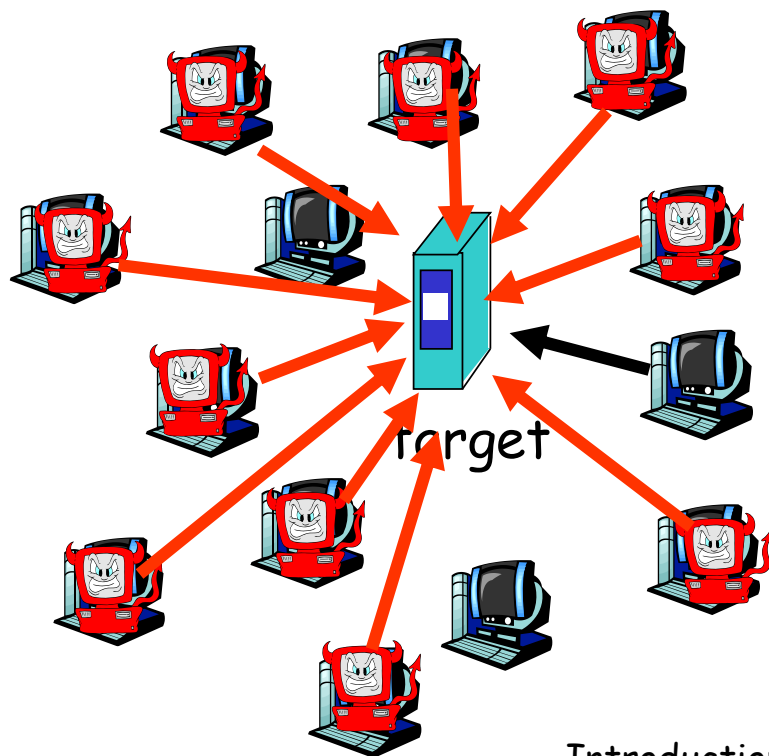
# 服务器和网络架构的攻击

- ❑ 拒绝服务攻击 (Digital Attack Map: <https://www.digitalattackmap.com/>)
- ❑ Denial of service (DoS): 攻击者通常使用过量的服务请求或流量, 使资源(server, bandwidth) 不可用

1. 选择目标
2. 闯入周边的主机
3. 发送数据包到目标主机

类型:

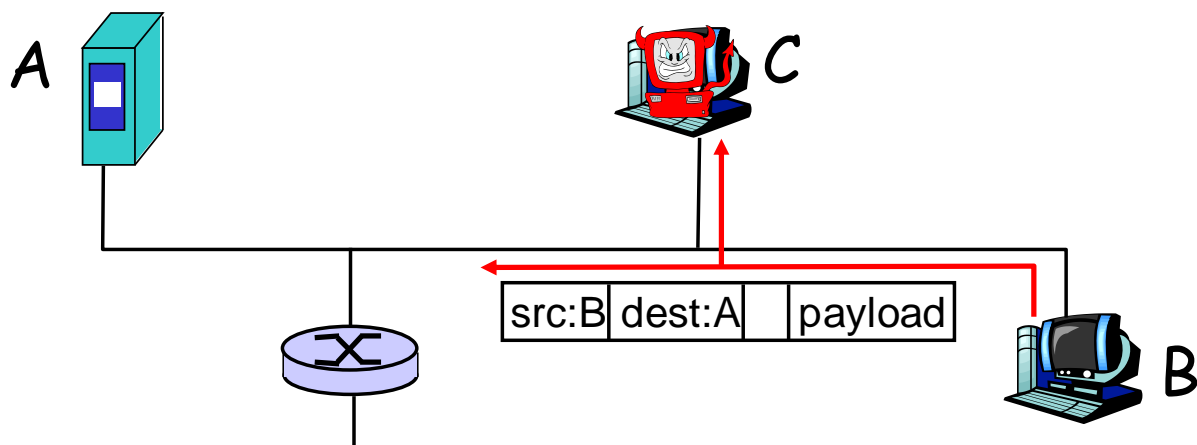
1. 弱点攻击
2. 带宽洪泛
3. 连接洪泛



# 数据报监听

## *Packet sniffing:*

- ❖ 广播介质(共享Ethernet, 无线局域网)
- ❖ 在网络接口混杂模式下读取/记录所有经过的数据报

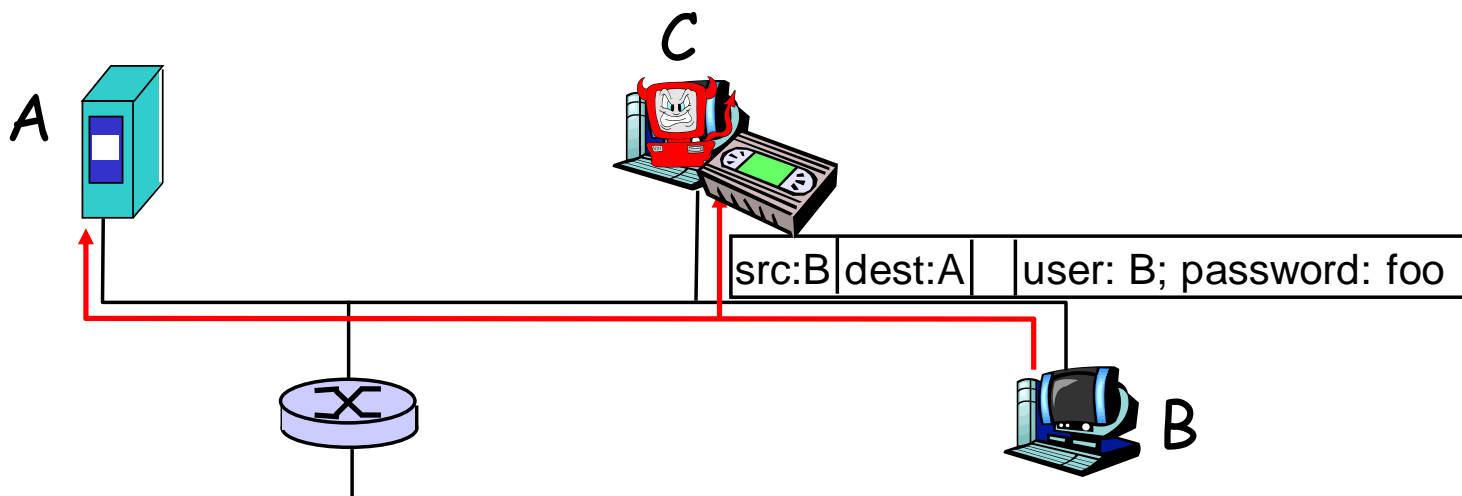


- ❖ Wireshark : 一个免费的sniffer, 可以用它来学习网络

# 重放攻击

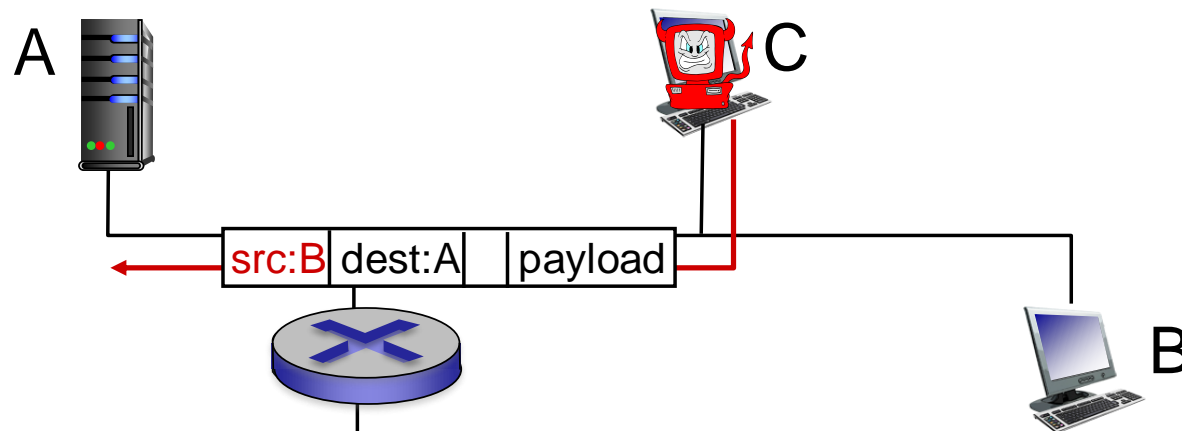
## ❑ *record-and-playback*:

- ❖ 监听敏感信息(e.g., password), 在以后使用
- ❖ 或录制正常数据, 通过重放, 麻痹网络管理员



# 伪装地址

*IP spoofing*: send packet with false source address



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

# Chapter 1: roadmap

1.1 什么是 Internet?

1.2 网络边缘

- end systems, access networks, links

1.3 网络核心

- circuit switching, packet switching, network structure

1.4 包交换网络中的延迟, 丢包和吞吐率

1.5 协议层次, 服务模型

1.6 网络攻击: 安全性

1.7 History

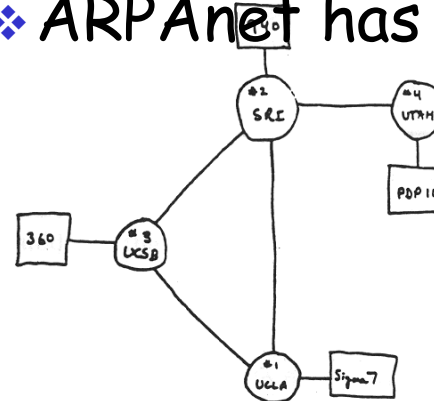
# Internet history

## 1961-1972: 早期分组交换技术(三个小组)

- 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



# Internet history

## 1972-1980: 专用网络和网络互联

- ❑ 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)

### 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

# Internet history

## *1980-1990: 新协议, 网络激增时代*

- ❑ 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(法国)
- 100,000 hosts connected to confederation of networks

# Internet history

*1990, 2000 's: 因特网爆炸时代, 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

# Internet history

## *2005-present*

- ❑ ~5B devices attached to Internet (2016)
  - ❖ smartphones and tablets
- ❑ aggressive deployment of broadband access
- ❑ increasing ubiquity of high-speed wireless access
- ❑ emergence of online social networks:
  - ❖ Facebook: ~ one billion users
- ❑ service providers (Google, Microsoft) create their own networks
  - ❖ bypass Internet, providing “instantaneous” access to search, video content, email, etc.
- ❑ e-commerce, universities, enterprises running their services in “cloud” (e.g., Amazon EC2)

# Introduction: summary

*covered a “ton” of material!*

- ❑ Internet overview
- ❑ what's a protocol?
- ❑ network edge, core, access network
  - ❖ packet-switching versus circuit-switching
  - ❖ Internet structure
- ❑ performance: loss, delay, throughput
- ❑ layering, service models
- ❑ security
- ❑ history

*you now have:*

- ❑ context, overview, “feel” of networking
- ❑ more depth, detail to follow!