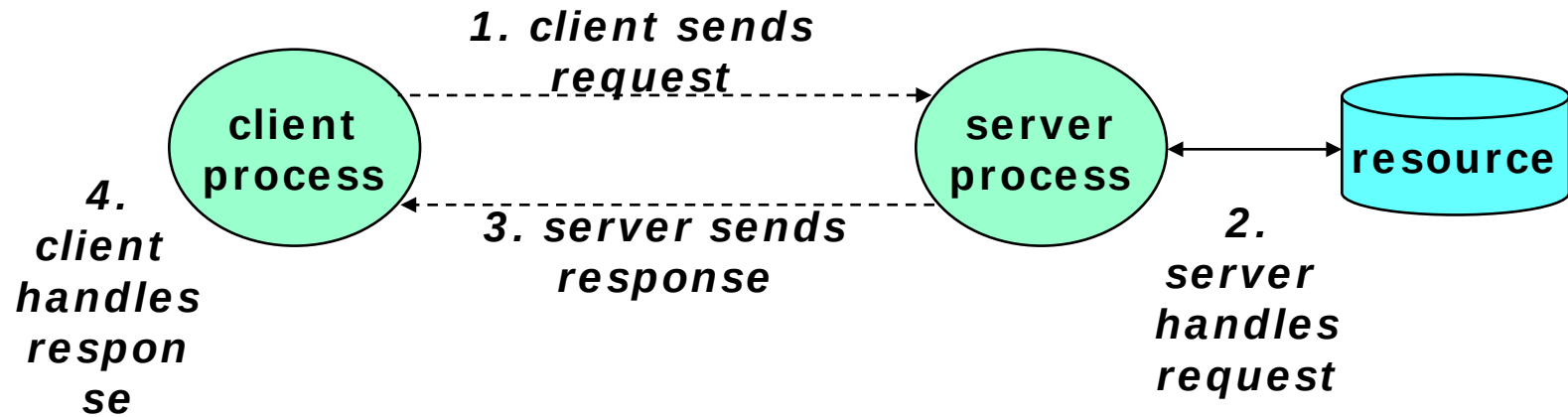


# **Networking Programming**

# A client-server transaction

---



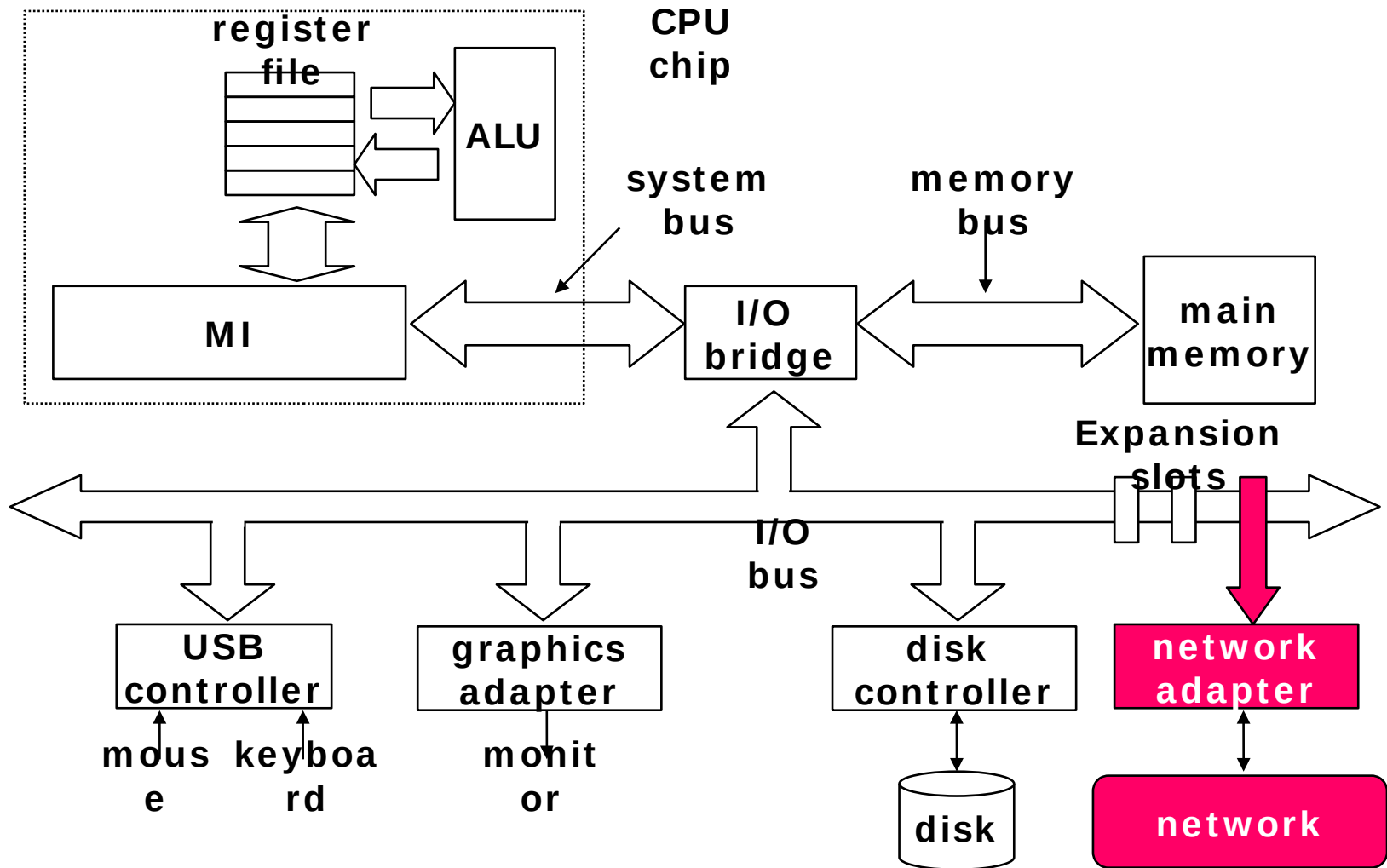
**Note: clients and servers are processes running on hosts (can be the same or different hosts).**

# Outline

---

- **计算机网络的组成部分**
  - 终端（硬件 + 软件）
  - 连接
  - 网络设备
  - 协议
  - 网络拓扑
- TCP/IP 协议栈

# Hardware organization of a network host



# 计算机网络的组成部分

---

- #1. 终端（硬件 + 软件）
- **#2. 连接**
- #3. 网络拓扑
- #4. 协议
- #5. 网络设备

# 连接

---

## *twisted pair (TP)*

- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps



*Wireless connections*

## *fiber optic cable:*

- ❖ glass fiber carrying light pulses, each pulse a bit
- ❖ high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gbps transmission rate)
- ❖ low error rate:



# Wireless access networks

---

- shared *wireless* access network connects end system to router
  - via base station aka “access point”

## **wireless LANs:**

- within building (100 ft)
- 802.11b/g (WiFi): 11, 54 Mbps transmission rate



to  
Internet

## **wide-area wireless access**

- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G 4G: LTE



to

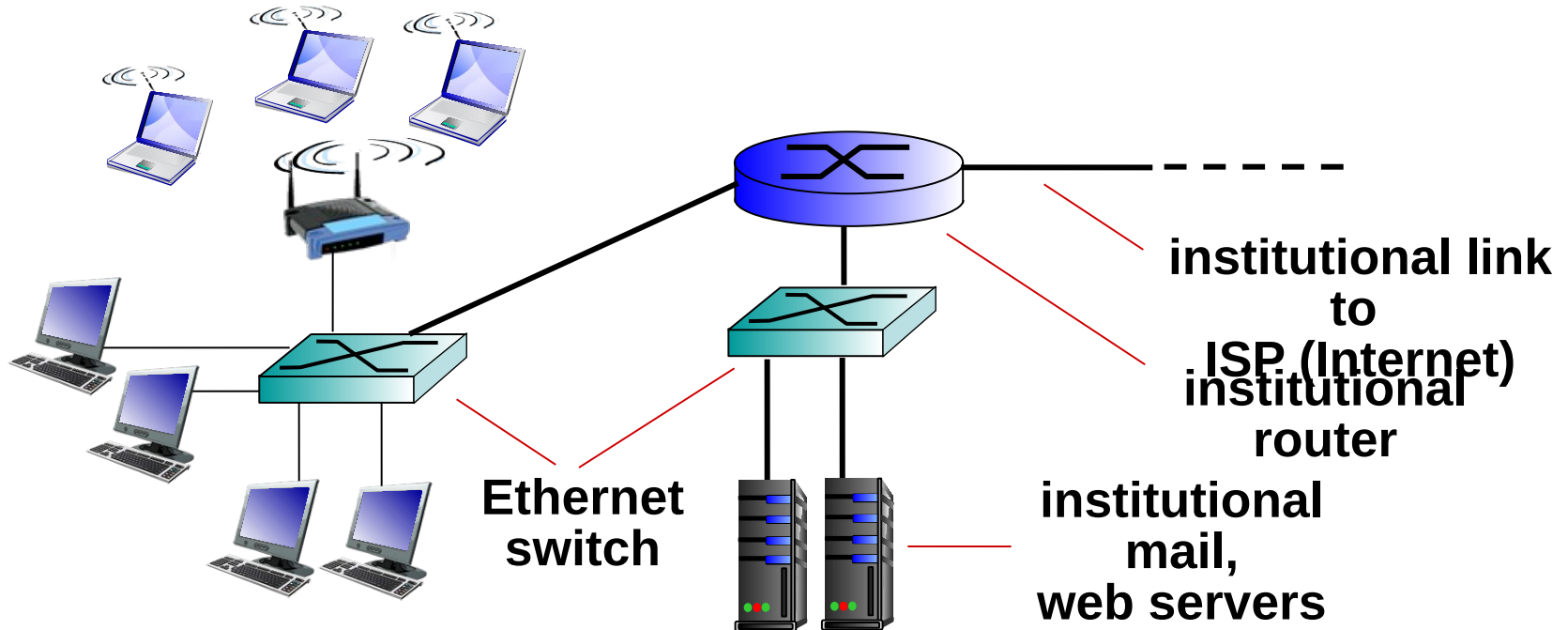
# 计算机网络的组成部分

---

- #1. 终端（硬件 + 软件）
- #2. 连接
- **#3. 网络拓扑**
- #4. 协议
- #5. 网络设备

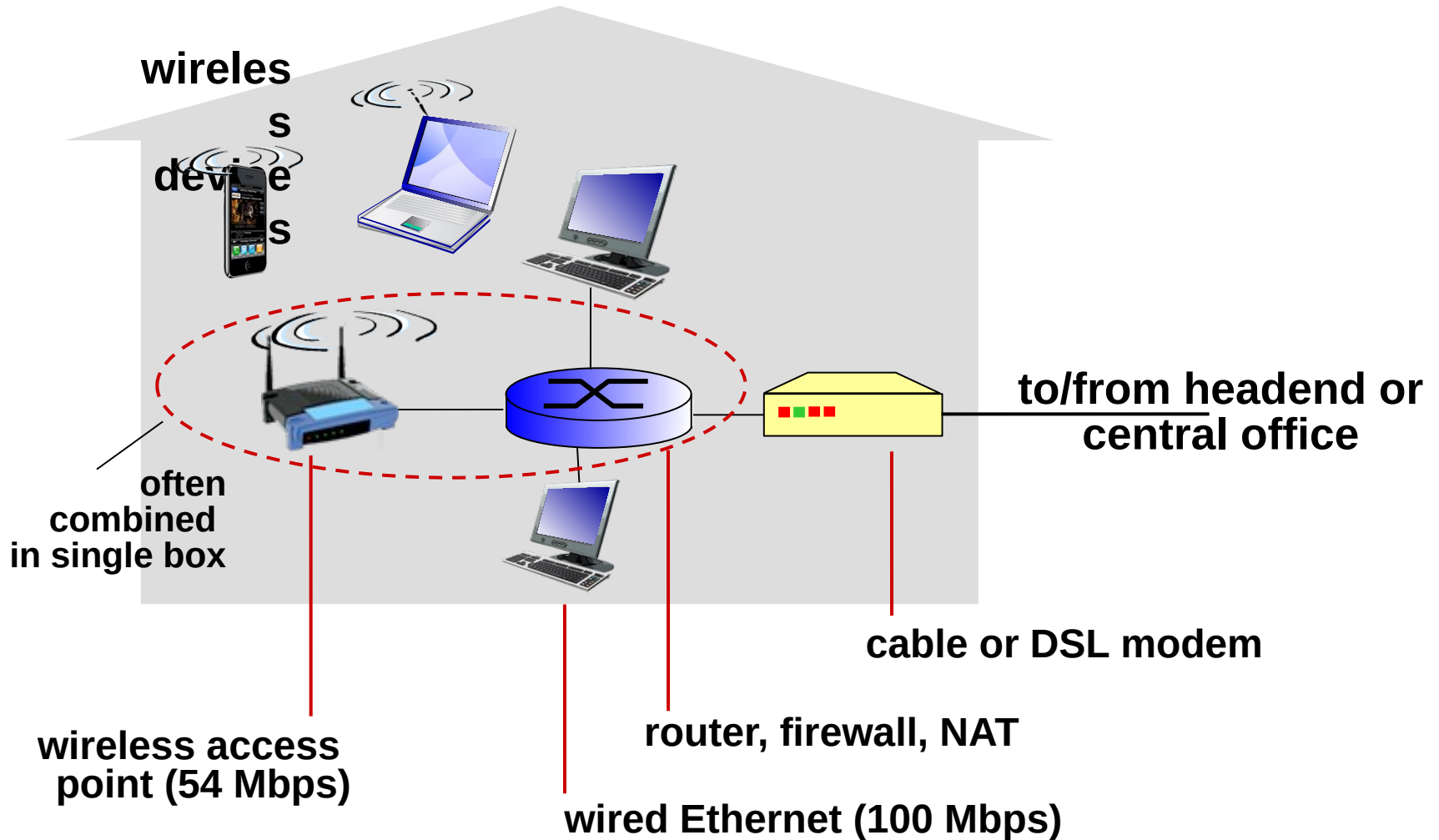
# Enterprise access networks (Ethernet)

---



- typically used in companies, universities, etc
- ❖ 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- ❖ today, end systems typically connect into Ethernet switch

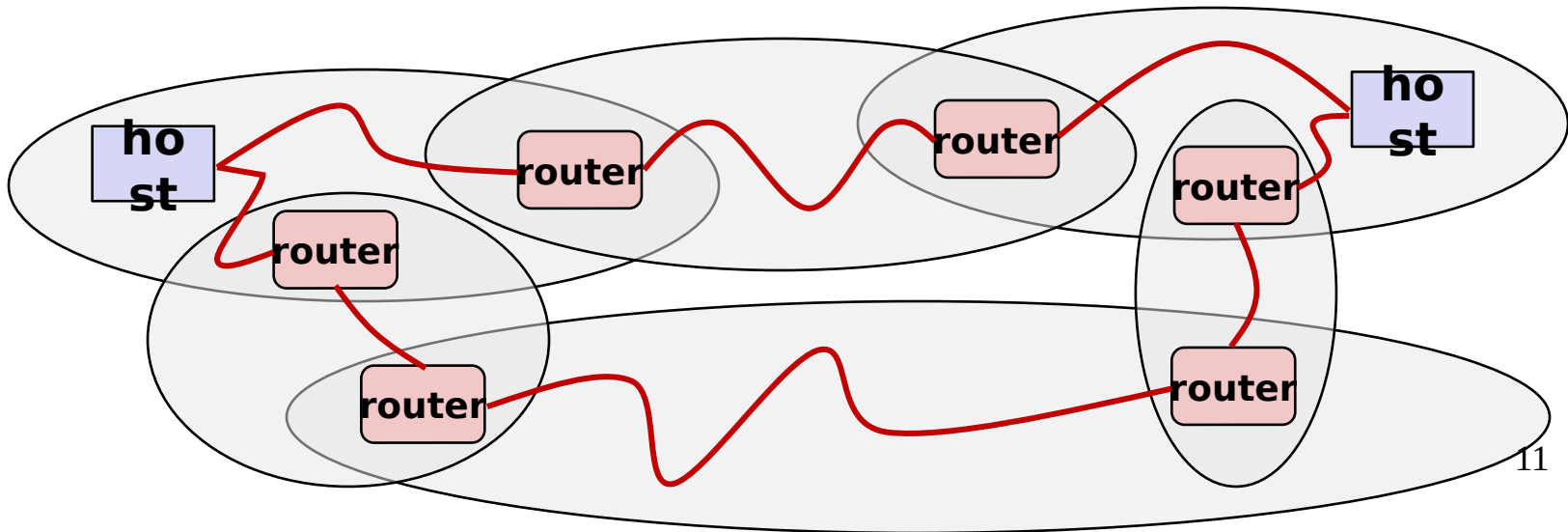
# Access net: home network



# Logical Structure of an internet

---

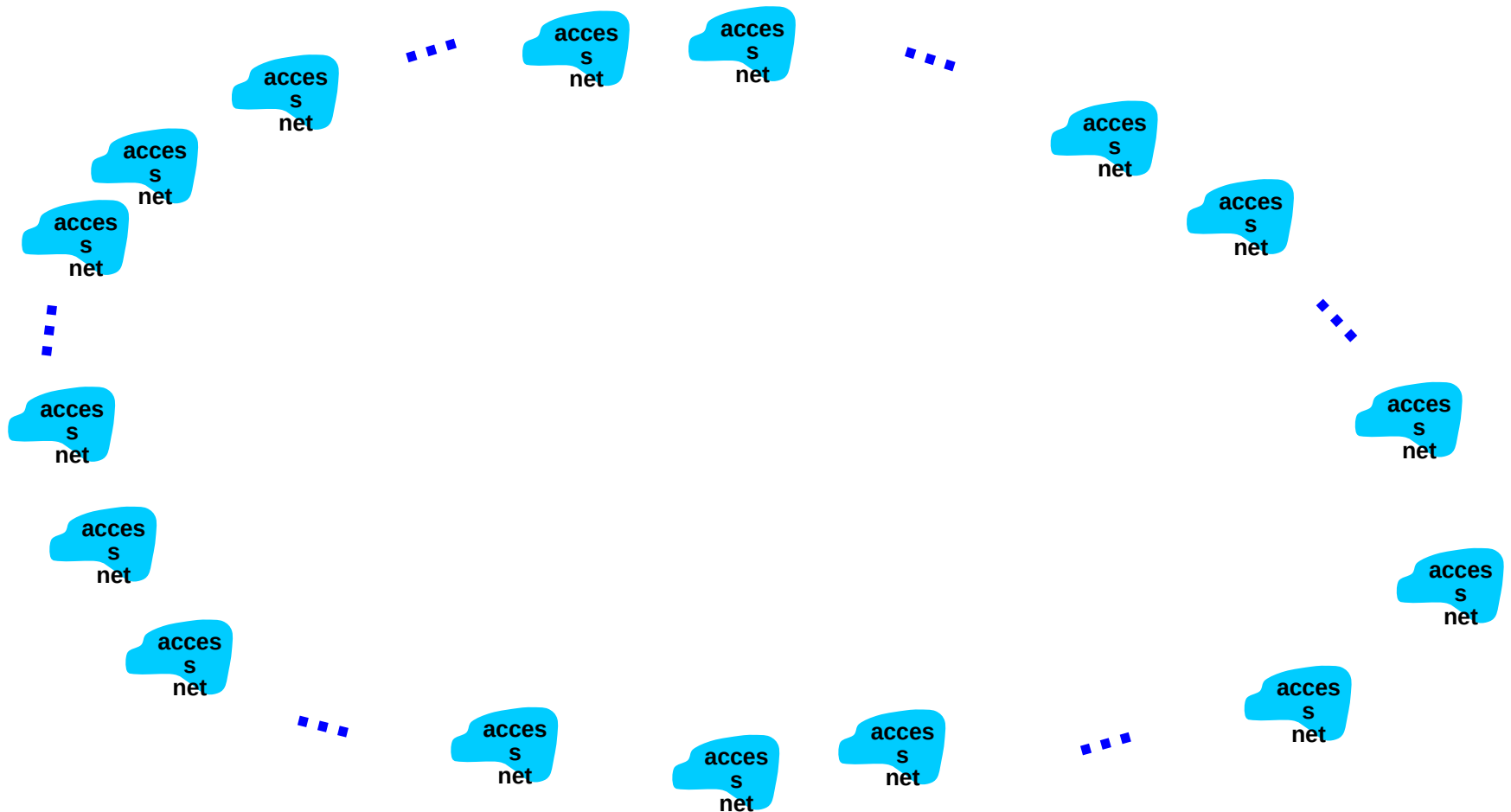
- **Ad hoc** interconnection of networks
  - No particular topology
  - Vastly different router & link capacities
- Send packets from source to destination by hopping through networks
  - Router forms bridge from one network to another



# Internet structure: network of networks

---

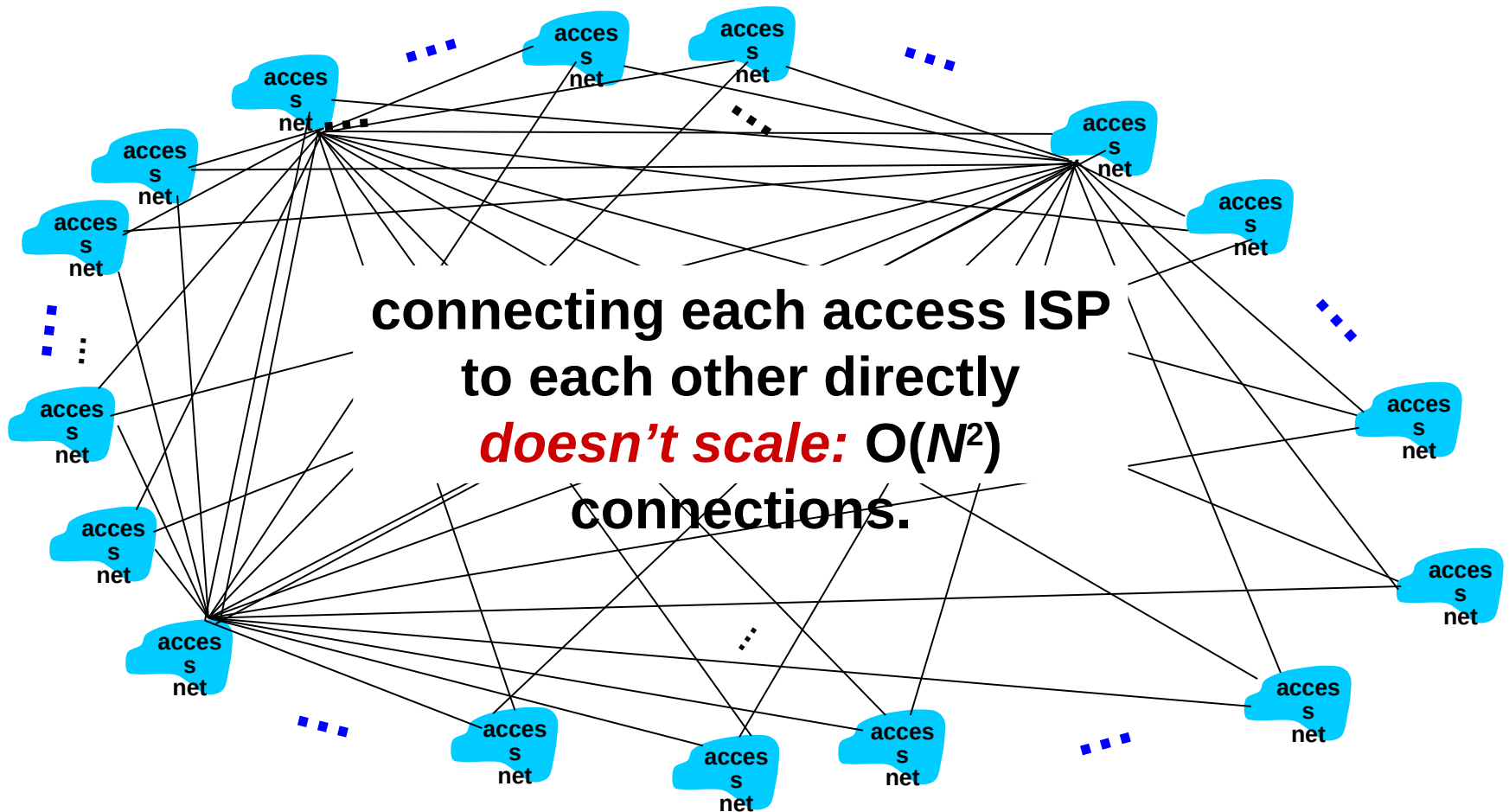
*Question:* given millions of access ISPs, how to connect them together?



# Internet structure: network of networks

---

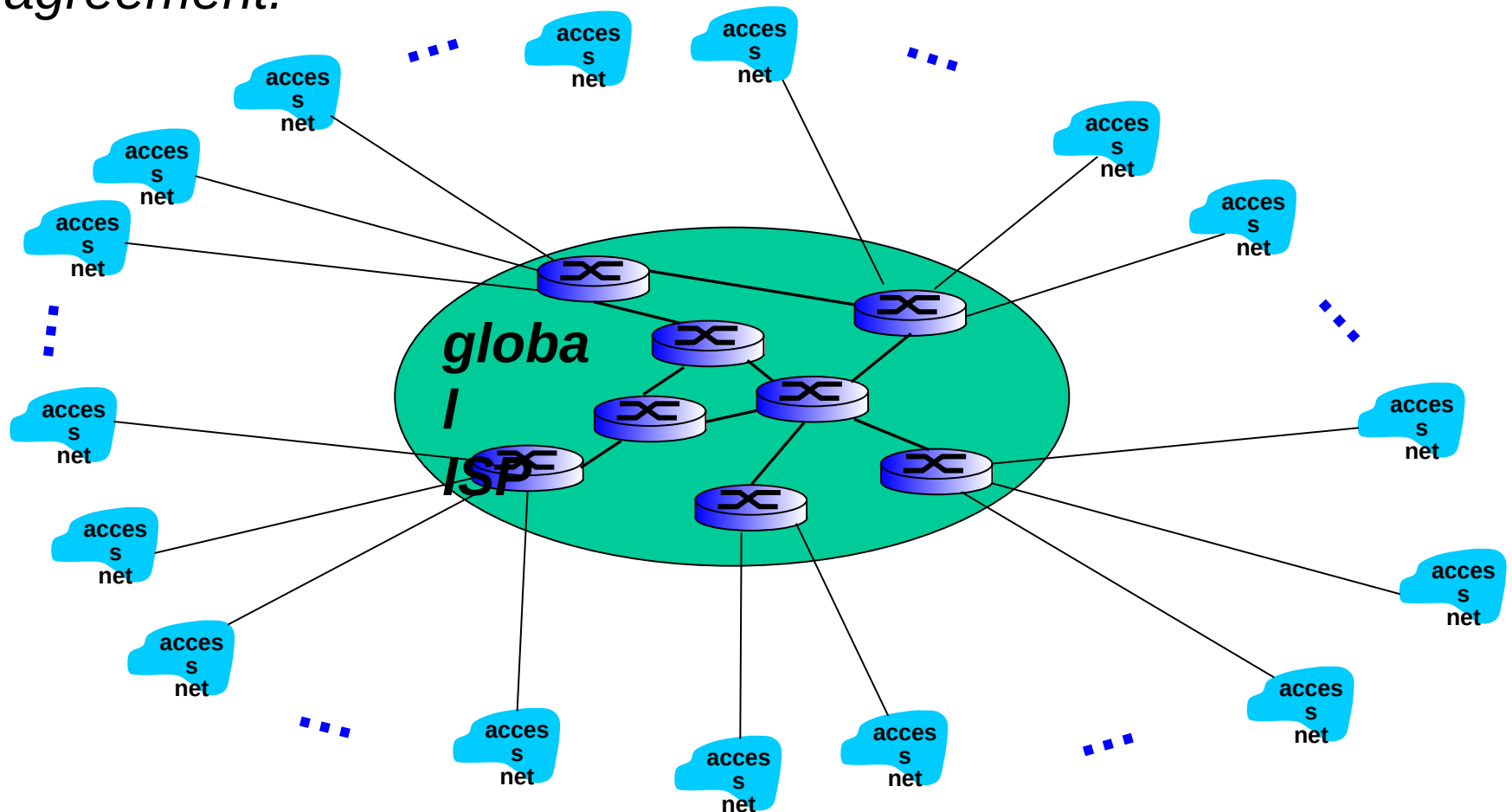
*Option:* connect each access ISP to every other access ISP?



# Internet structure: network of networks

---

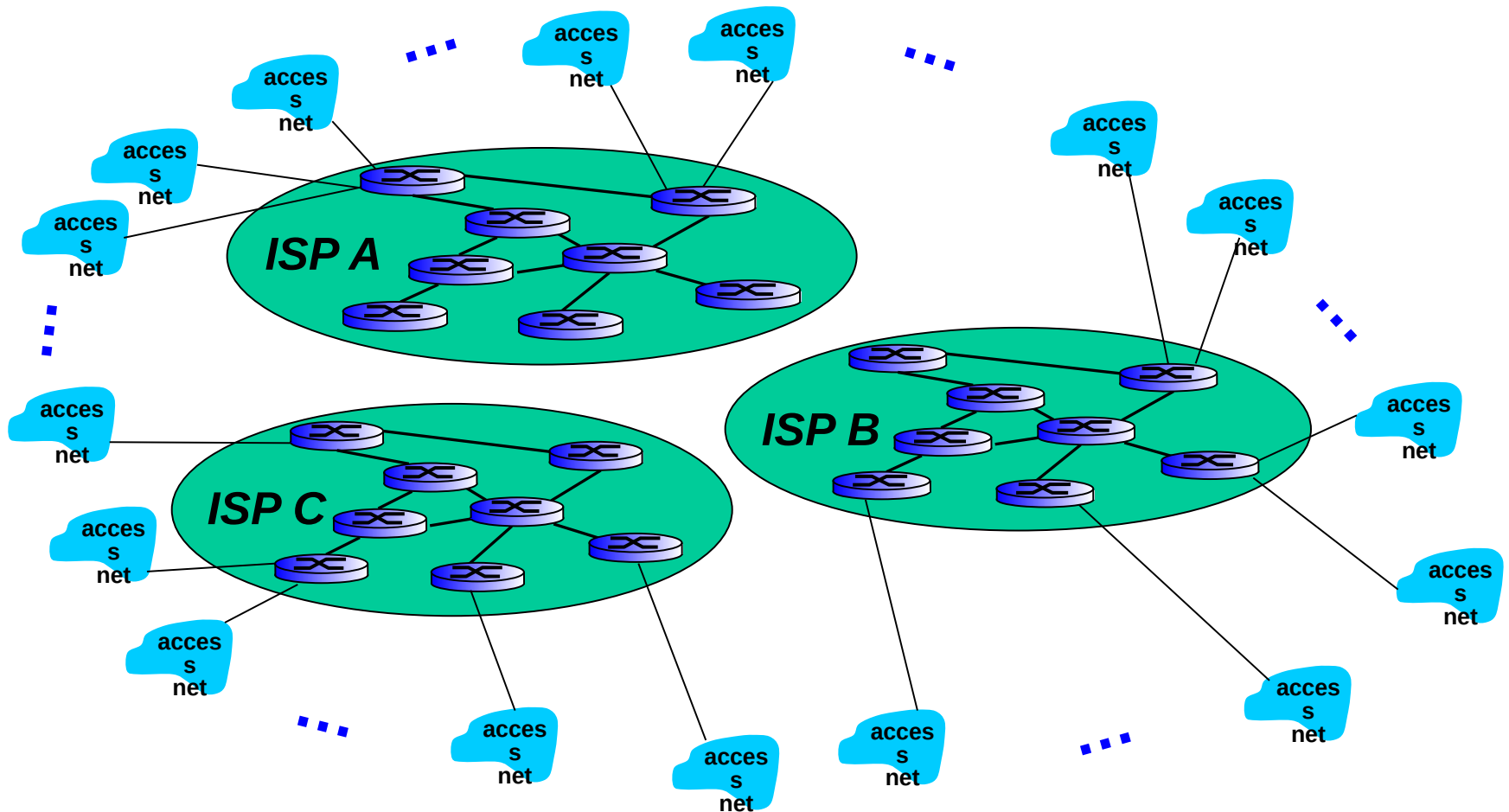
*Option: connect each access ISP to a global transit ISP? Customer and provider ISPs have economic agreement.*



# Internet structure: network of networks

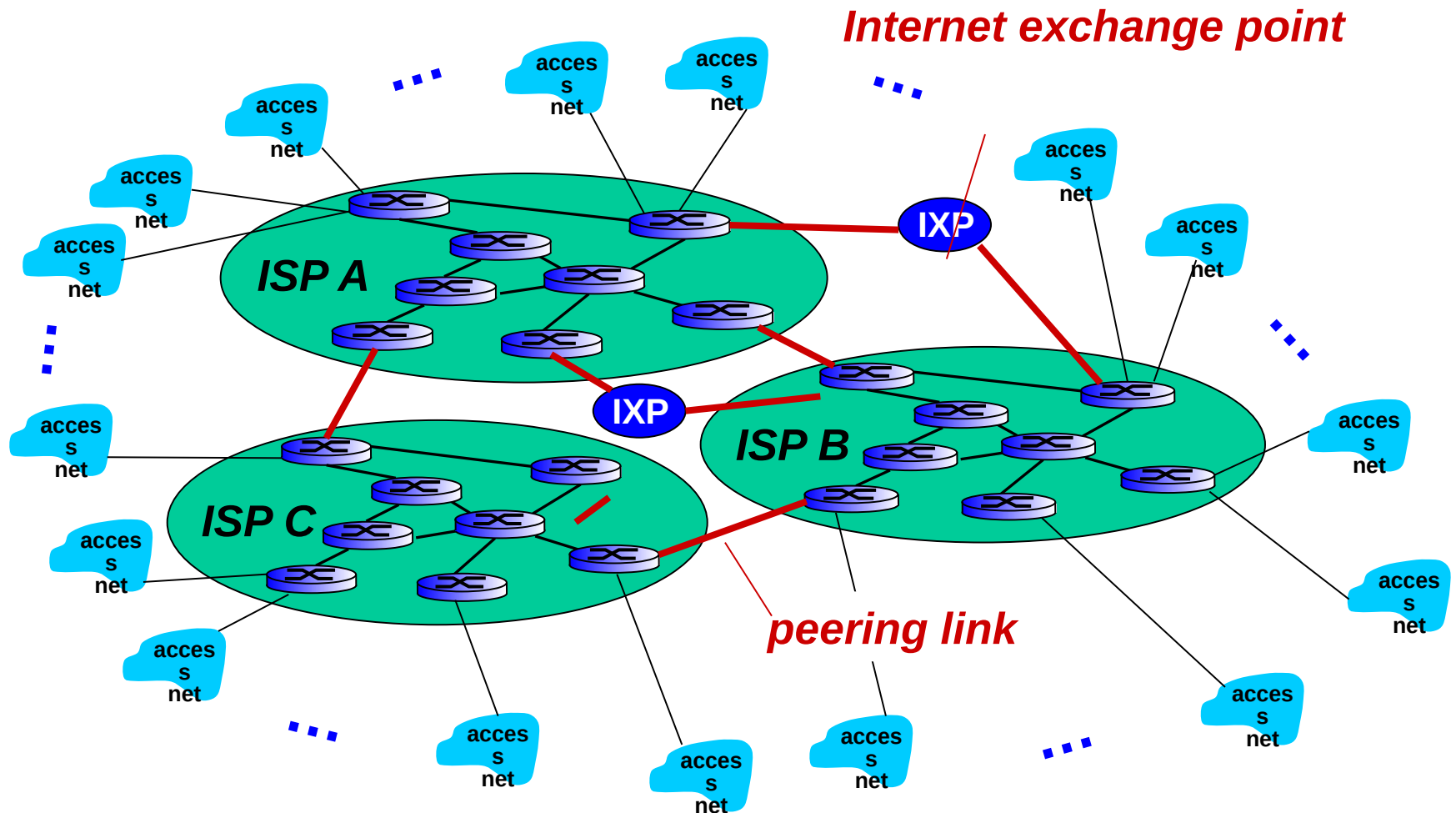
---

But if one global ISP is viable business, there will be competitors ....



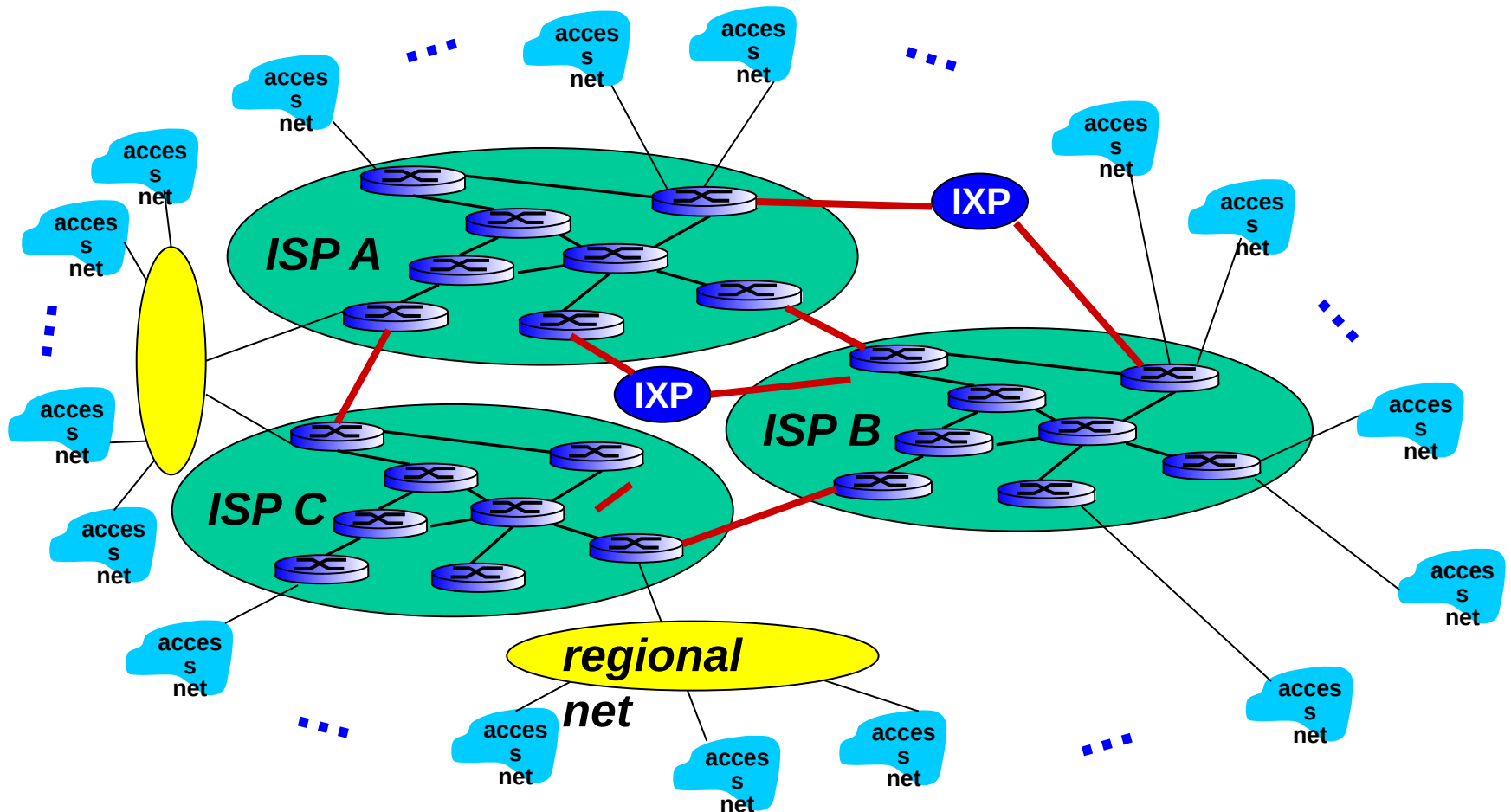
# Internet structure: network of networks

But if one global ISP is viable business, there will be competitors .... which must be interconnected



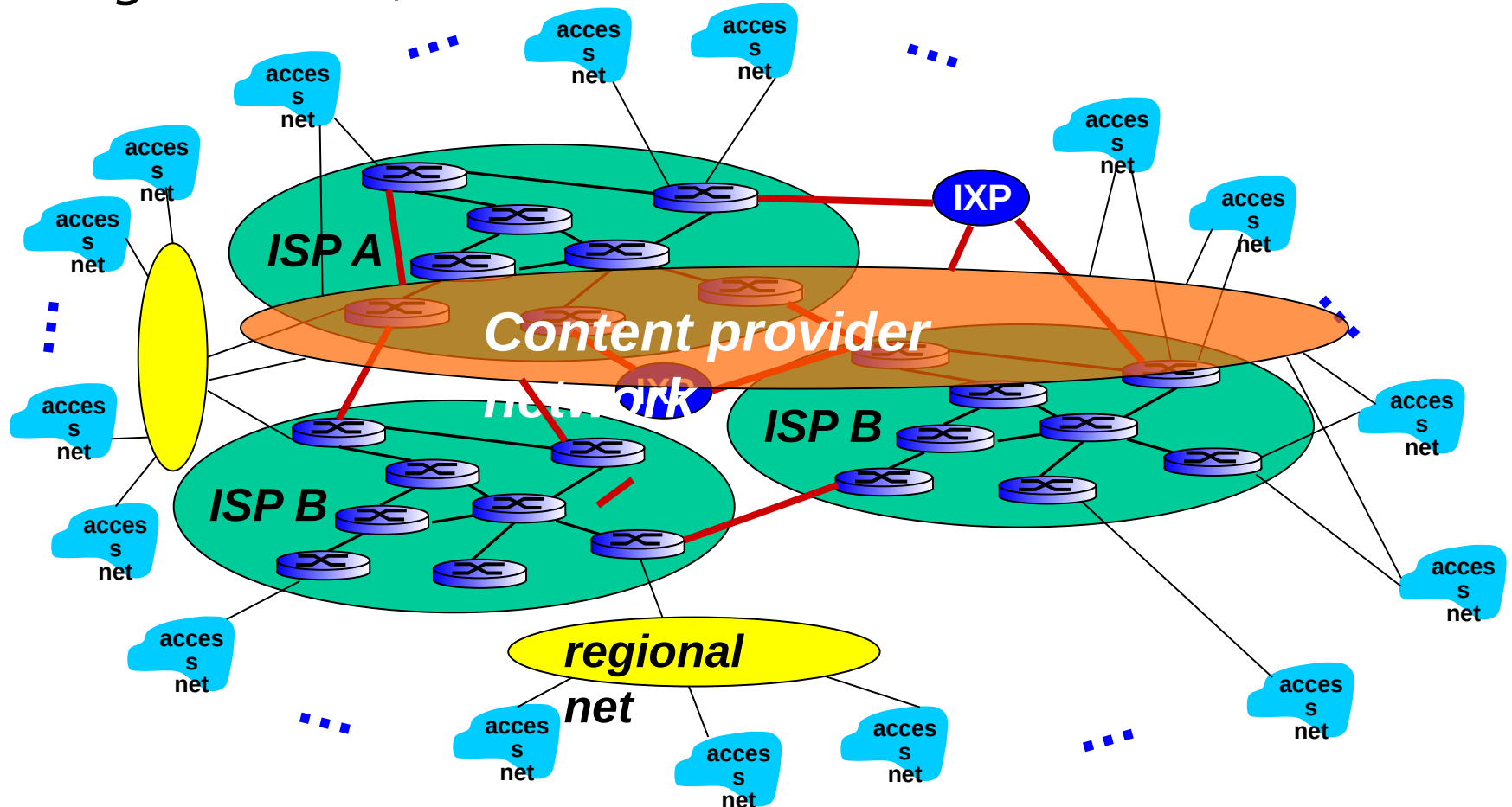
# Internet structure: network of networks

... and regional networks may arise to connect access nets to ISPS

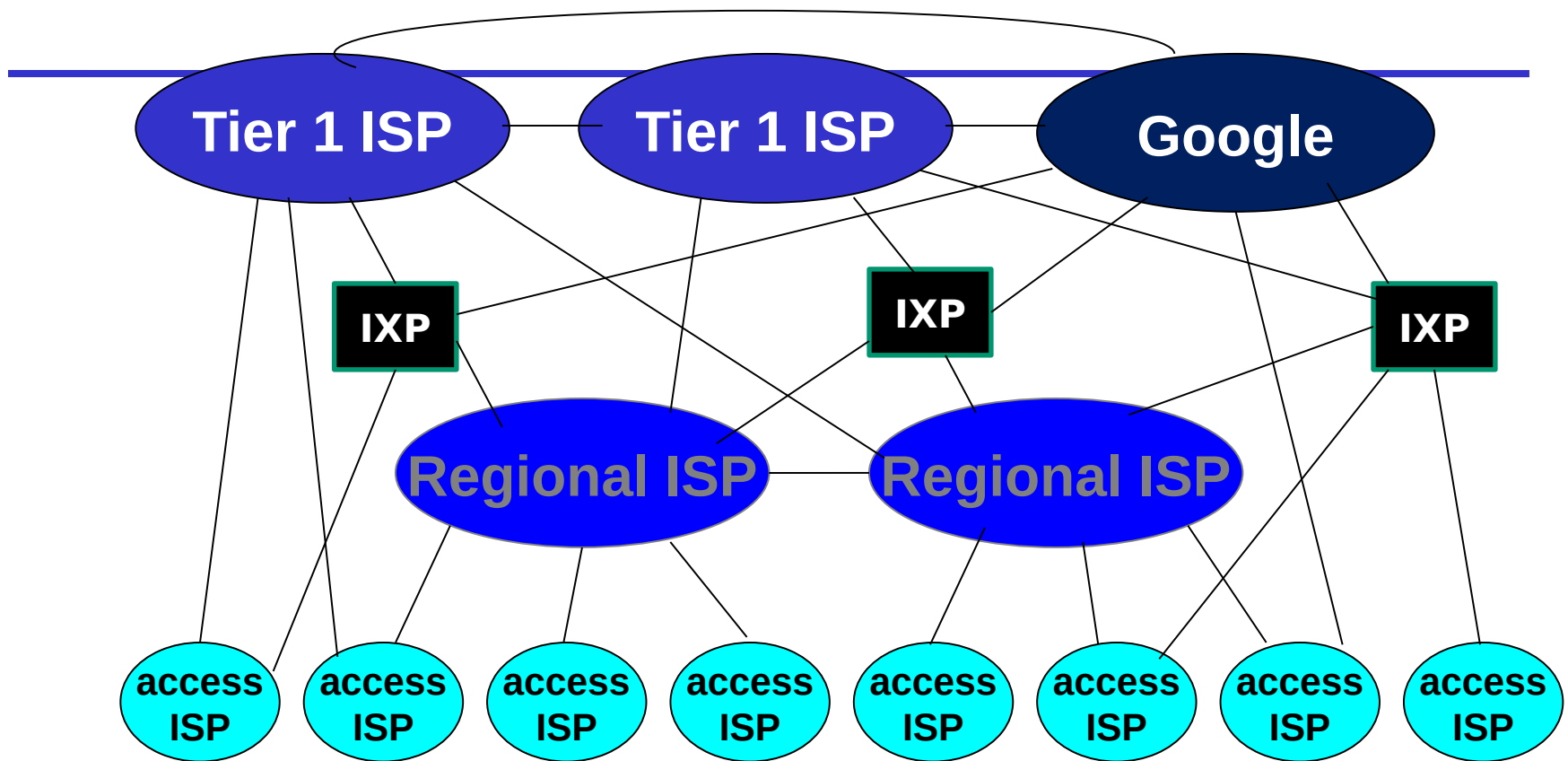


# Internet structure: network of networks

*... and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users*

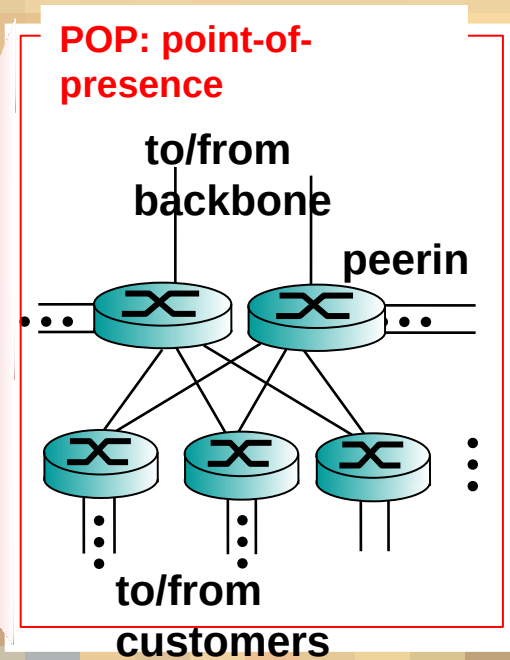


# Internet structure: network of networks



- at center: small # of well-connected large networks
  - “**tier-1**” **commercial ISPs** (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - **content provider network** (e.g, Google): private network that connects its data centers to Internet, often **bypassing tier-1**, regional ISPs

# Tier-1 ISP



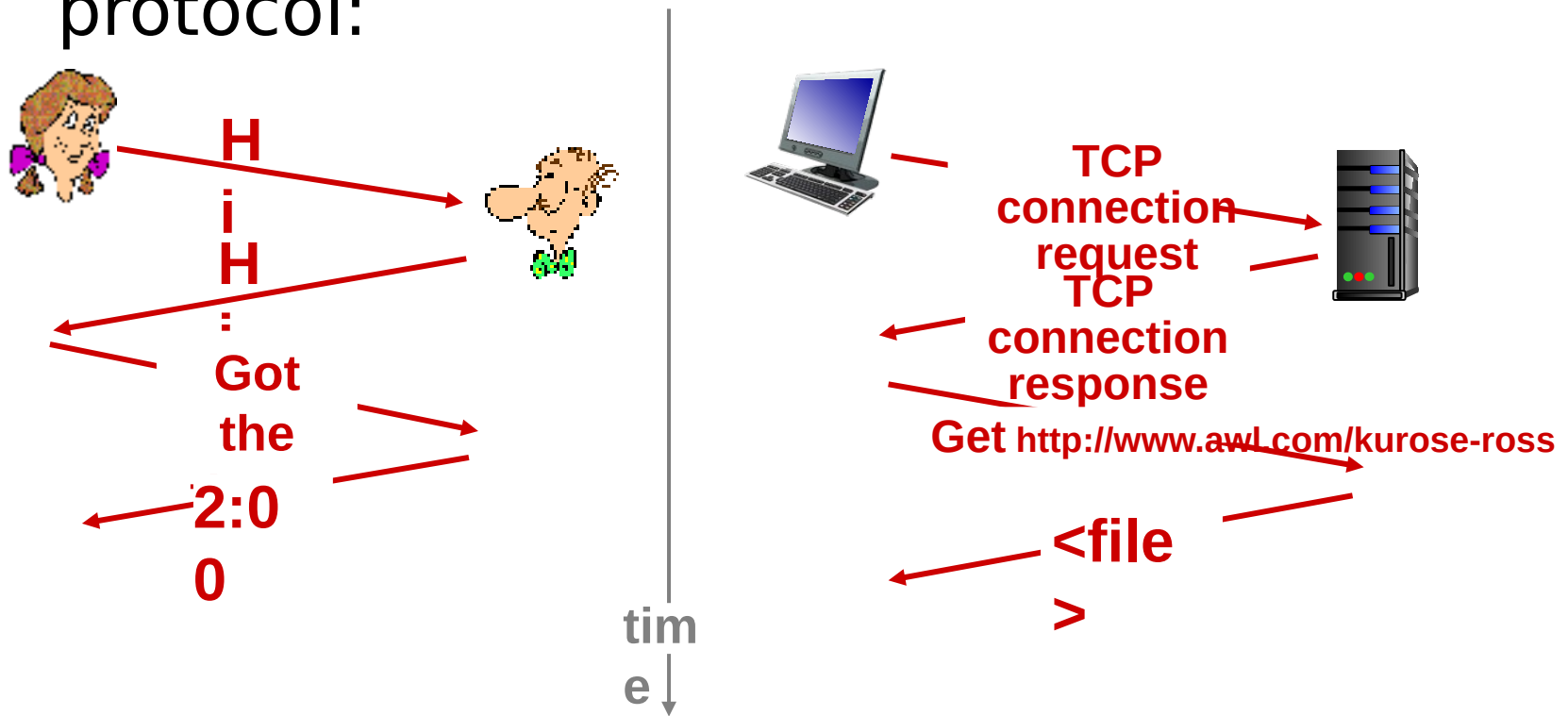
# 计算机网络的组成部分

---

- #1. 终端（硬件 + 软件）
- #2. 连接
- #3. 网络拓扑
- **#4. 协议**
- #5. 网络设备

# What's a protocol?

a human protocol and a computer network protocol:



**Q: other human protocols?**

# What's a protocol?

---

## *human protocols:*

- “what’s the time?”
- “I have a question”
- introductions

... specific **msgs sent**

... specific **actions taken** when msgs received, or other events

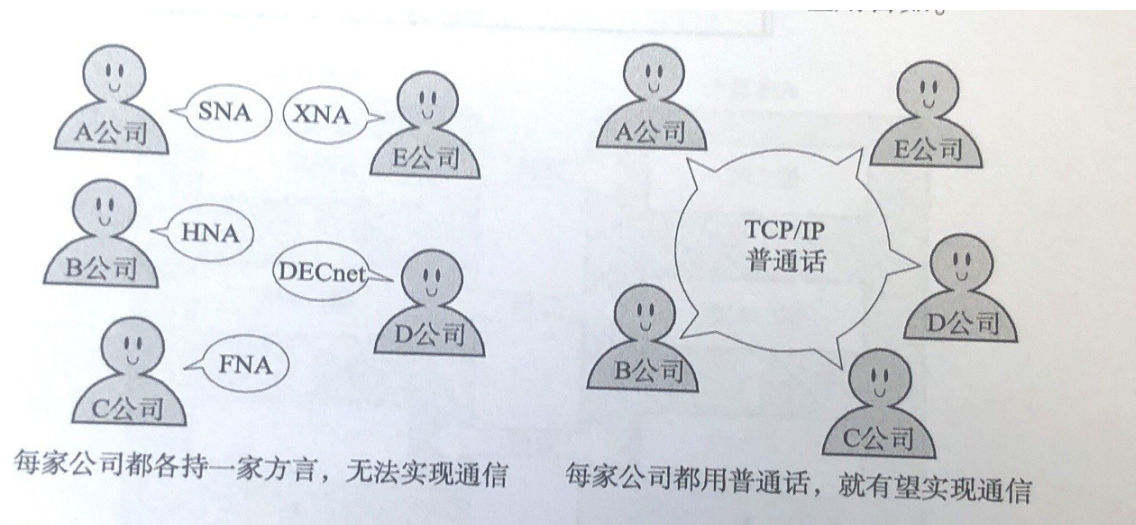
## *network protocols:*

- machines rather than humans
- all communication activity in Internet governed by protocols

***protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt***

# OSI 协议

- 1974 年，IBM 发布了 SNA 协议，用户 IBM 计算机之间的网络通信协议
- 其他公司也纷纷发布自己的协议
- 对用户来说不方便，不同机器之间无法沟通



# OSI 协议

- 国际标准化机构 ISO 制定了国际标准 OSI
- TCP/IP 是 IETF 建议和推广的一种协议（业界标准）

7 应用层	<p>&lt;应用层&gt; TELNET, SSH, HTTP, SMTP, POP, SSL/TLS, FTP, MIME, HTML, SNMP, MIB, SIP, RTP ...</p> <p>&lt;传输层&gt; TCP, UDP, UDP-Lite, SCTP, DCCP</p> <p>&lt;网络层&gt; ARP, IPv4, IPv6, ICMP, IPsec</p> <p>以太网、无线LAN、PPP…… (双绞线电缆、无线、光纤……)</p>
6 表示层	
5 会话层	
4 传输层	
3 网络层	
2 数据链路层	
1 物理层	

# OSI 协议

- 分为 7 层
  - 7. 应用层
  - 6. 表示层
  - 5. 会话层
  - 4. 传输层
  - 3. 网络层
  - 2. 数据链路层
  - 1. 物理层

分层名称	功 能	每层功能概览
7 应用层	针对特定应用的协议。	<p>针对每个应用的协议</p> <p>电子邮件 ↔ 电子邮件协议</p> <p>远程登录 ↔ 远程登录协议</p> <p>文件传输 ↔ 文件传输协议</p>
6 表示层	设备固有数据格式和网络标准数据格式的转换。	<p>接收不同表现形式的信息，如文字流、图像、声音等</p> <p>网络标准格式</p>
5 会话层	通信管理。负责建立和断开通信连接（数据流动的逻辑通路）。管理传输层以下的分层。	<p>何时建立连接，何时断开连接以及保持多久的连接？</p>
4 传输层	管理两个节点之间的数据传输。负责可靠传输（确保数据被可靠地传送到目标地址）。	<p>是否有数据丢失？</p>
3 网络层	地址管理与路由选择。	<p>经过哪个路由传递到目标地址？</p>
2 数据链路层	互连设备之间传送和识别数据帧。	<p>数据帧与比特流之间的转换</p> <p>分段转发</p>
1 物理层	以“0”、“1”代表电压的高低、灯光的闪灭。界定连接器和网线的规格。	<p>比特流与电子信号之间的切换</p> <p>连接器与网线的规格</p>

# What does an internet protocol do?

---

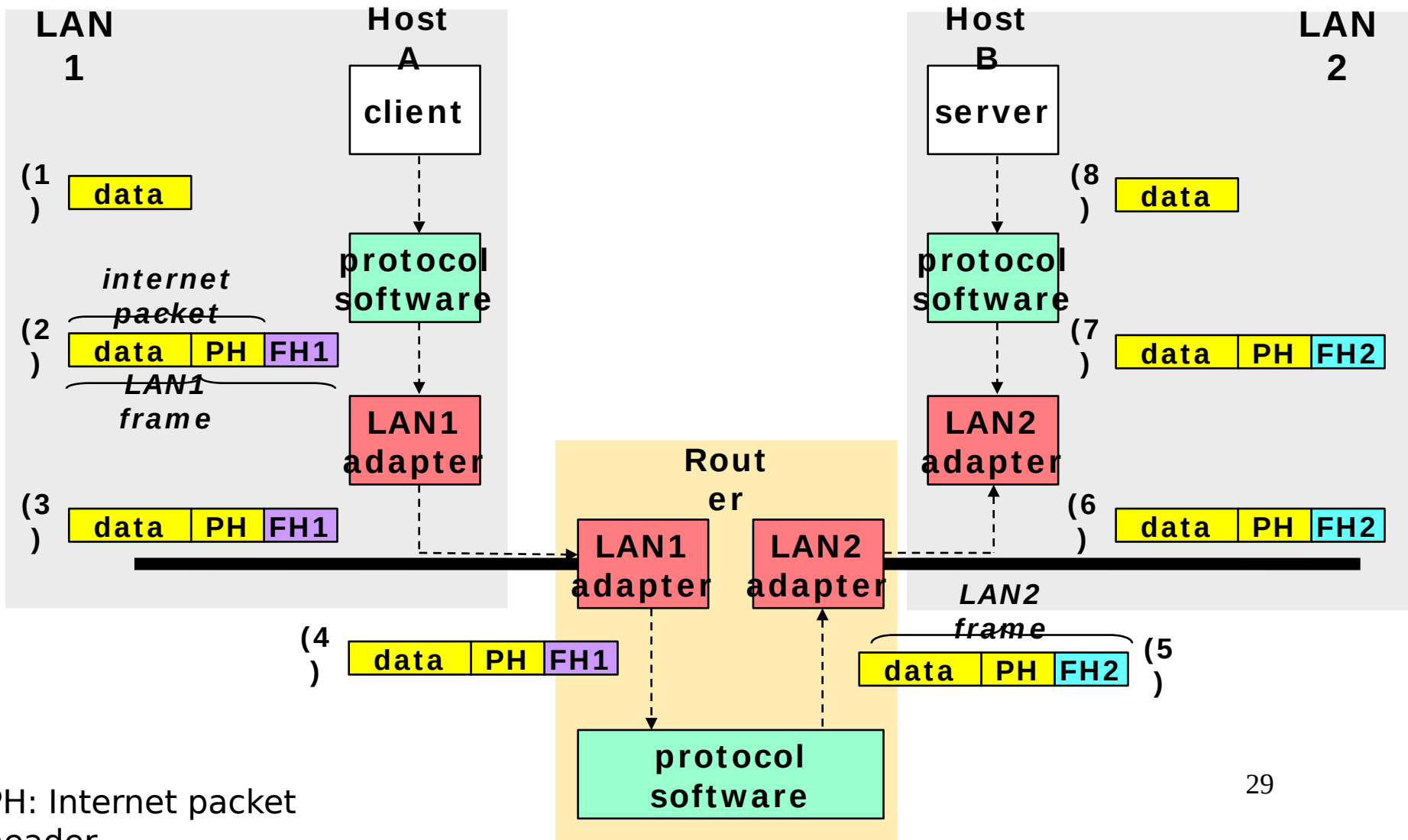
- Provides a **naming scheme**
  - The internet protocol defines a uniform format for **host addresses**
  - Each host (and router) is assigned at least one of these internet addresses that uniquely identifies it.

# What does an internet protocol do?

---

- Provide a **delivery mechanism**
  - The internet protocol defines a standard transfer unit (**packet**)
  - Packet consists of **header** and **payload**
    - header: contains info such as packet size, source and destination addresses.
    - payload: contains data bits sent from source host.

# Transferring data over an internet



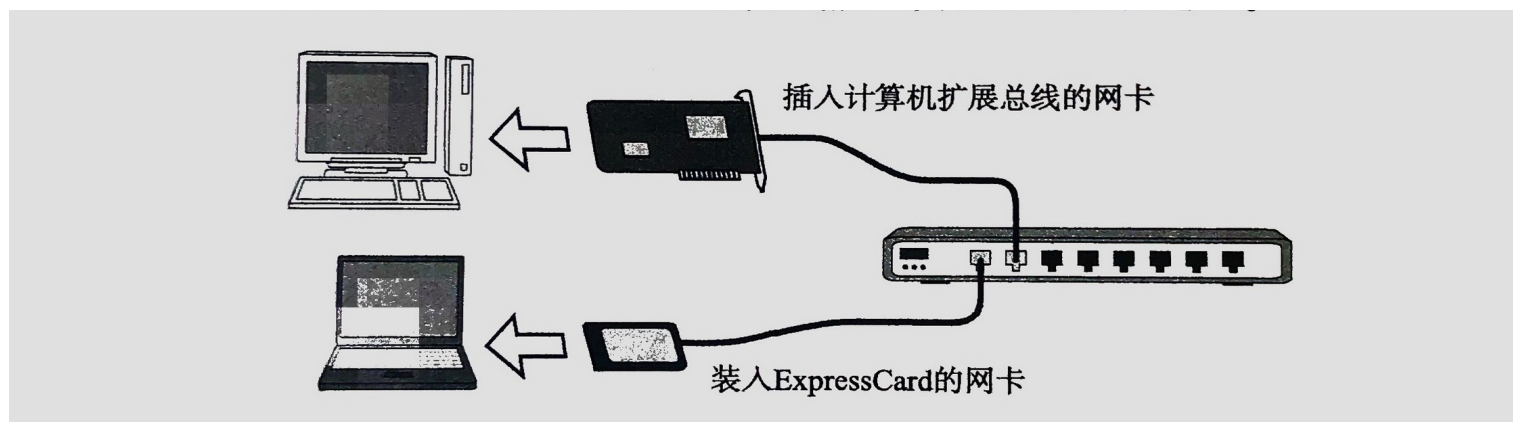
# 计算机网络的组成部分

---

- #1. 终端（硬件 + 软件）
- #2. 连接
- #3. 网络拓扑
- #4. 协议
- **#5. 网络设备**

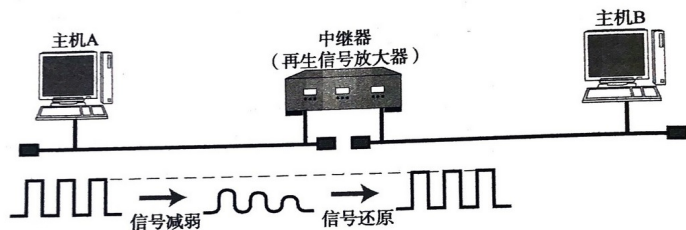
# 网络设备

- 网卡
  - 网络接口卡 NIC/ 网络适配器
    - 以太网卡
    - 光纤网卡
    - Infiniband 网卡

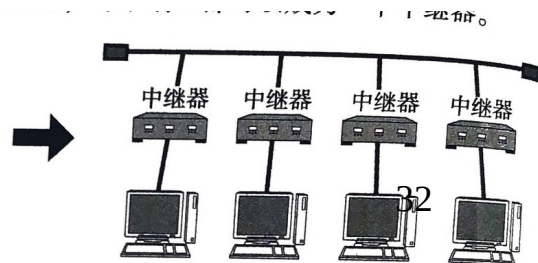
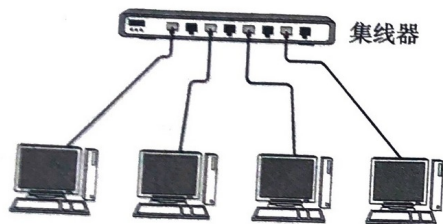


# 中继器

- 中继器（Repeater）：工作在物理层
  - 在物理层面延长网络的设备
  - 将收到的信号放大后再继续传输
  - 或者两端连接不同媒介，完成转换功能
  - 集线器（Hub）的每个端口都有中继器的功能

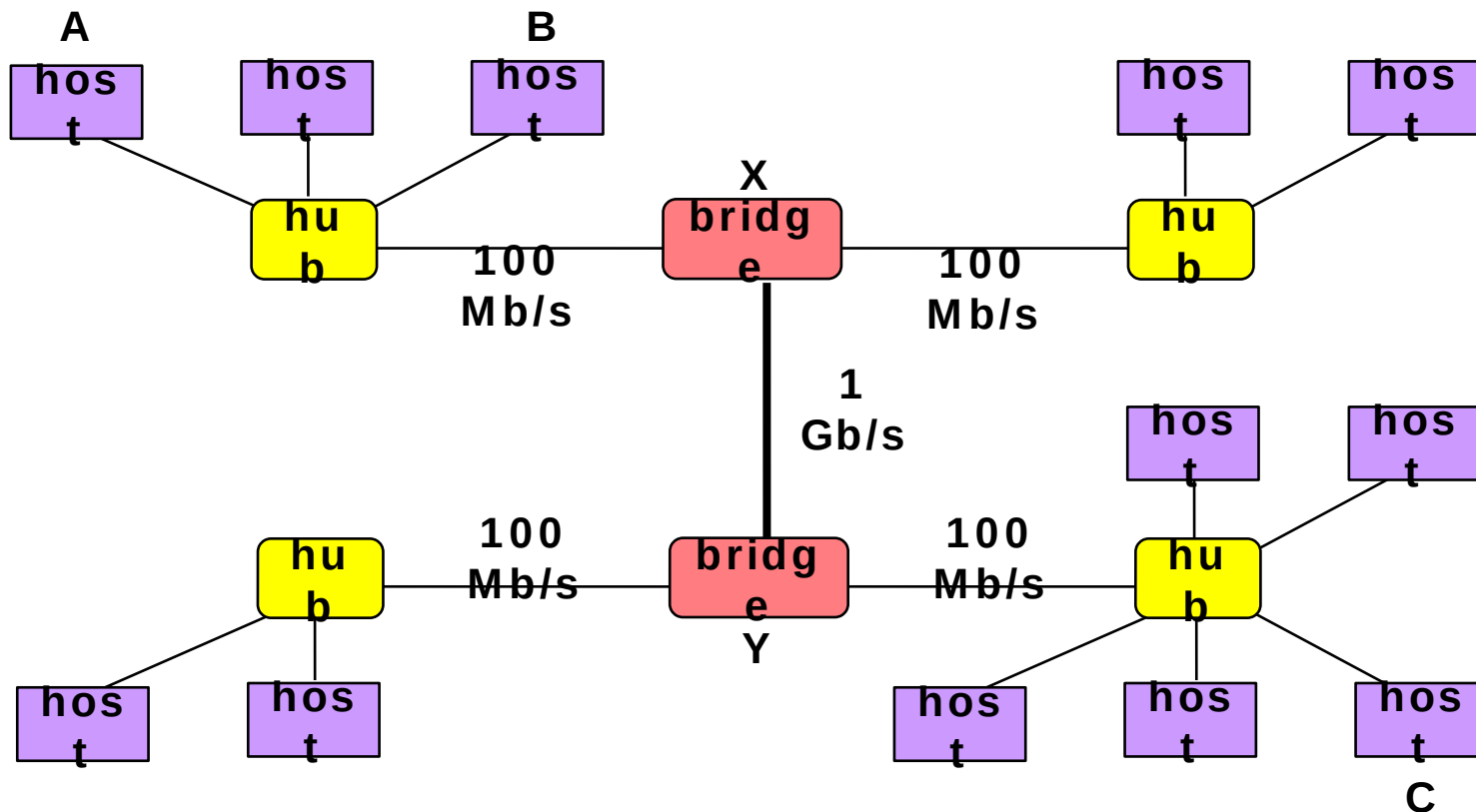


- 中继器是对减弱的信号进行放大和发送的设备。
- 中继器通过物理层的连接延长网络。
- 即使在数据链路层出现某些错误，中继器仍然转发数据。
- 中继器无法改变传输速度。



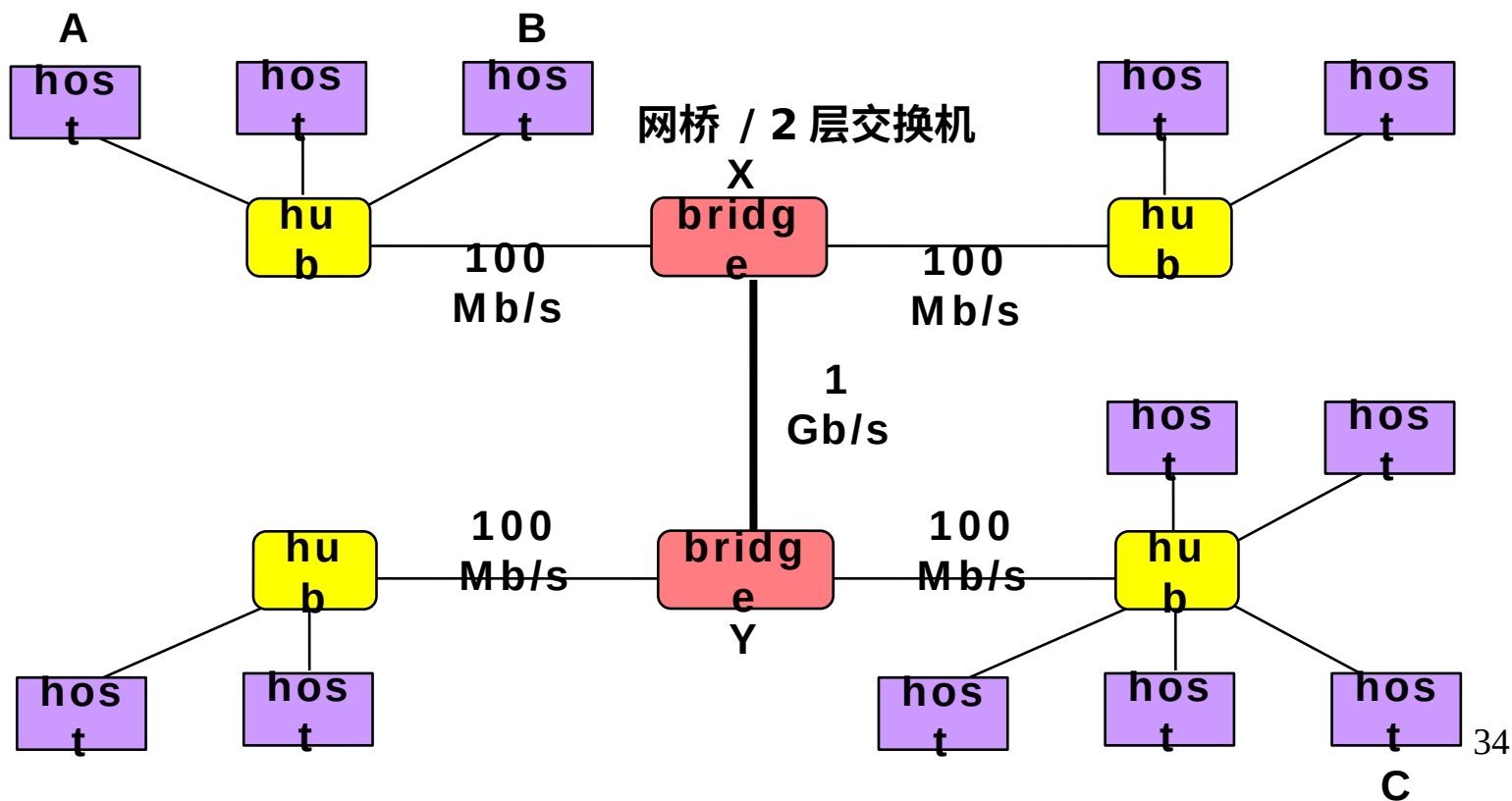
# 网桥 / 2 层交换机

- 网桥：连接两个网络
  - 工作在数据链路层（第 2 层）
  - 两端一般是相同类型的网络（但可以速率不同）



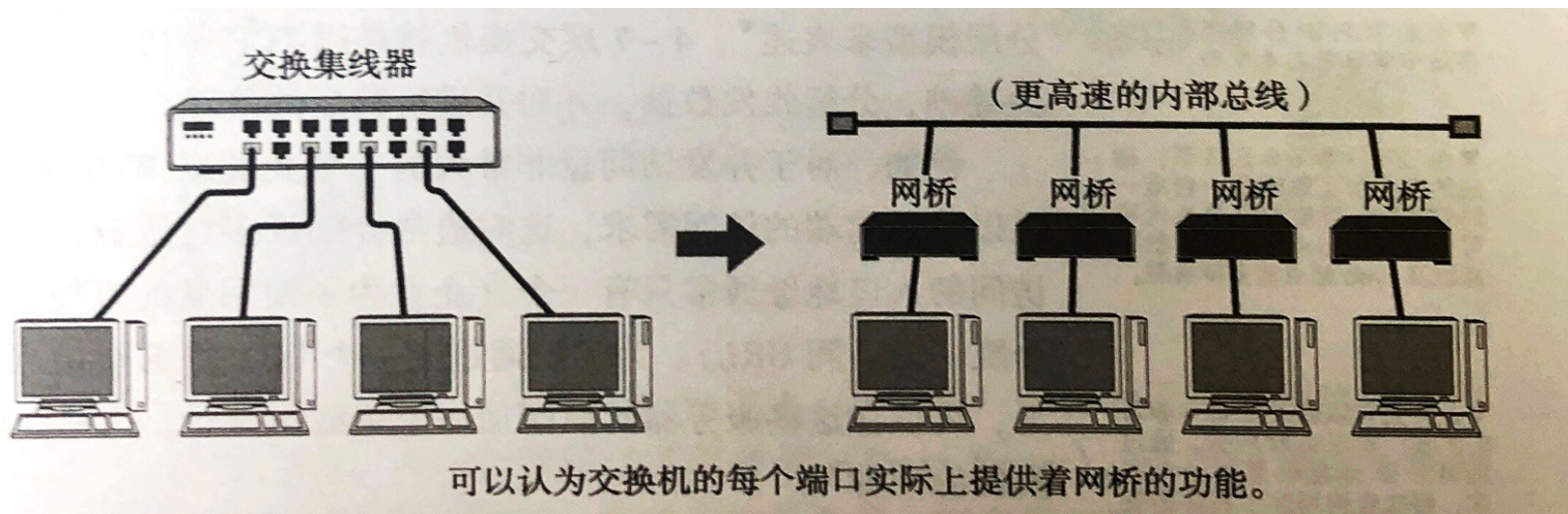
# 网桥 / 2 层交换机

- 自学式网桥：逐渐记住哪个主机可以通过哪个端口可达
  - A->B: 在网桥 X 丢弃; A->C: X->Y



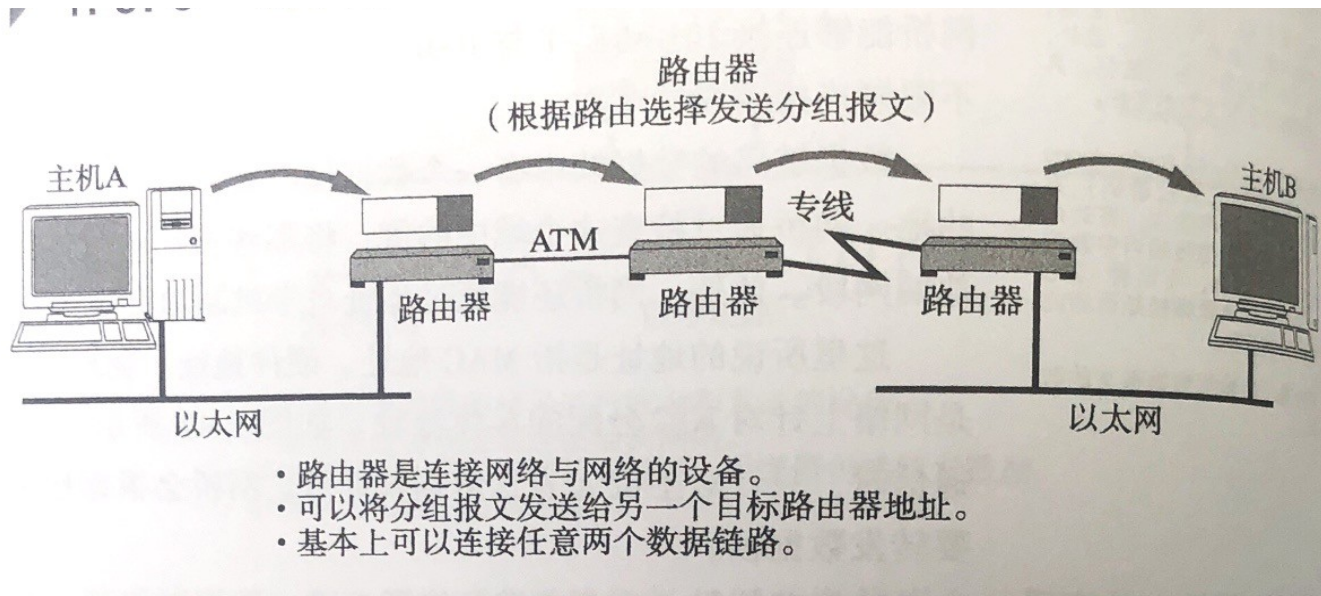
# 网桥 / 2 层交换机

- 常用的交换集线器（Hub），现在基本上也属于网桥的一种
  - 每个端口提供类似网桥的功能

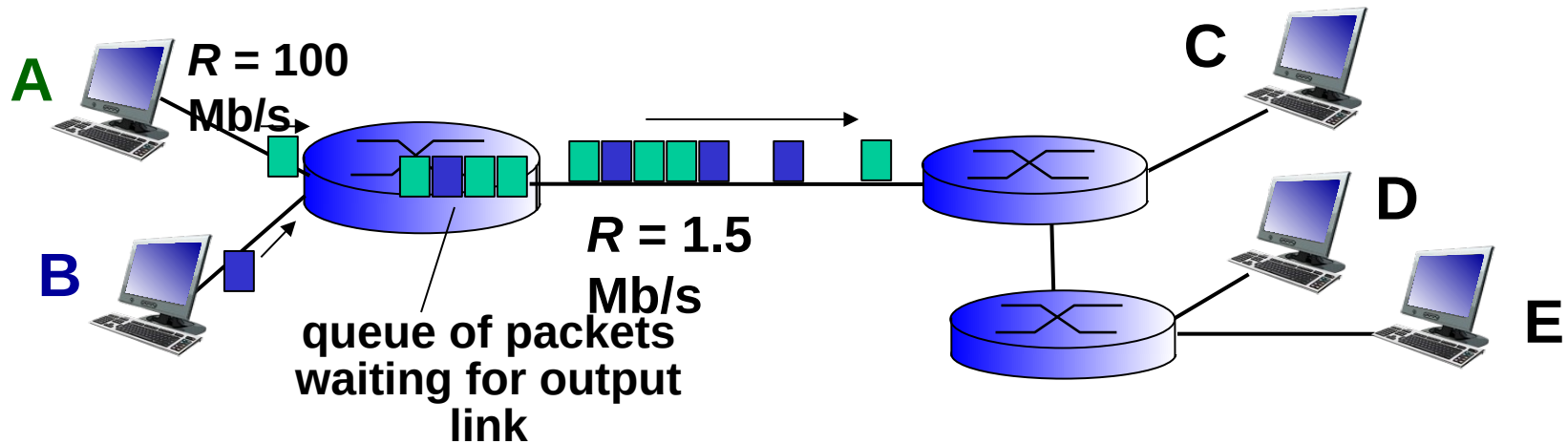


# 路由器 / 3 层交换机

- 路由器（Router）工作在 IP 层
  - 连接网络与网络（互联网的基石）
  - 主要功能是根据路由规则转发
  - 特殊计算机，高端 router 内存全是 SRAM



# 路由器 / 3 层交换机



## queuing and loss:

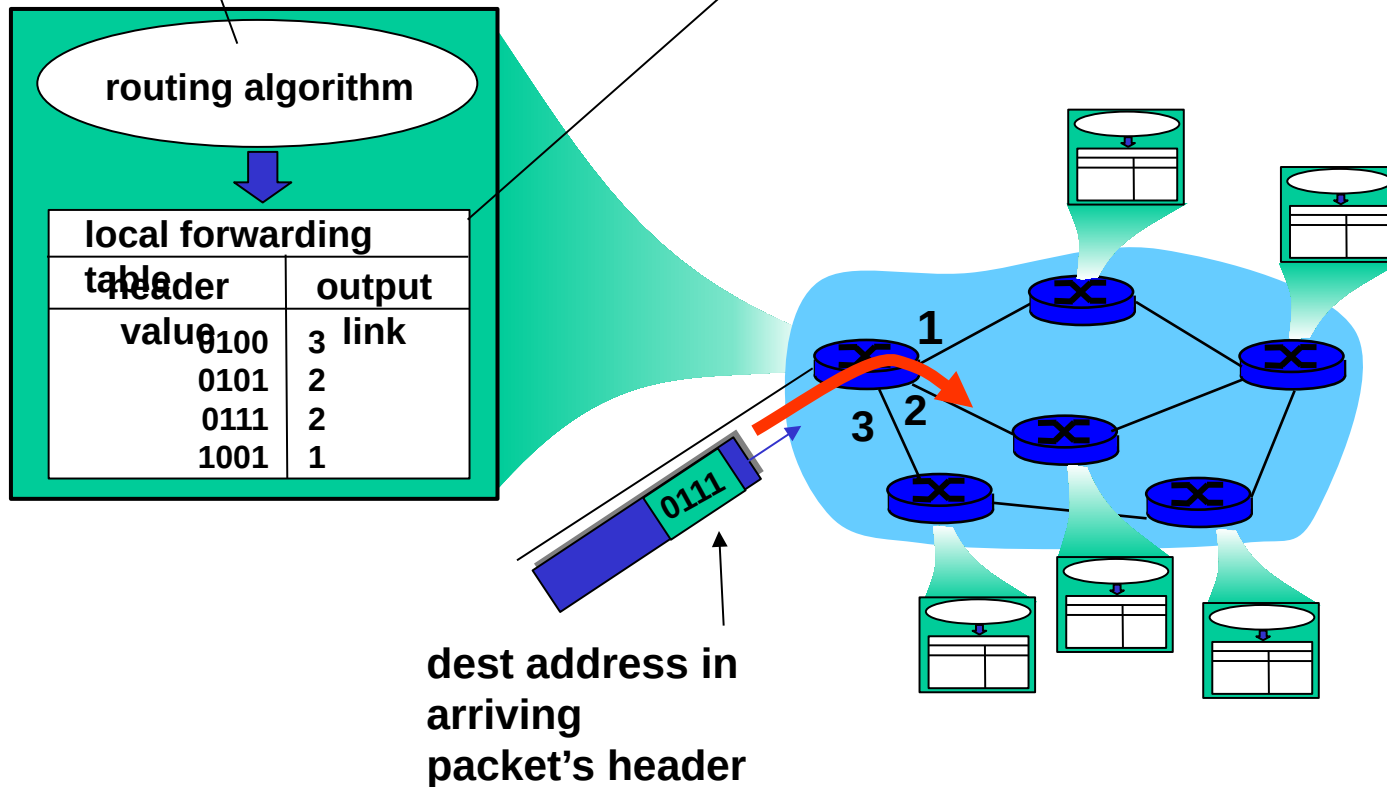
- ❖ If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

# Two key network-core functions

***routing***: determines source-destination route taken by packets

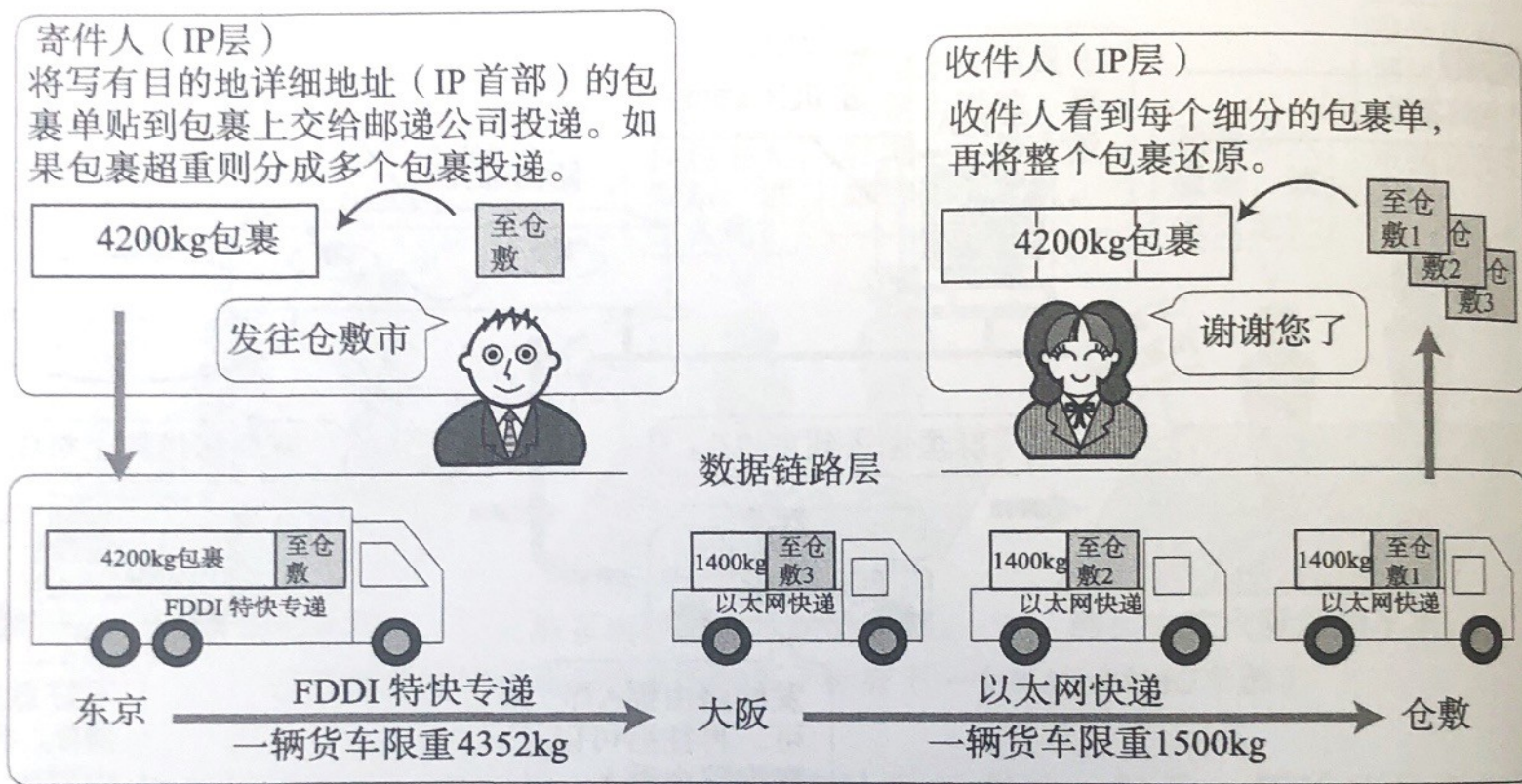
- ***routing algorithms***

***forwarding***: move packets from router's input to appropriate router output



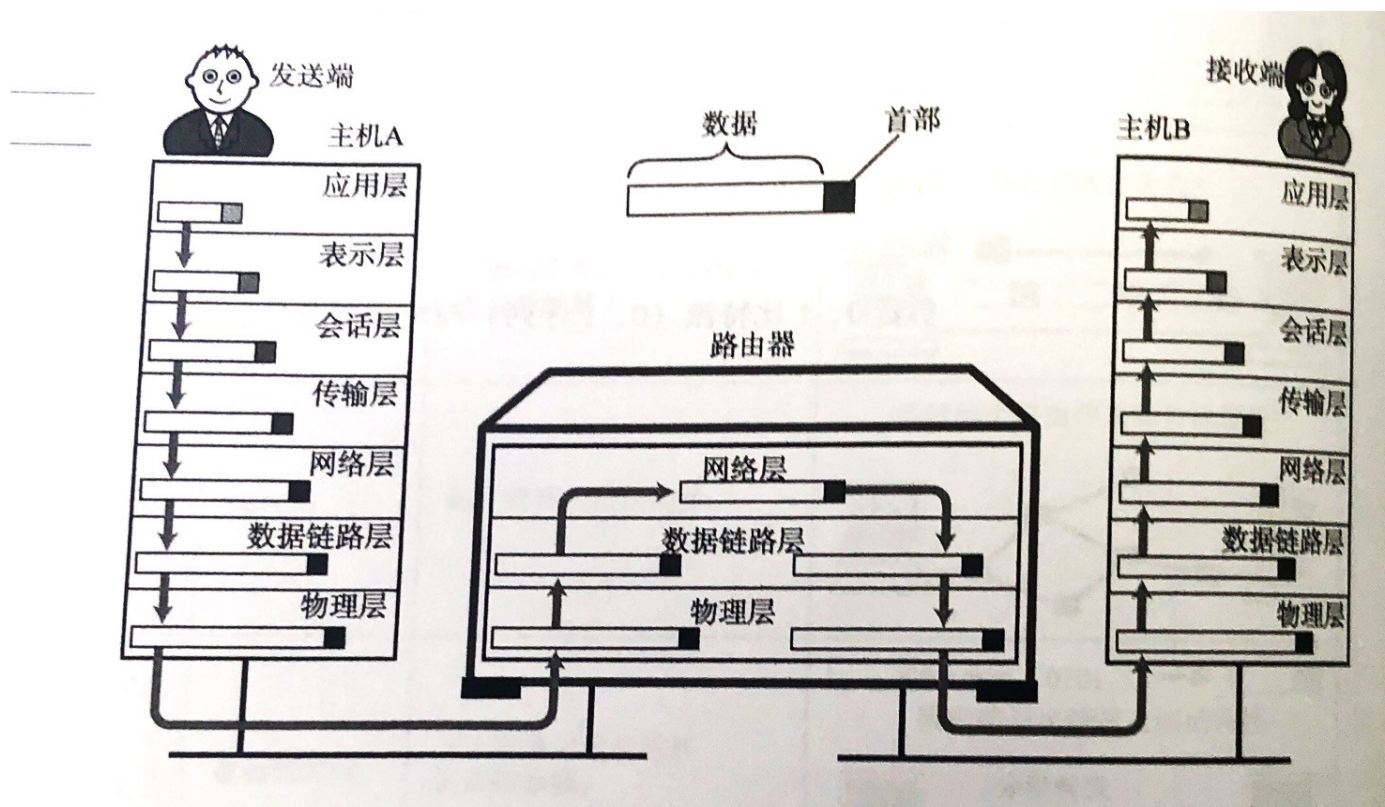
# 路由器 / 3 层交换机

- IP 层和数据链路层的区别



# 路由器 / 3 层交换机

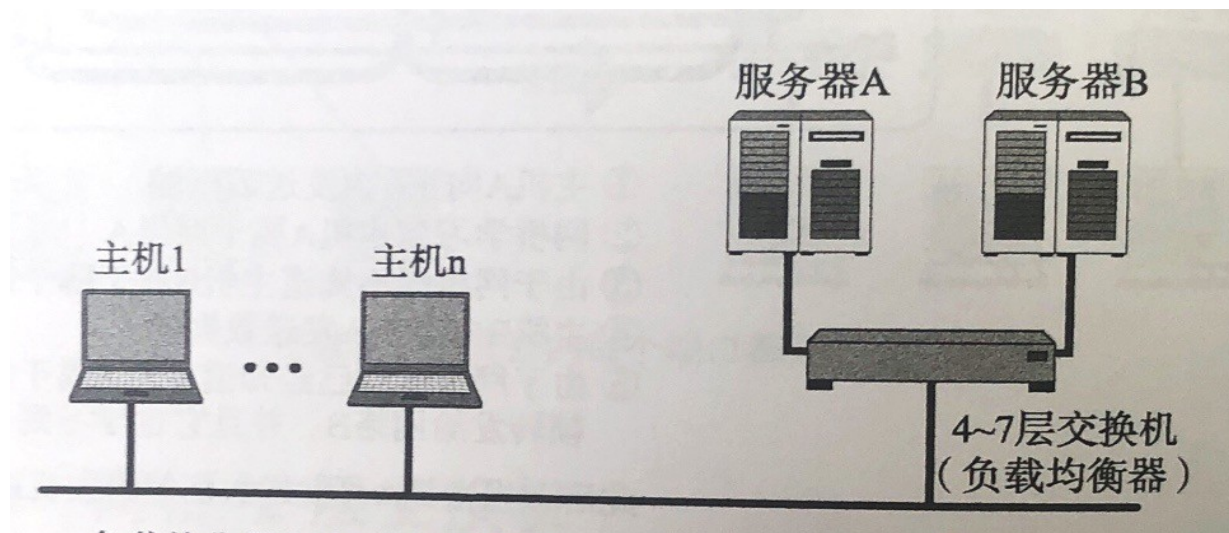
- 路由器与 OSI 7 层模型的关系



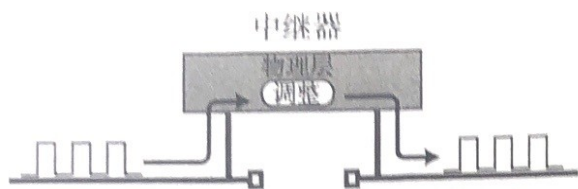
## 4 ~ 7 层交换机

---

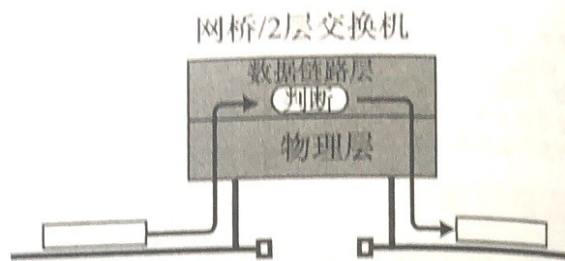
- 应用决定
- 例如针对 Web 服务器的负载均衡器



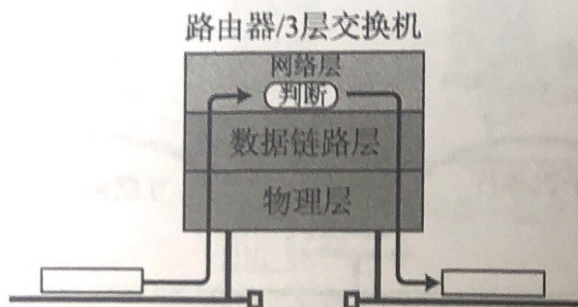
# 总结



识别0、1序列调整波形进行相应放大与转发。  
可以在双绞线电缆与光纤电缆之间转换。

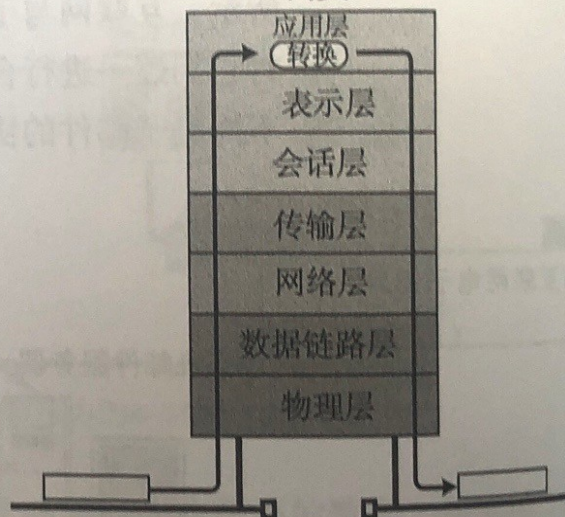


识别数据链路层中的数据帧，重构数据帧转发。  
丢弃错误的帧。



判断网络路径，转发至目标地址。

4~7层交换机  
网关



负责传输层以上的数据转发及其协议的转换。

# Outline

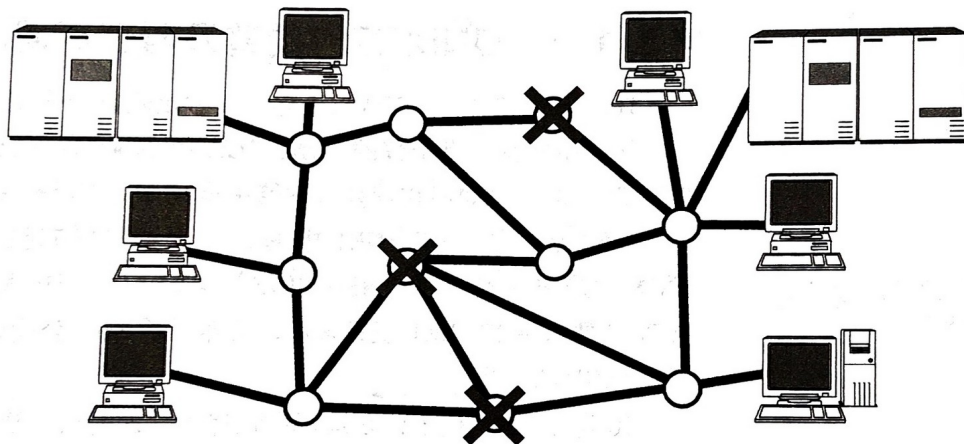
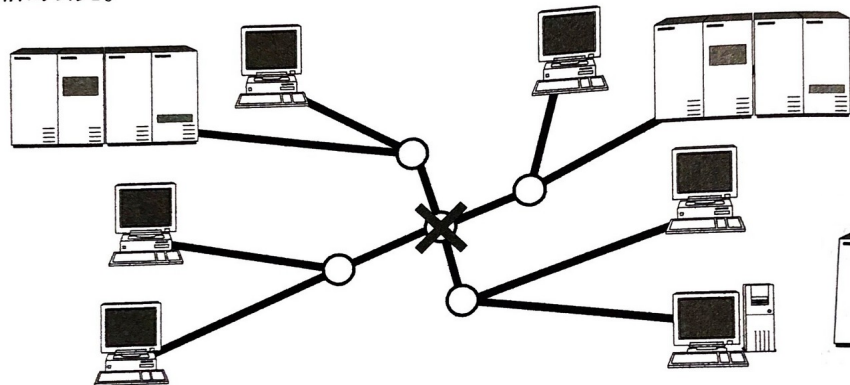
---

- 计算机网络的组成部分
- **TCP/IP 协议栈**

# TCP/IP 协议出现的背景

- 20 世纪 60 年代，美国国防部（DoD）组织网络通信技术的研究
- 容灾性较强的网络，个别节点被攻击，网络仍然可用

4188981260



# TCP/IP 协议出现的背景

---

- ARPANET
  - 1969 年，军方的实验性网络
  - 连接了美国西海岸大学和研究所等 4 个节点
  - 后来发展迅速，形成了巨大规模的网络
    - 3 年内从 4 个节点发展到 34 个节点
  - 是 Internet 的鼻祖

# TCP/IP 协议出现的背景

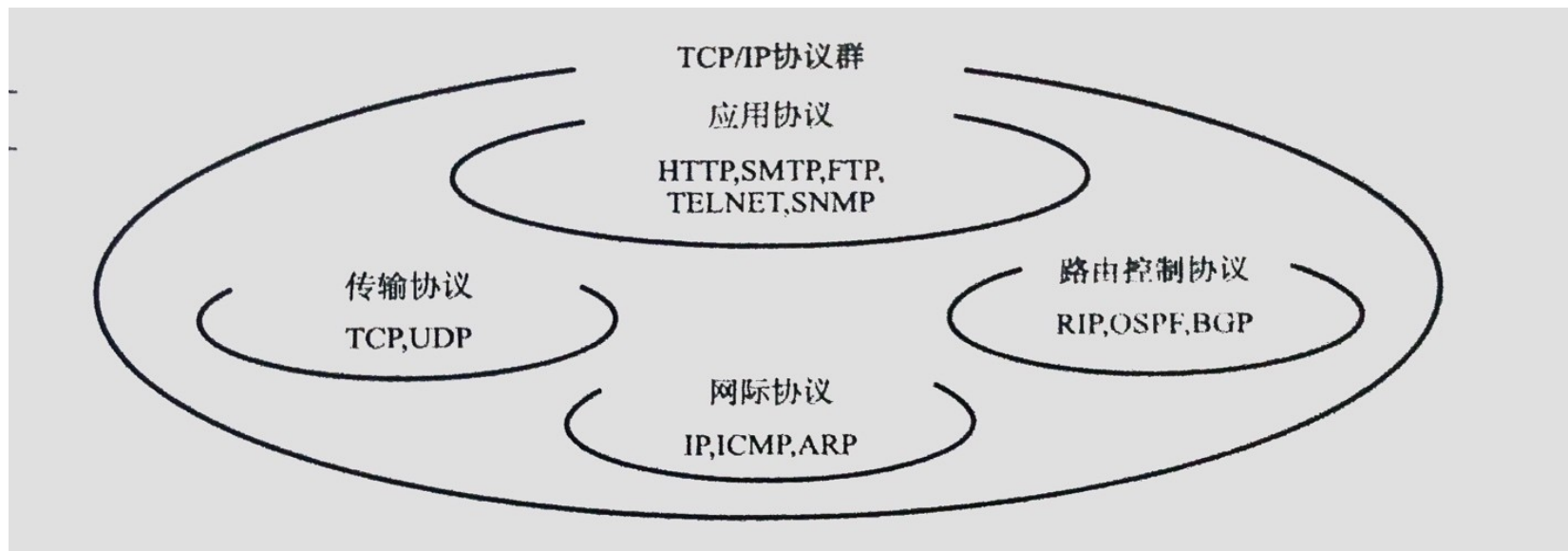
---

- TCP/IP 的诞生
  - 20 世纪 70 年代初，ARPANET 中的一个研究机构研发出了 TCP/Ip 协议
  - 1982 年，TCP/Ip 最终确定下，成为 ARPANET 网络唯一指定的协议
  - UNIX 是最早实现 TCP/IP 的操作系统
  - 随着 UNIX 的广泛应用，TCP/Ip 成为业界标准
  - 1996 年，IPv6 规范出炉

# TCP/Ip 协议族

---

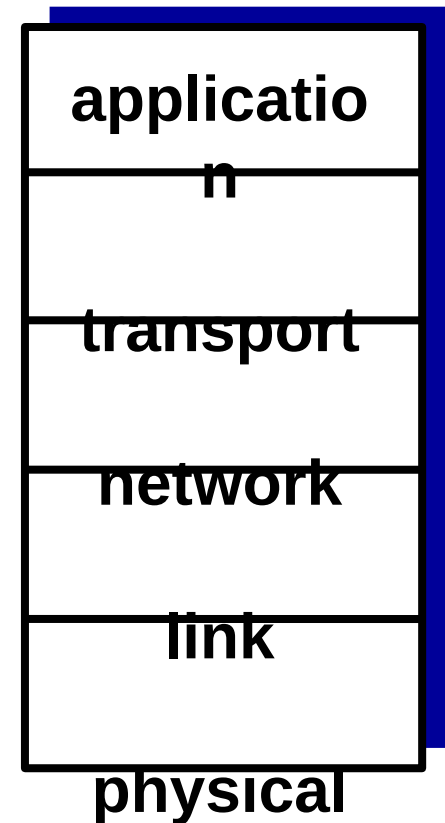
- 不止 TCP 和 Ip 两种协议，但这两种协议非常重要



# 互联网协议栈 (Internet protocol stack)

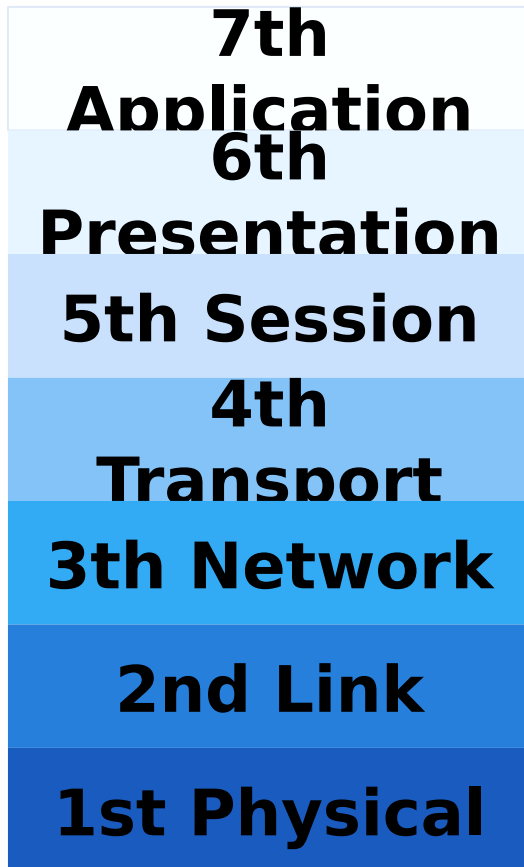
---

- *application*: supporting network applications
  - FTP, SMTP, HTTP
- *transport*: process-process data transfer
  - TCP, UDP
- *network*: routing of datagrams from source to destination
  - IP, routing protocols
- *link*: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical*: bits “on the wire”

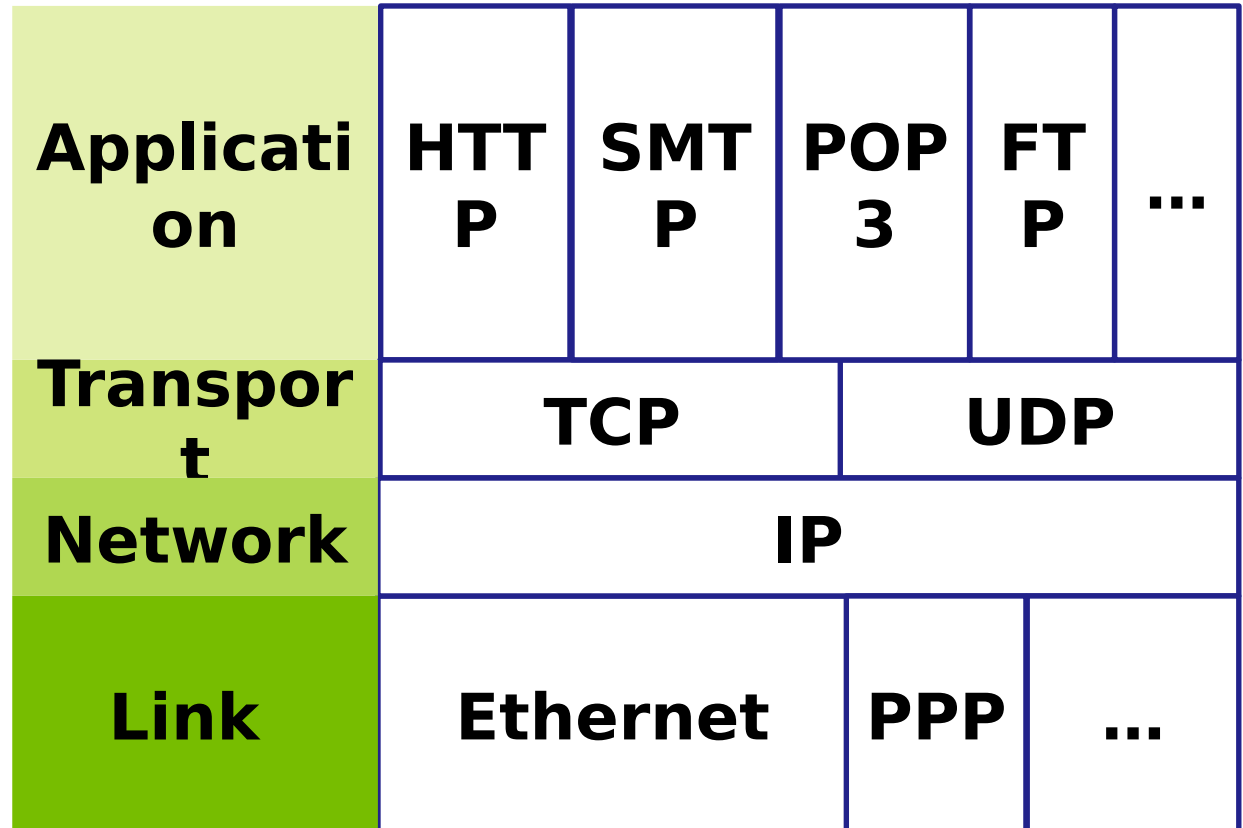


# OSI, TCP/IP & Protocol Stack

---

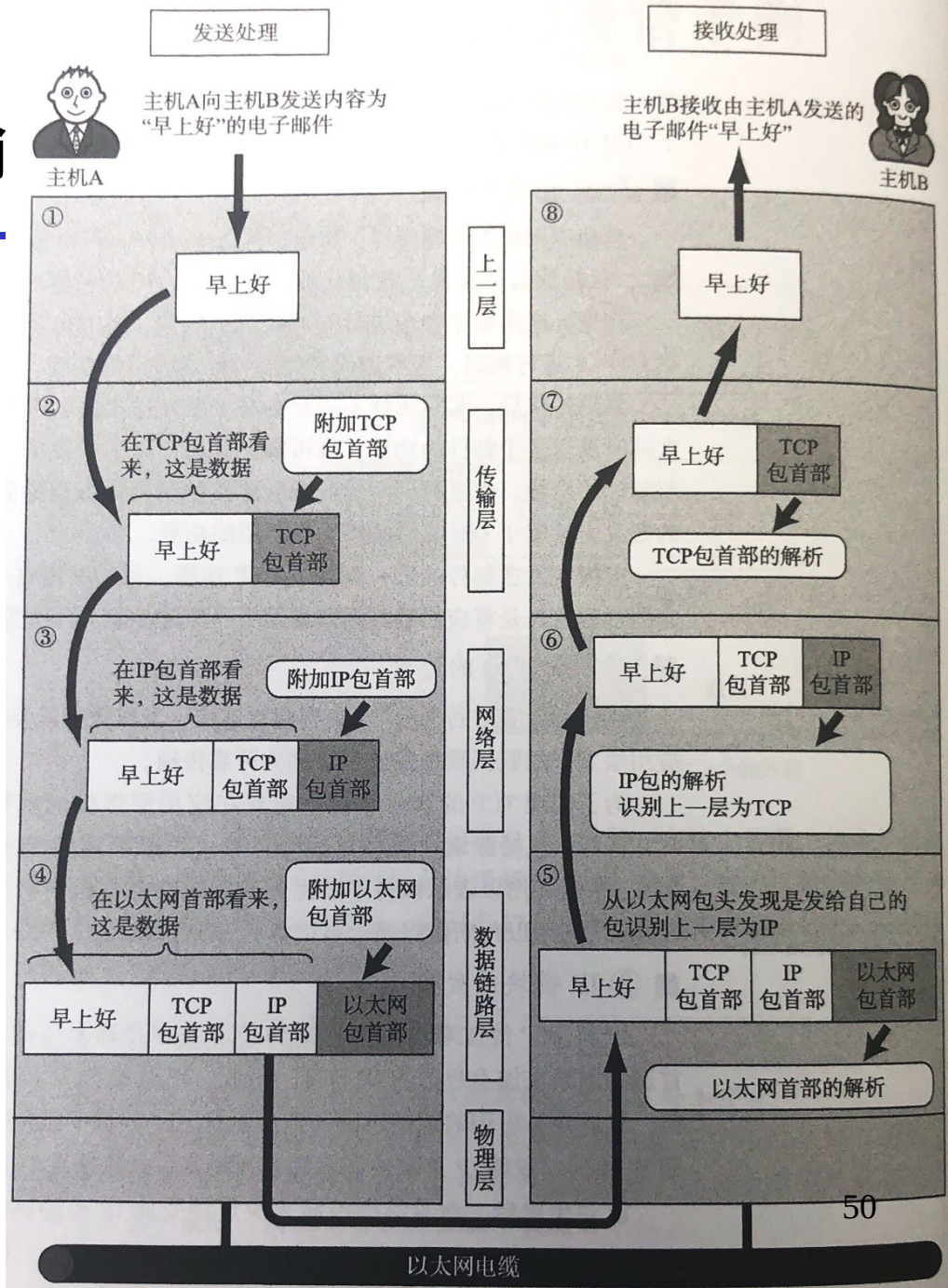


**OSI**



**TCP/IP**

# TCP/IP 数据包传输



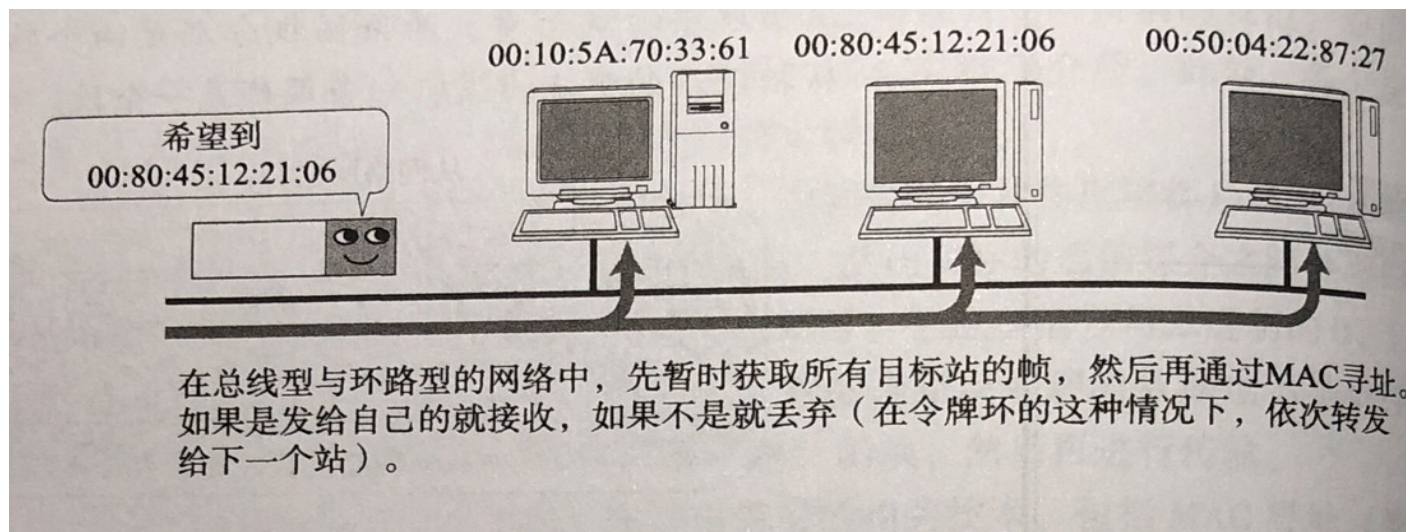
# 数据链路层

---

- 定义了通过通信媒介互连的设备之间的传输规范
- 通信媒介包括
  - 双绞线、同轴电缆、光纤、电波、红外等
- 设备之间可能有交换机、网桥、中继器等

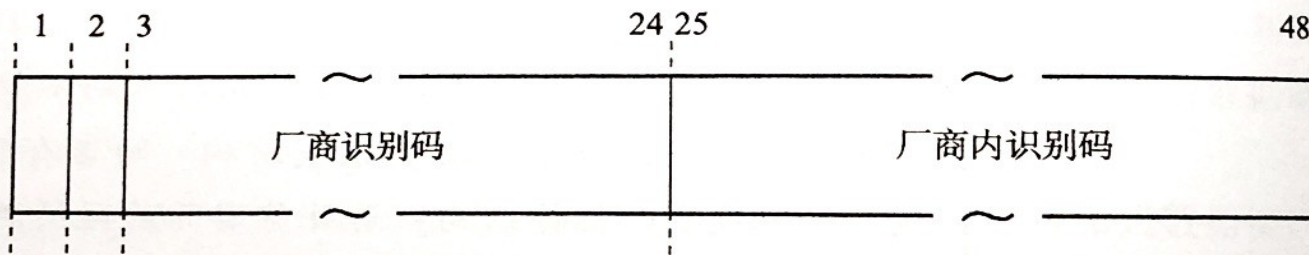
# 数据链路层

- MAC 地址
  - 用于识别数据链路中互连的节点



# 数据链路层

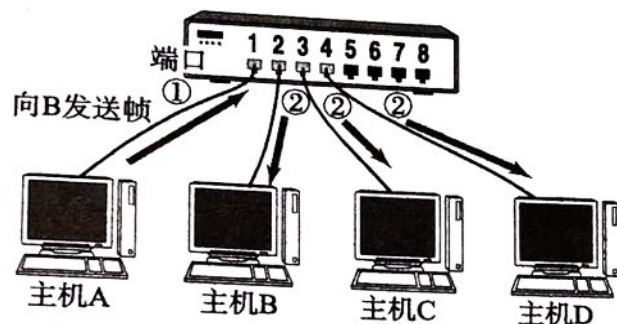
- MAC 地址
  - 48-bit ，一般烧入网卡的 ROM 中
  - 全球唯一，不重复
  - 但虚拟机可以指定，不一定全球唯一（只要不属于同一个数据链路就不会出问题）



第1位：单播地址（0）/多播地址（1）  
第2位：全局地址（0）/本地地址（1）  
第3~24位：由IEEE管理并保证各厂家之间不重复  
第25~48位：由厂商管理并保证产品之间不重复

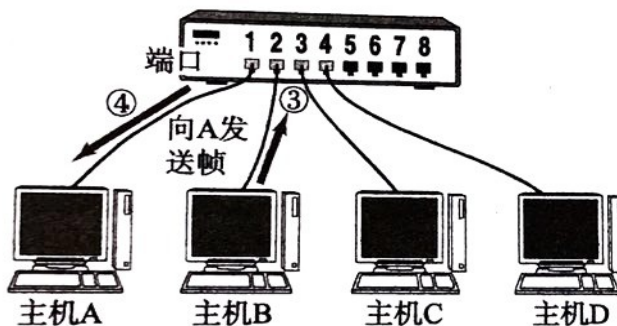
# 数据链路层

- 交换机工作在数据链路层，能够自学
  - MAC 地址没有层次性，网络设备阅读，转发表也会变大，检索转发表需要的时间变长，需要切分网络



① 从源MAC地址可以获知主机A与端口1相连接。

② 拷贝那些以“未知”MAC地址为目标的帧给所有的端口。



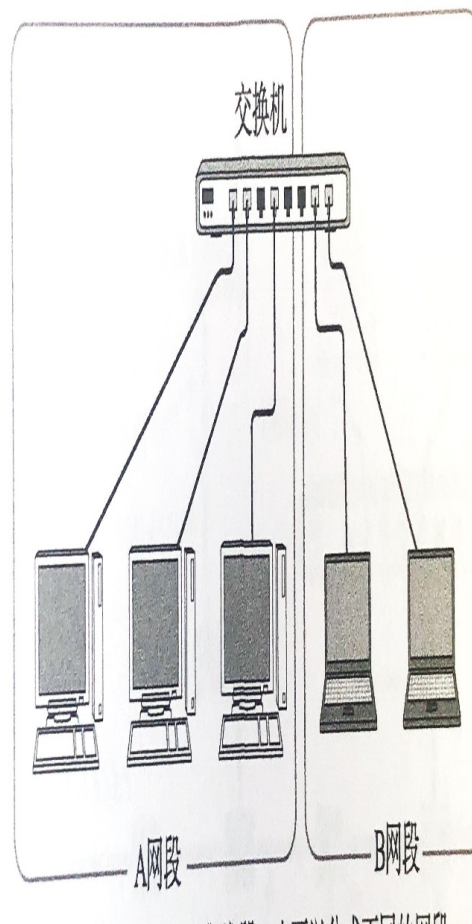
③ 从源MAC地址可以获知主机B与端口2相连接。

④ 由于已经知道主机A与端口1相连接，那么发给主机A的帧只拷贝给端口1。

以后，主机A与主机B的通信就只在它们各自所连接的端口之间进行。

# 数据链路层

- VLAN :
  - 在交换机上改变网络拓扑
  - 不用真的改变硬件线路



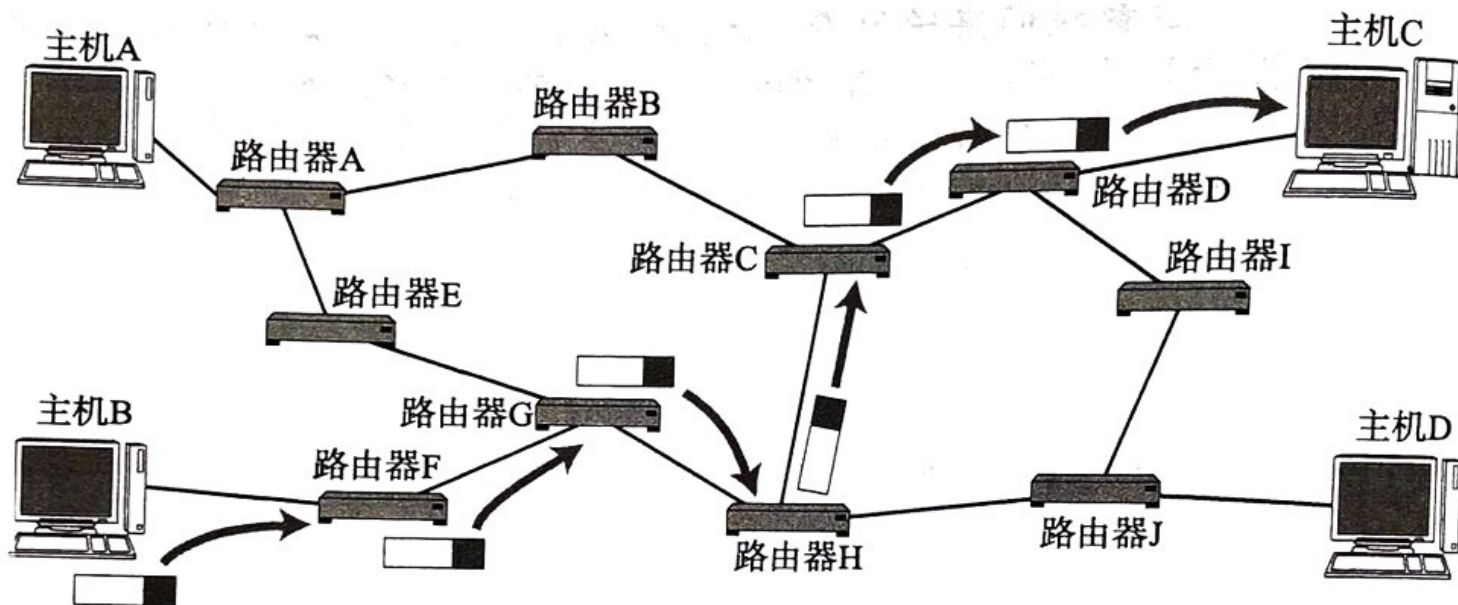
# IP 层

---

- IP 层简介
- IP 地址 ( IPv4, IPv6 )
- DHCP
- IP 路由
- DNS
- NAT

# IP 层

- IP 层的作用是在复杂的网络环境中，将数据包发给最终的目标地址
- IP 层是面向无连接型



# IP 报文 (datagram) 格式

IP protocol version number

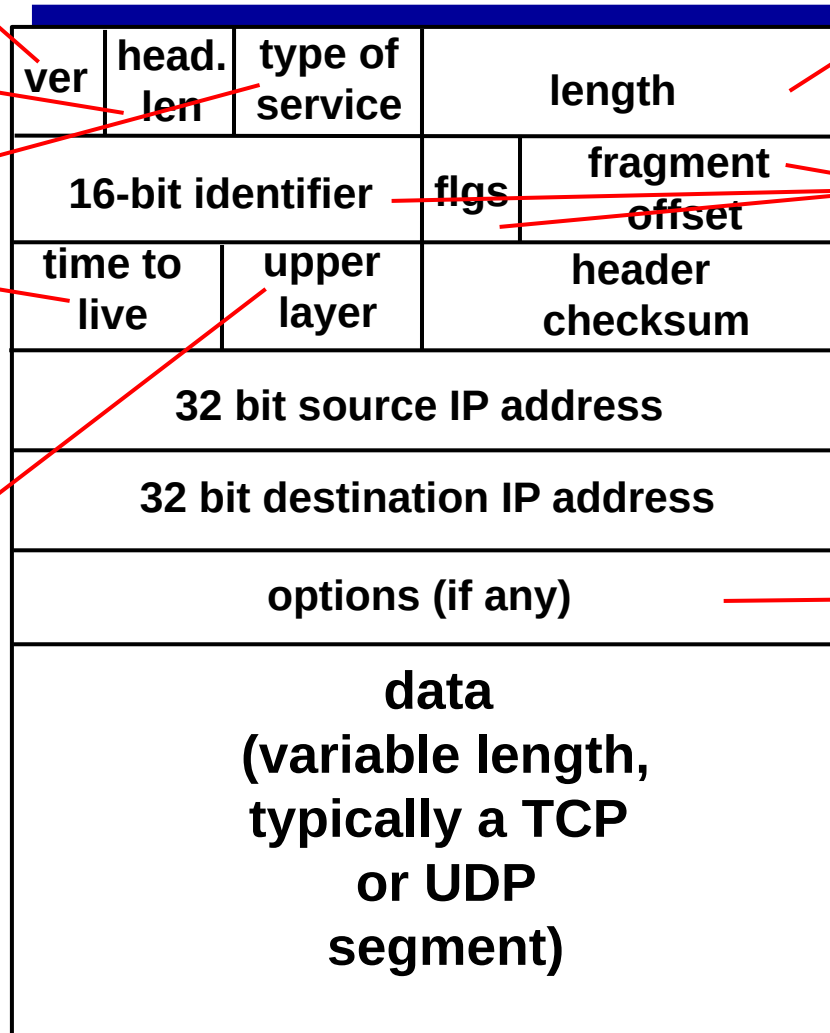
header length (bytes)

“type” of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

32 bits



total datagram length (bytes)

for fragmentation/ reassembly

e.g. timestamp, record route taken, specify list of routers to visit.

## how much overhead?

- ❖ 20 bytes of TCP
- ❖ 20 bytes of IP
- ❖ = 40 bytes + app layer overhead

# Programmer's view of the Internet

---

- 网络上的主机 (host) 地址都表示为 32-bit 的 **IP 地址**
  - `0xca7828bc`
- IP 地址可以进一步被映射到互联网域名 (Internet domain names)
  - `0xca7828bc` is mapped to `xxx.yyy.edu.cn`
- 一个主机上的进程通过连接 (connection) 与另一个主机上的进程通信

# 点分十进制表示 (Dotted decimal notation)

---

- 通常，32-bit IP 地址中的每个字节被表示为一个十进制数，字节之间通过点符号分隔
  - IP address **0xca7828bc** = **202.120.40.188**

# 课堂练习

---

- IP 地址换算
  - 0x7f000001
  - 202.112.113.25
  - 14.152.72.131

# 课堂答案

---

- IP 地址换算

- 0x7f000001

127.0.0.1

- 202.112.113.25

0xca707119

- 14.152.72.131

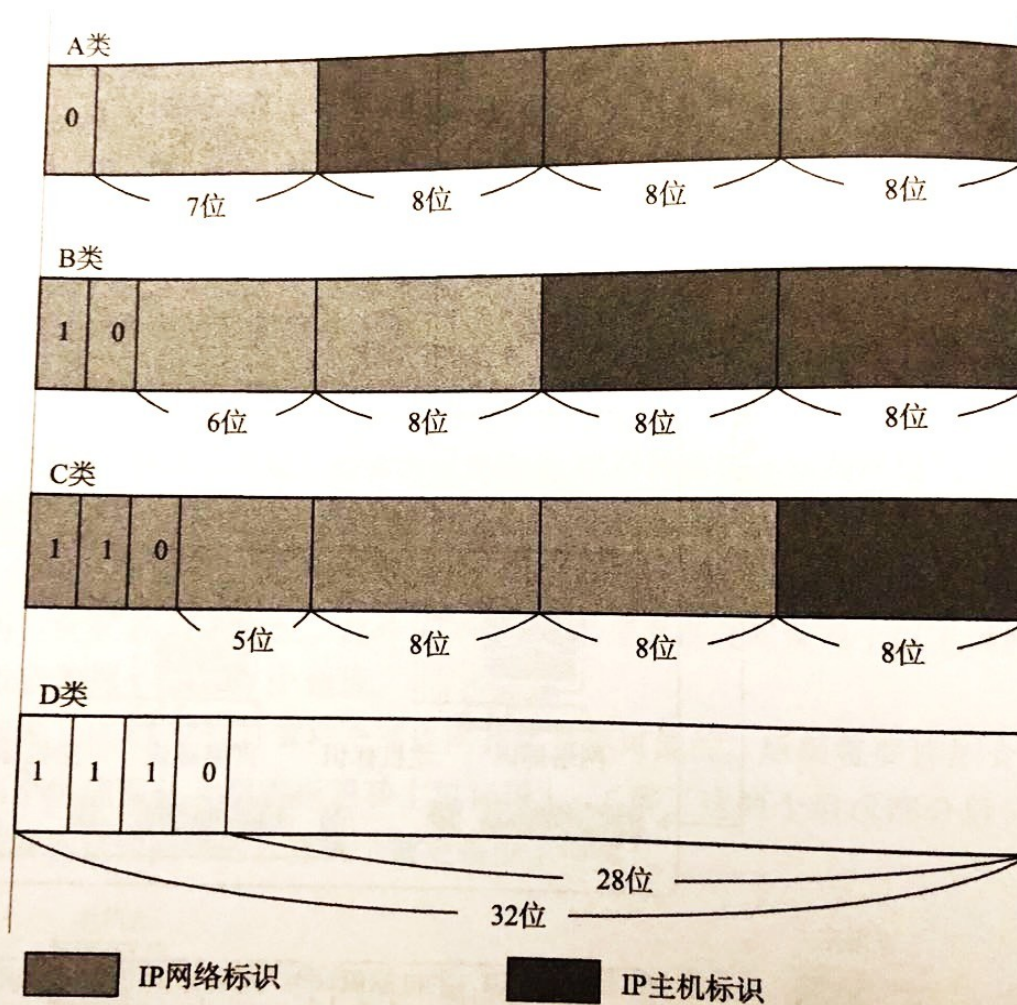
0x0e984883

# Ip 地址分类

---

- A 类地址
  - 0 开头：0.0.0.0 ~ 127.0.0.0
  - 前 8 位为网络段，后 24 位为主机段，每个网络包括 16,777,214 个地址
- B 类地址
  - 10 开头：128.0.0.0~191.255.0.0
  - 前 16 位为网络段，后 16 位为主机段，每个网络包括 65,534 个地址
- C 类地址
  - 110 开头：192.0.0.0~223.255.255.0
  - 前 24 位为网络段，后 8 位为主机段，每个网络包括 254 个地址
- D 类地址
  - 1110 开头：224.0.0.0 ~ 239.255.255
  - 一般用于多播 (multicast)

# Ip 地址分类



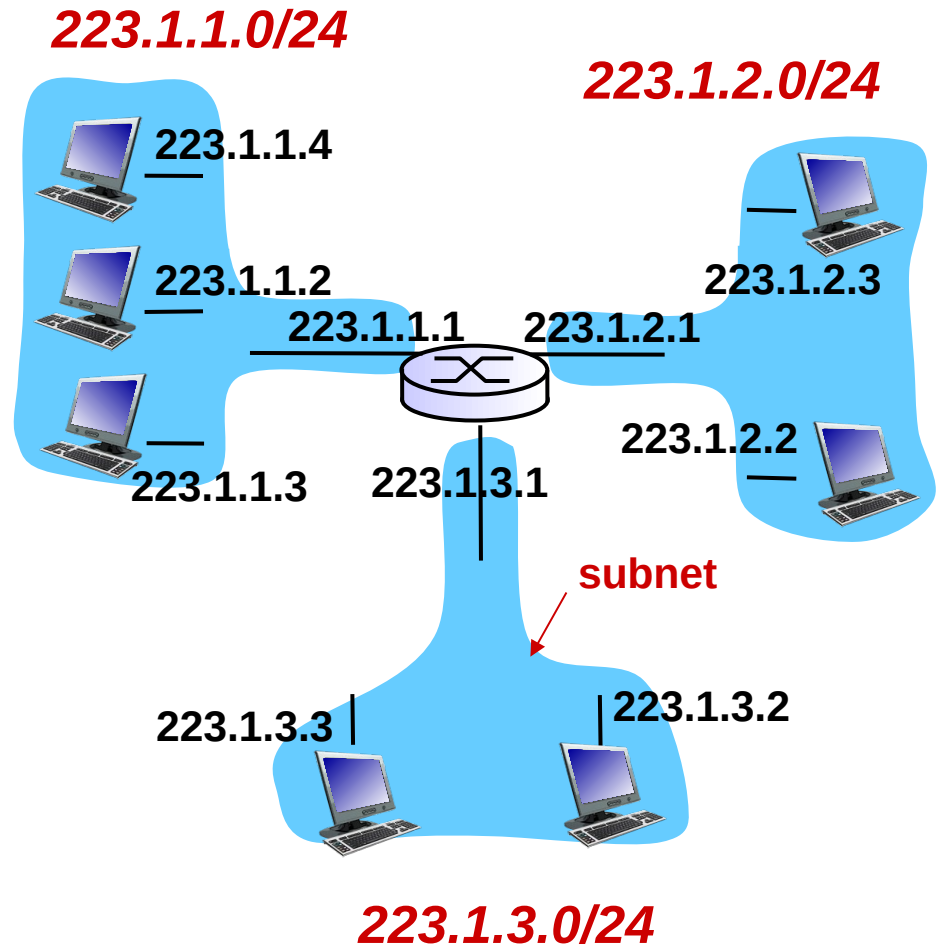
# IP 地址

---

- 内网 IP 段
  - 10.xx.xx.xx
  - 172.16.xx.xx - 172.31.xx.xx
  - 192.168.xx.xx
- 子网掩码
  - 位数：区分网络段和主机段（1 为网络段，0 为主机段）
  - E.g., 255.255.255.0
- 网关
  - 通常是子网中的第一个合法主机地址，负责对外联系，消息转发
  - E.g., 192.168.0.1

# 子网 (Subnets)

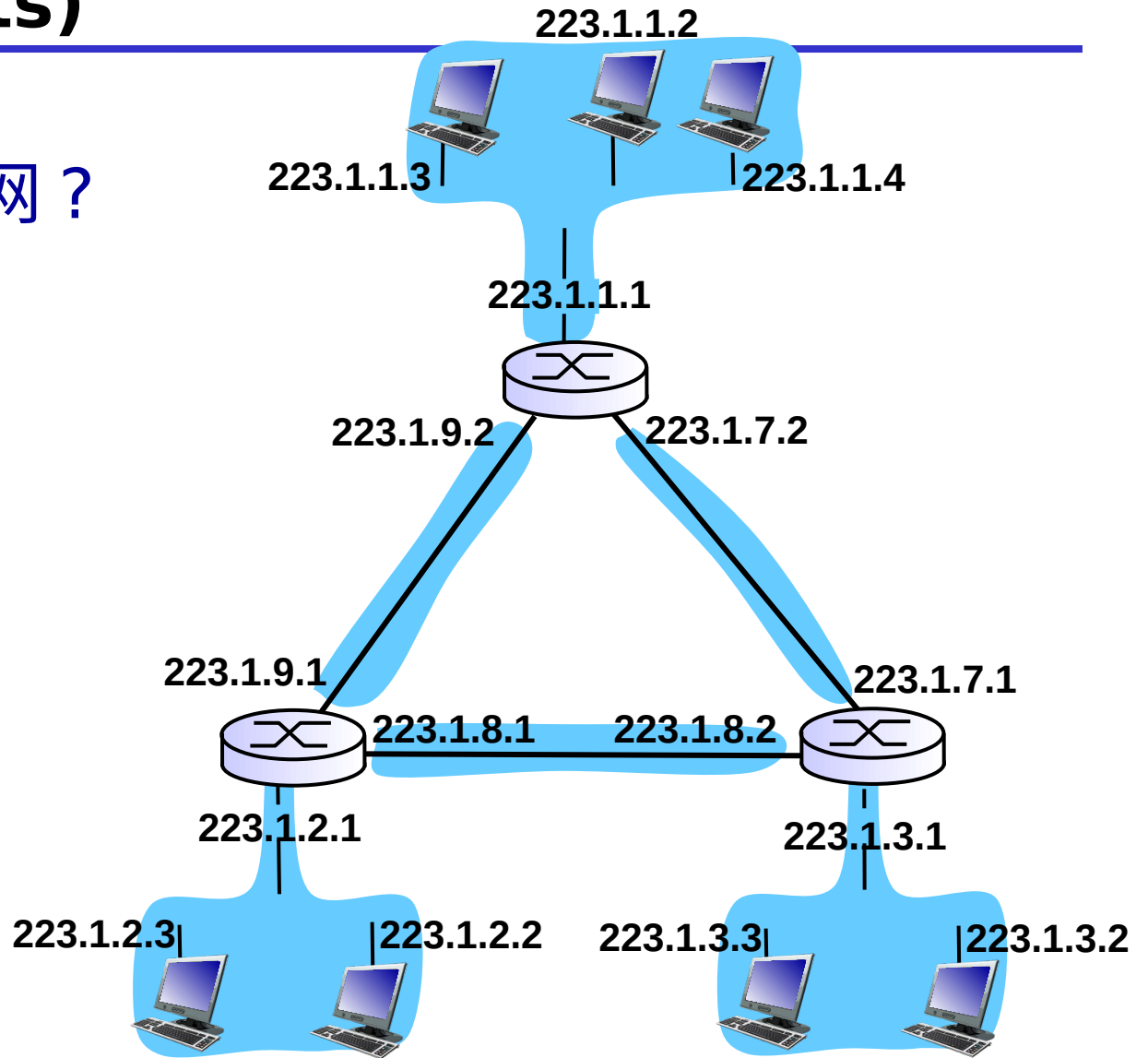
- IP 地址：
  - 高 n 位为子网段
  - 低 31-n 位为主机段
- 什么是子网？
  - 具有相同子网段的 ip 地址的主机构成一个子网
  - 同一个子网内的主机无需 3 层路由即可通信 *without intervening router*



subnet mask:  
/24

# 子网 (Subnets)

图中有几个子网？



# 广播 (broadcast) 地址

---

- 本地广播
  - 例如 192.168.0.0/24 的广播地址为 192.168.0.255
- 直接广播
  - 不同网络之间的广播
  - 例如 192.168.0.0/24 主机向 192.168.1.255/24 的目标地址发送 IP 包
  - 则 192.168.1.1 ~ 192.168.1.254 都能收到
  - \* 主机地址不能全 0 或全 1
    - 全 0 表示网络地址
    - 全 1 表示广播

## 课堂练习

---

- 一个用户的 IP 是 192.168.45.172 ，子网掩码是 255.255.240.0
- 请问网关是多少？广播地址是多少？

## 课堂答案

---

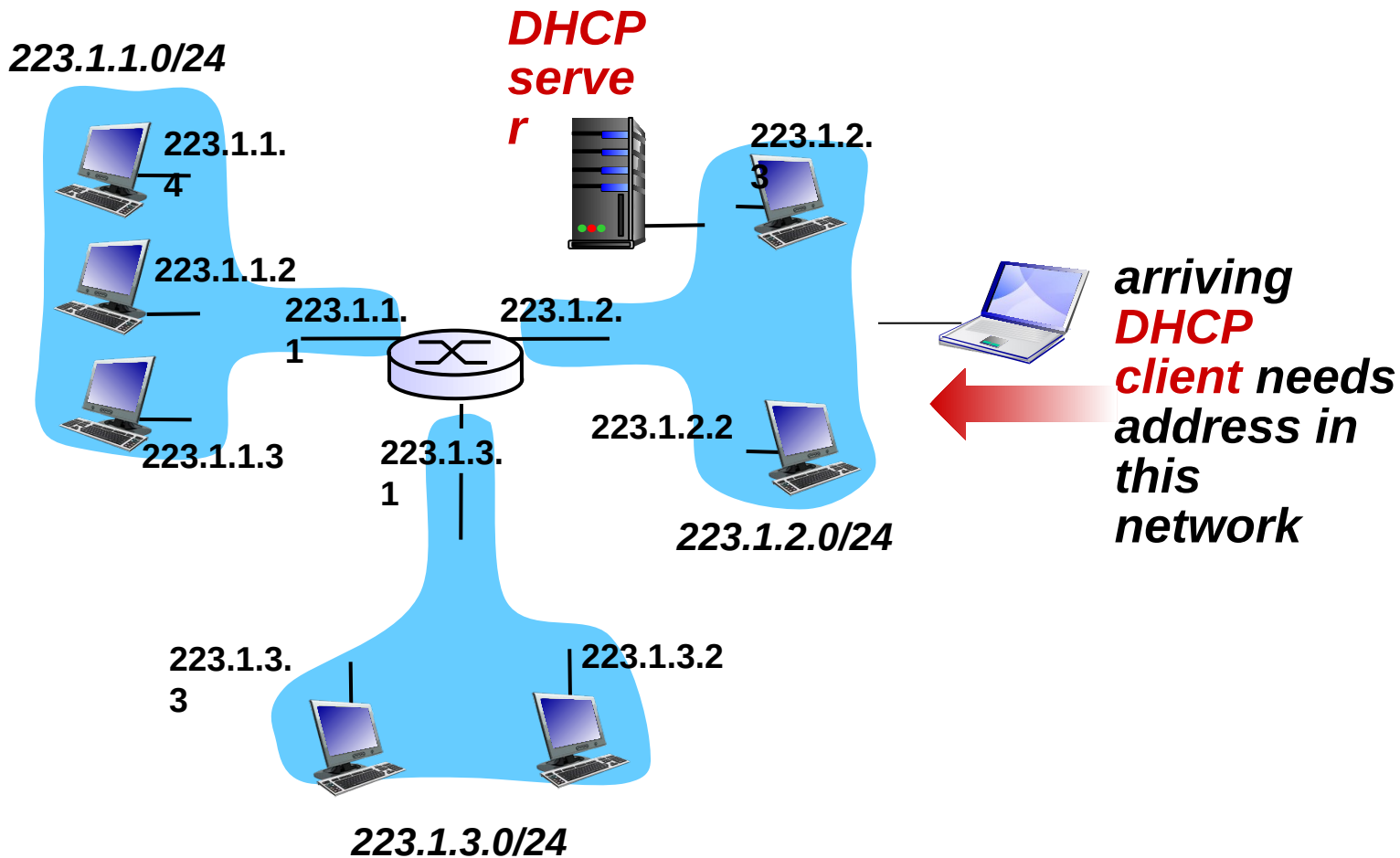
- 一个用户的 IP 是 192.168.45.172 ，子网掩码是 255.255.240.0
- 请问网关是多少？广播地址是多少？
- 240 = 0xF0 ，所以前 20 位为网络段，后 12 位为主机段
- 192.168.45(0x2D).172 所在网络地址范围为
  - 192.168.32(0x20).0 ~
  - 192.168.47.255
- 网关通常为 192.168.32.1
- 广播地址为 192.168.47.255

# 如何获取 IP 地址？

---

- 人工分配并写入一个系统配置文件
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- 通过 DHCP 协议 : Dynamic Host Configuration Protocol
  - 实现网络设备即插即用 (plug-and-play)
  - 这是一个基于 UDP 的应用层协议

# DHCP 应用场景



实际上，DHCP server 通过运行在网关路由器上

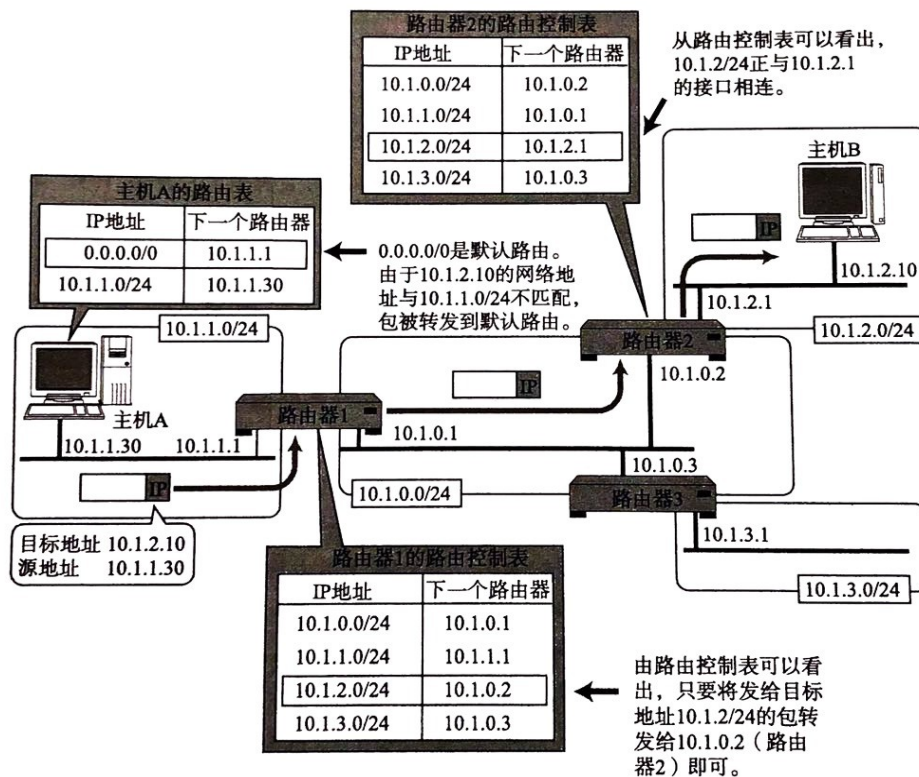
# IPv6

---

- Motivation: IPv4 地址不够用
- IPv6: 128-bit (IPv4 是 32 位)
  - 分为 8 段, 每段 16 位
  - E.g., 1080: 0:0:0:8:800:200C:417A
  - 可以简化为 1080::8:800:200C:417A

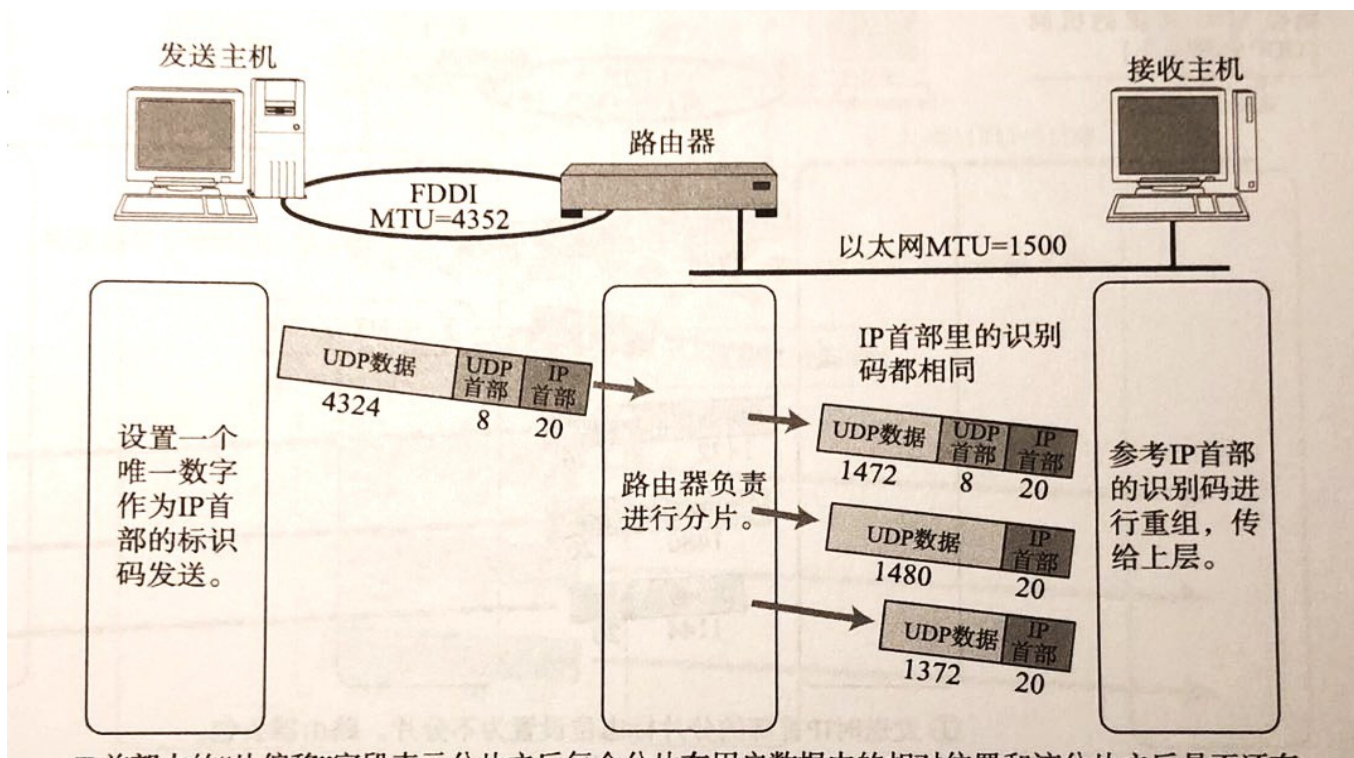
# IP 层工作：路由控制

- 路由器：
  - 包含路由表，在其各个网络端口间转发消息
    - 先解开 IP 包，获得目标地址，再重新封装消息



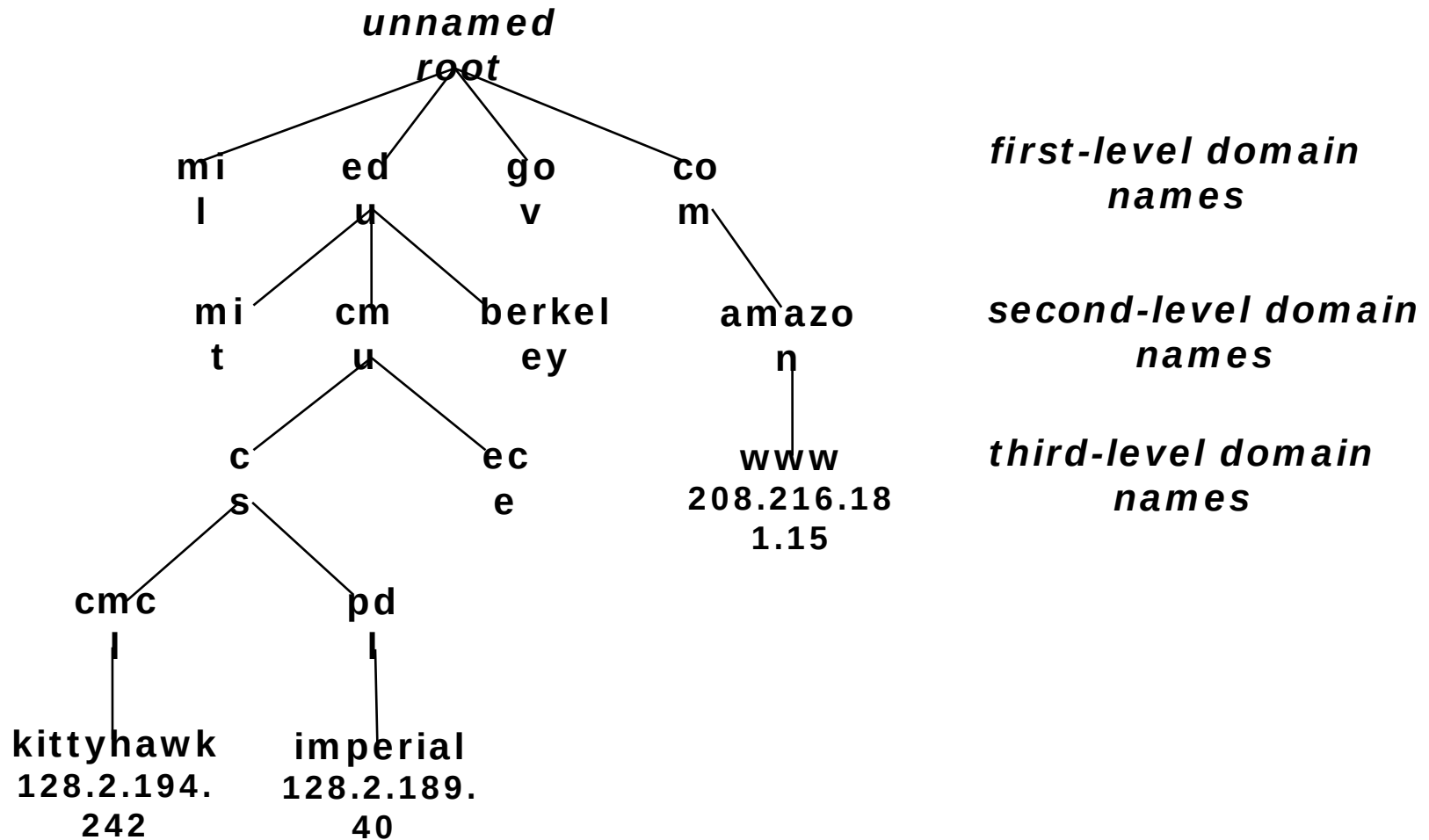
# IP 层工作： IP 包分割处理与再构成处理

- 不同网络的最大传输单元（MTU）可能不同
  - 最小 68B，最大 65535B
  - 所以可能需要 IP 包重组



# 域名

---



# 域名

---

域	分配给
com	商业组织
edu	教育机构
gov	美国政府组织
mil	美国军事组织
net	主要网络提供商
org	非赢利组织
arpa	临时 ARPA 域（仍在使用）
int	国际组织

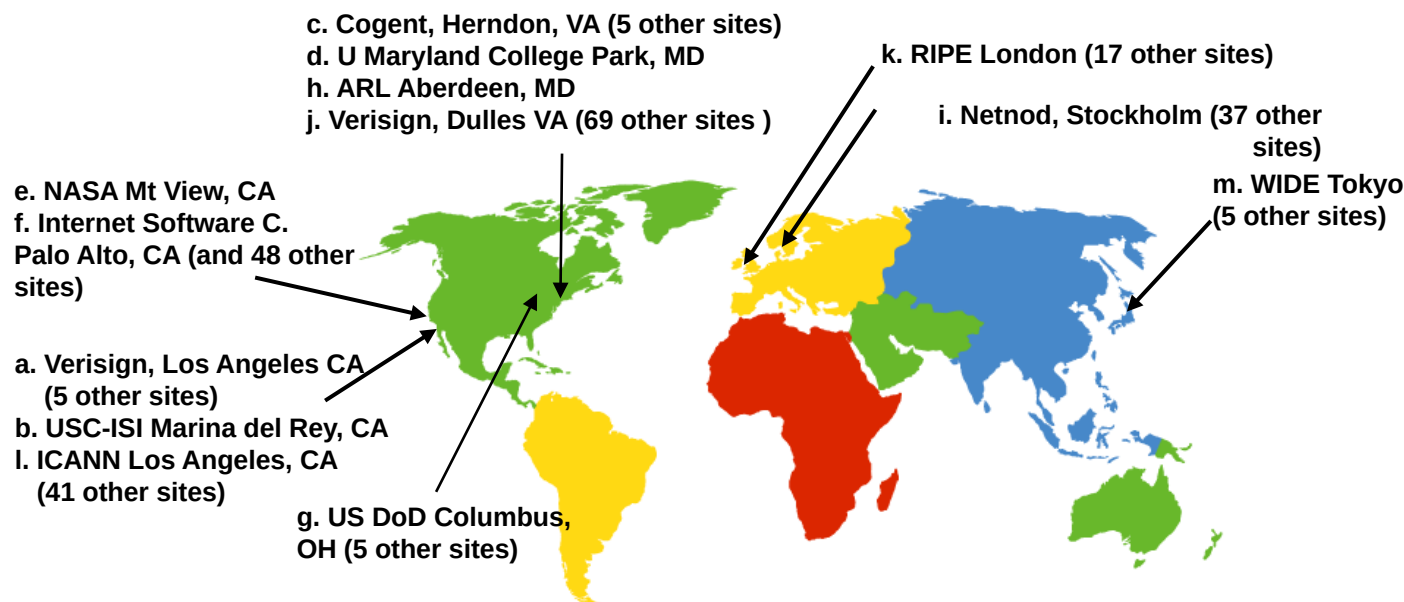
# Domain Naming System (DNS)

---

- DNS 是互联网上的一个大型分布式数据库，维护着域名到 IP 地址的映射
- DNS 系统可以看作是一个巨型的分布式 KV 数据库，具有分级缓存架构

# 根服务器 (root name servers)

- 全球有 13 台根服务器
- 局部 name server 无法解析域名时，会向根服务器发起查询
  - 根服务器对于未缓存的映射，会进一步查询权威服务器 (authoritative name server)



# DNS host entries

---

- 有一个特殊的域名 localhost ，固定映射到本地环回地址 (loopback address) 127.0.0.1

```
hank@cds:~$ ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.168.3.124 netmask 255.255.255.0 broadcast 10.168.3.255
    inet6 fe80::b396:f0a8:9510:dfc9 prefixlen 64 scopeid 0x20<link>
    ether 8c:ec:4b:d1:eb:b8 txqueuelen 1000 (Ethernet)
    RX packets 2361418 bytes 2256238475 (2.2 GB)
    RX errors 0 dropped 2355 overruns 0 frame 0
    TX packets 2603743 bytes 378854588 (378.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enx806d974452c1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 80:6d:97:44:52:c1 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1221962 bytes 1014363235 (1.0 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1221962 bytes 1014363235 (1.0 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

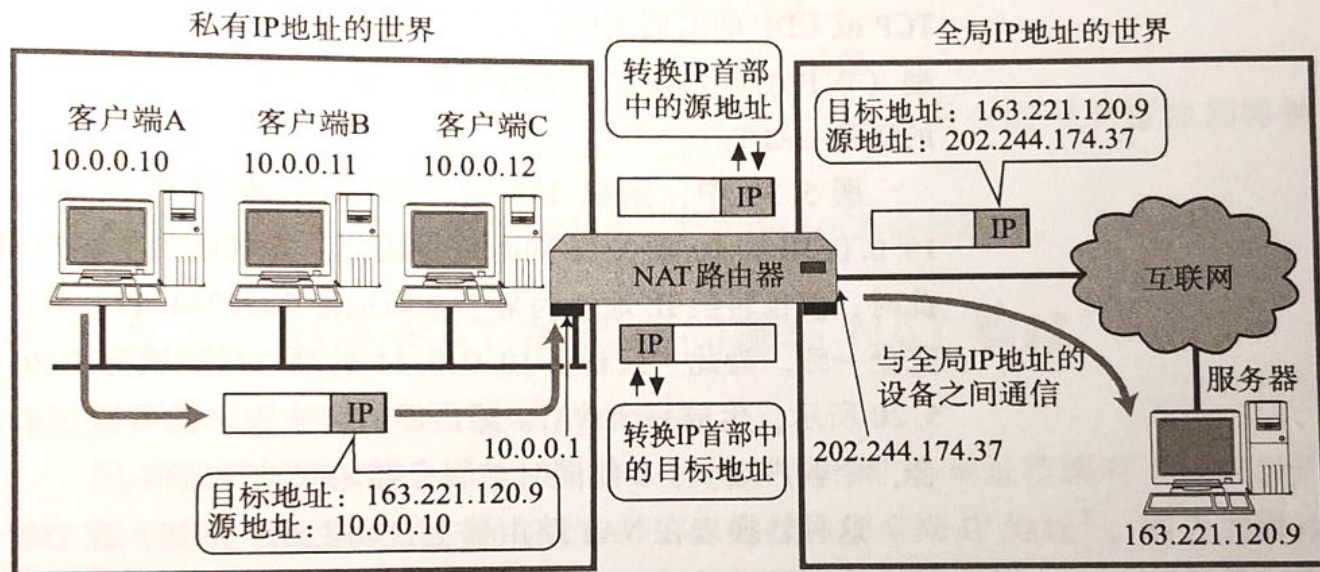
# DNS host entries

---

- 不同类型的映射：
  - **1-1**: 一个域名对应一个 ip
  - **M-1**: 多个域名对应一个 ip
    - `google.com.cn` and `google.cn` both map to `74.125.128.160`
  - **M-N**: 多个域名对应多个 ip:
    - `baidu.com` and `baidu.com.cn` map to three different IP addresses
  - **1-?**: 有些域名没有对应的 ip
    - `Csm1.cs.cmu.edu`

# NAT (Network Address Translation)

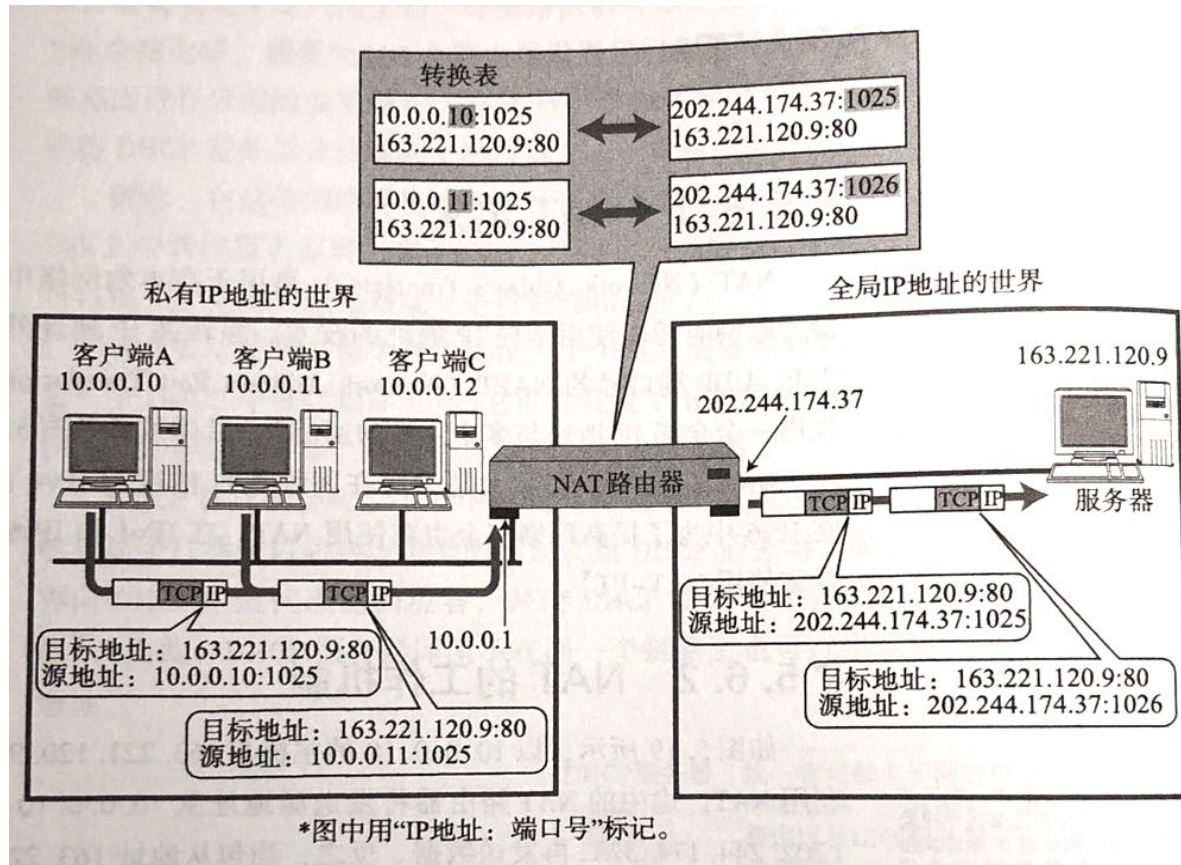
- Network Address Translation (网络地址转换)
  - 在本地网络使用私有地址，在连接 Internet 时转而使用全局 IP 地址



局域网内设置为私有IP地址，在与外部通信时被替换成全局IP地址。

# NAT (Network Address Translation)

- 内网多台机器同时访问外网，全局 IP 可能不够用
- 不同内外主机对应全局 IP 的不同端口



## 常用网络命令

---

- ping 192.168.0.100/www.baidu.com
- Ipconfig / ifconfig

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether f0:18:98:60:08:6a
inet6 fe80::1497:b083:a4ac:efd%en0 prefixlen 64 secured scopeid 0xa
inet 192.168.0.49 netmask 0xffffffff broadcast 192.168.0.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
```

- Arp: 确定 Ip 地址对应的 MAC 地址
  - Arp -a; arp -a IP

# 常用网络命令

---

- **Tracert** / traceroute
  - 跟踪一台主机到世界上任意一台主机的路由
  - 例如 traceroute [www.yangbo.pro](http://www.yangbo.pro)
- **route**
  - 显示、修改本机路由表
  - 例如 route print
- **netstat**
  - 显示活动的 TCP 连接、计算机侦听的端口、以太网统计信息、IP 路由表、IPv4 统计信息以及 IPv6 统计信息
  - 使用时如果不带参数， netstat 显示活动的 TCP 连接

# TCP 与 UDP

---

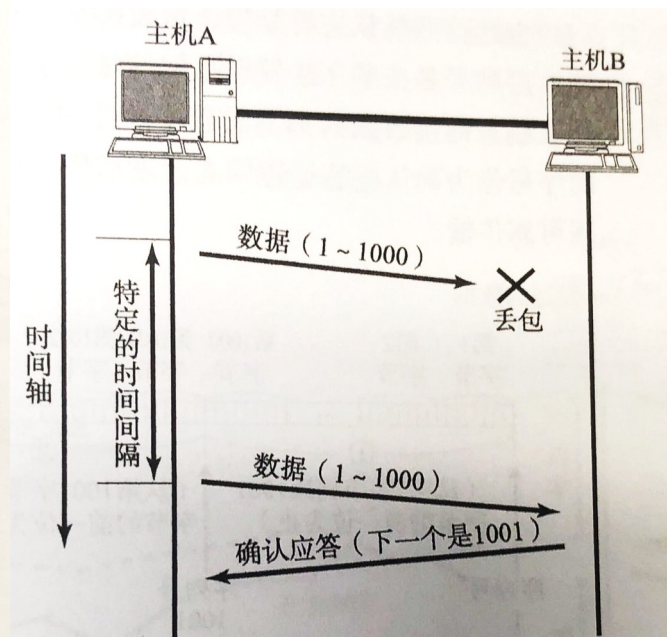
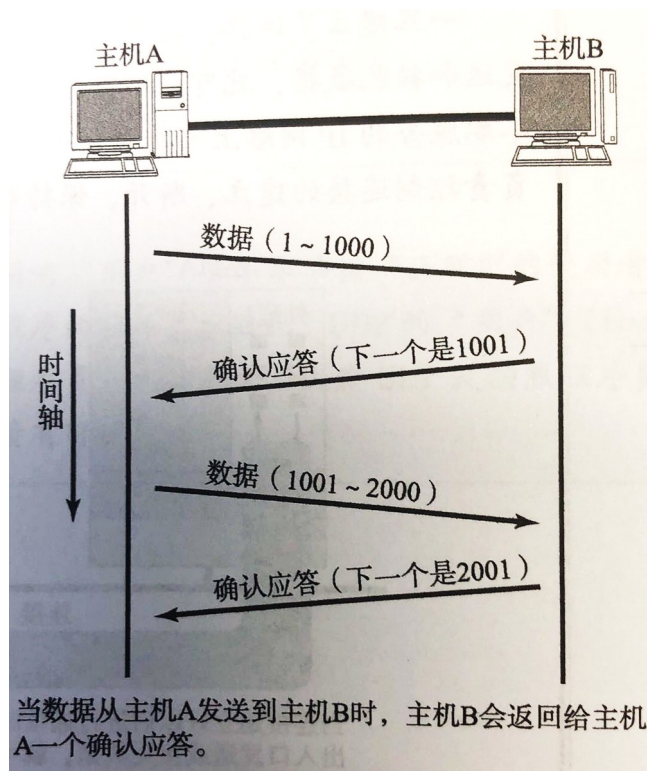
- IP 层：主机 -> 主机（通过 IP 地址）
- 但是究竟是发给主机上的哪个进程处理消息呢？
  - 端口
  - 传输层协议（TCP/UDP）

# TCP 与 UDP

- TCP (Transmission Control Protocol)

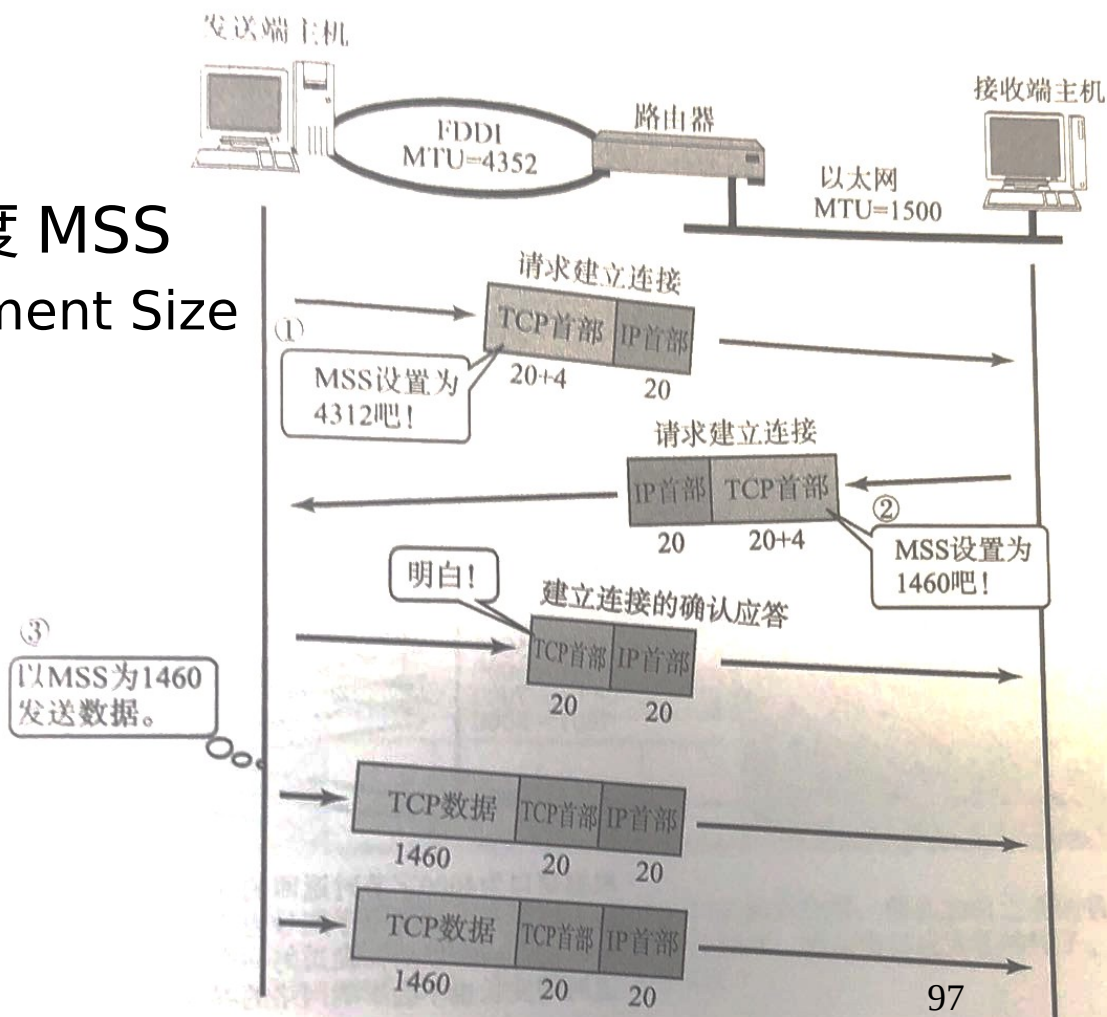
- 有连接的**可靠**连接

- 顺序控制
    - 重发控制
    - 拥塞控制



# TCP 与 UDP

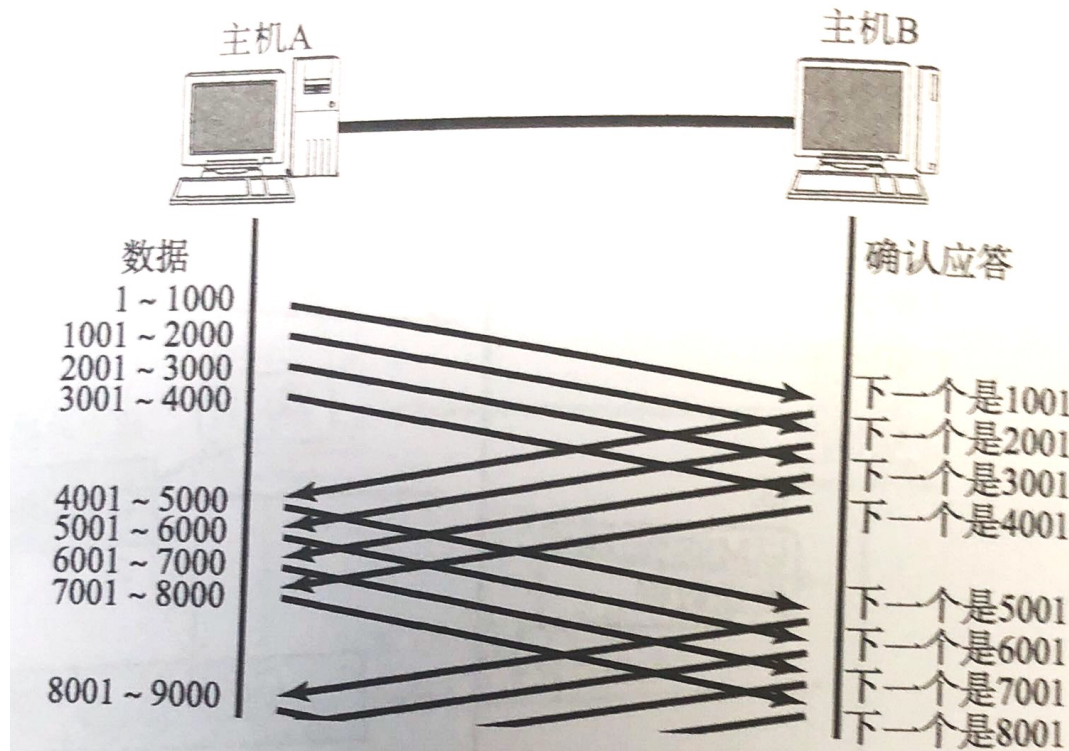
- TCP 三次握手
  - 建立连接
  - 确定最大消息长度 MSS
    - Maximum Segment Size



(图中数字表示数据的长度, 单位为8位字节。确认应答的段有一部分已省略。)

# TCP 与 UDP

- TCP 的窗口机制
  - 批量确认，提高效率
  - 网络延迟较大



# TCP 与 UDP

---

- UDP (User Datagram Protocol)
  - 无连接，没有额外的控制
  - 发送数据之前不需要先建立连接
  - 简单、高效
    - TCP 首部开销 20 字节 ;UDP 的首部只有 8 个字节
  - UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）

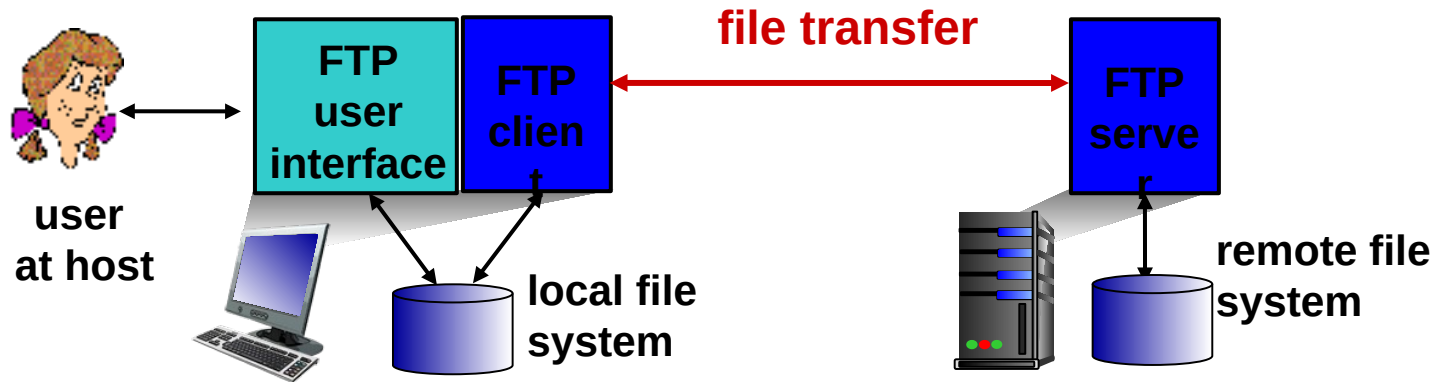
# 应用层

---

- Telnet
- SSH
- FTP
- Electronic mail
  - SMTP, POP3, IMAP
- Web and HTTP

# FTP: the file transfer protocol

---



- ❖ transfer file to/from remote host
- ❖ client/server model
  - *client*: side that initiates transfer (either to/from remote)
  - *server*: remote host
- ❖ ftp: RFC 959
- ❖ ftp server: port 21

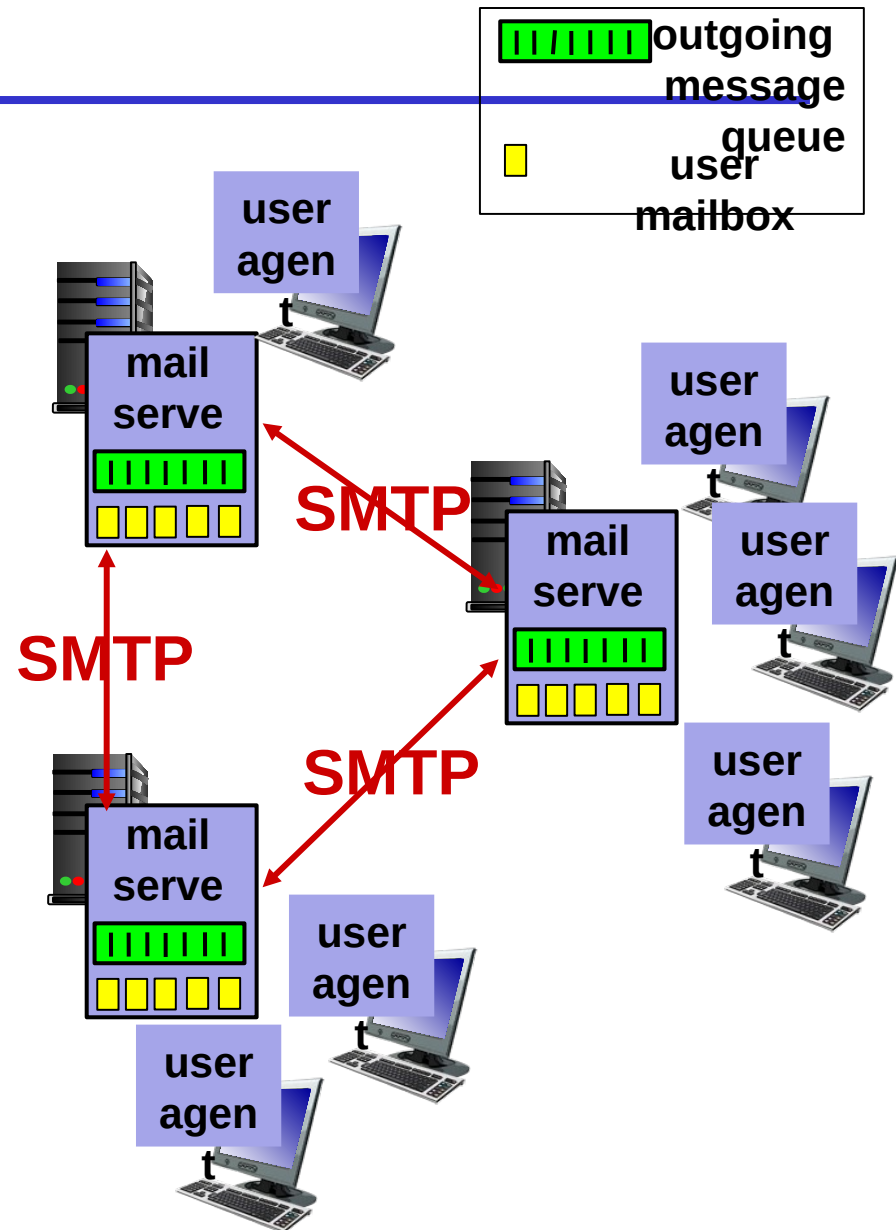
# Electronic mail

## *Three major components:*

- user agents
- mail servers
- simple mail transfer protocol: SMTP

## *User Agent*

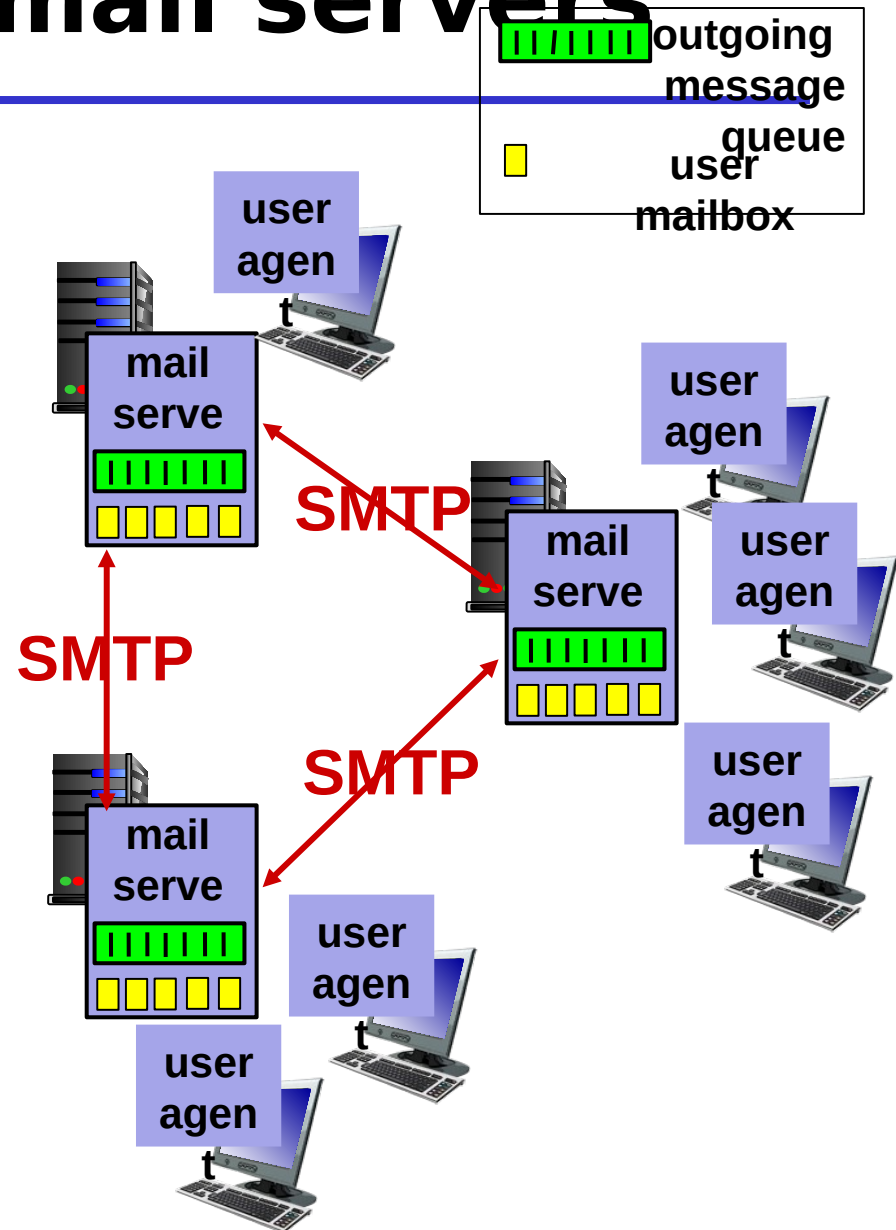
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird, iPhone mail client
- outgoing, incoming



# Electronic mail: mail servers

## mail servers:

- *mailbox* contains incoming messages for user
- *message queue* of outgoing (to be sent) mail messages
- *SMTP protocol* between mail servers to send email messages
  - client: sending mail server
  - “server”: receiving mail server



# Web and HTTP

---

*First, a review...*

- *web page* consists of *objects*
- object can be HTML file, JPEG image, Java applet, audio file,...
- web page consists of *base HTML-file* which includes *several referenced objects*
- each object is addressable by a *URL*, e.g.,

`www.someschool.edu/someDept/`

`pic.gif`

host  
name

path  
name

# HTTP overview

---

## HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client*: browser that requests, receives, (using HTTP protocol) and “displays” Web objects
  - *server*: Web server sends (using HTTP protocol) objects in response to requests



# HTTP overview (continued)

---

## *uses TCP:*

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

## *HTTP is “stateless”*

- server maintains **no** information about past client requests

**protocols that maintain “state” are complex!**

- ❖ **past history (state) must be maintained**
- ❖ **if server/client crashes, their views of “state” may be inconsistent, must be reconciled**