



计 算 机 网 络

西北工业大学 软件学院

计算机网络

第2章 应用层

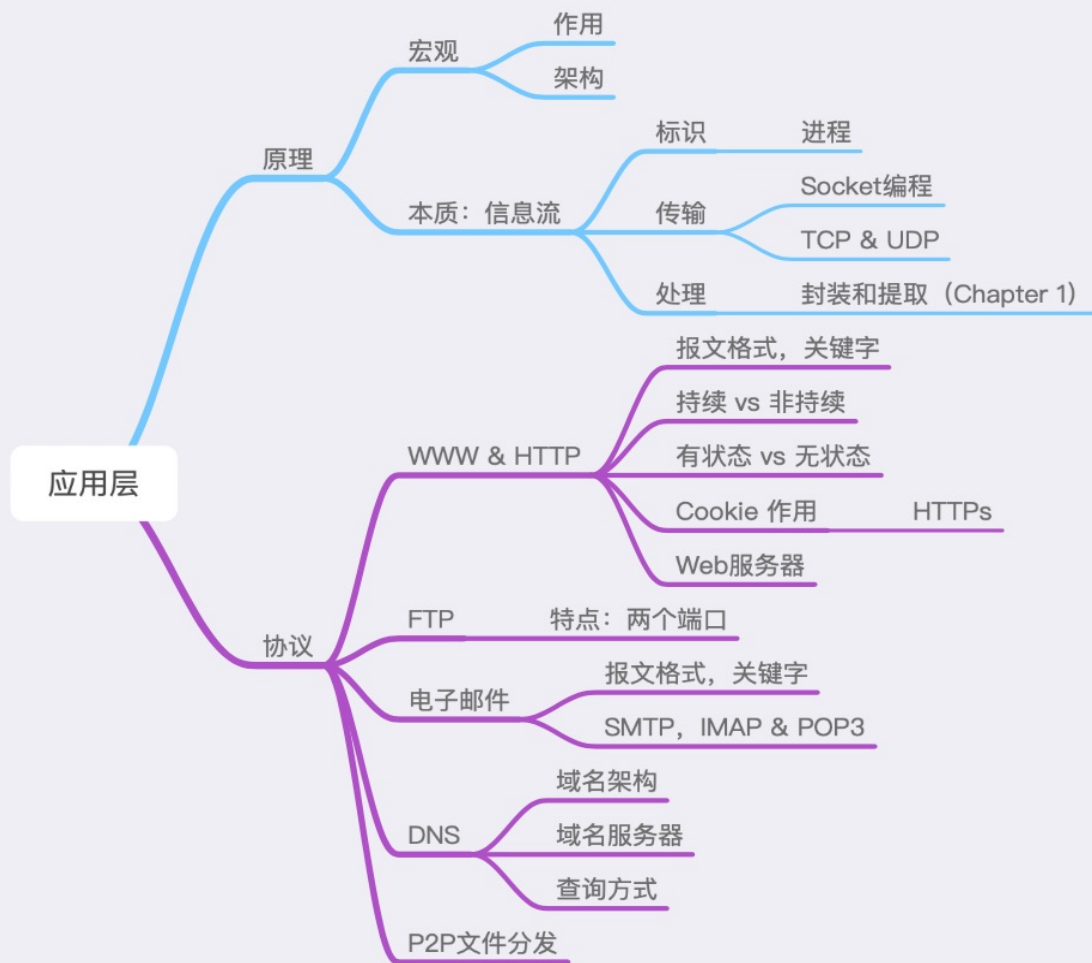
上网？做什么？

- ❑ 浏览新闻
- ❑ 聊天
- ❑ 听音乐
- ❑ 看电影
- ❑ 看电视
- ❑ 收发Email
- ❑ 下载软件

- ❑ 网上图书馆
- ❑ 网上商店
- ❑ 网上银行
- ❑ 网上医院
- ❑ 网上大学
- ❑ 电子商务

.....

第2章 应用层：概览



第2章 应用层：功能

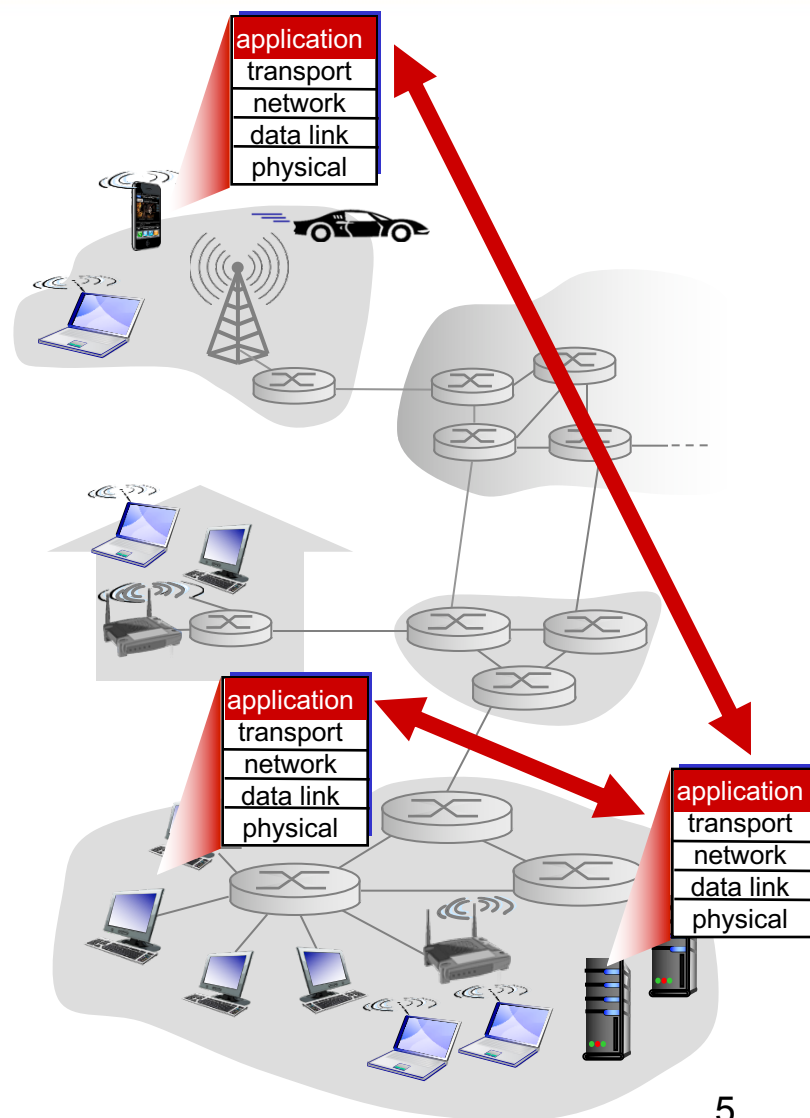
编写程序能够

- ◆能够在不同端系统上运行
- ◆信息、软件和硬件的共享

并非所有的设备都有应用层

- ◆三层协议软件(路由器)
- ◆二层协议软件(交换机、路由器)

本质：信息的流动



2 应用层

应用层面临的问题：

1. 应用层使用的通信网络的架构？
2. 应用层内部，应用层之间如何区分不同的APP消息？
3. 应用层如何把消息传输出去？接收端如何处理从而得到正确的消息

➤ 2.1 应用层协议原理

➤ 2.2 Web应用和HTTP协议

➤ 2.3 文件传输协议：FTP

➤ 2.4 电子邮件

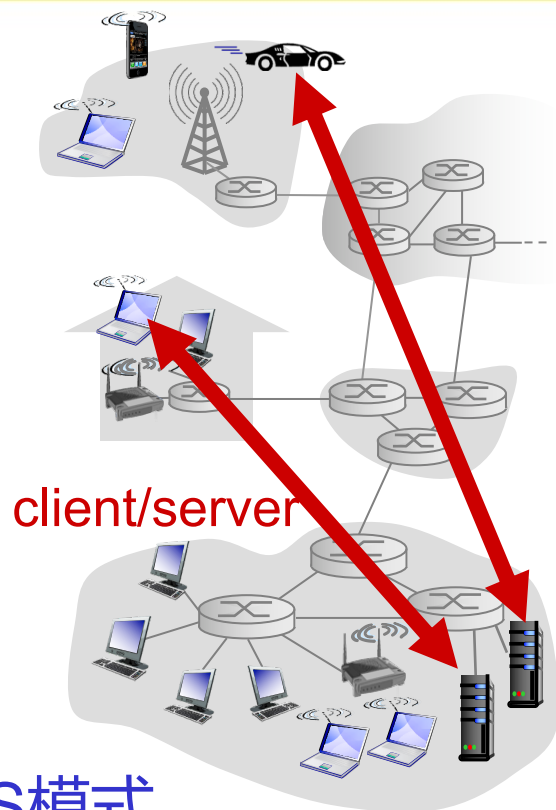
➤ 2.5 域名系统DNS

➤ 2.6 P2P 应用

2.1应用层协议原理：体系结构

- 客户机/服务器
- 对等 (P2P)
- 客户机/服务器与P2P的混合

2.1应用层协议原理：体系结构



C/S apps:
Web, FTP, E-mail



C/S模式

- ◆ 集中结构，一对多
- ◆ 服务器共享资源，客户机资源不共享
- ◆ 服务器可能负载过重
- ◆ 网络带宽限制

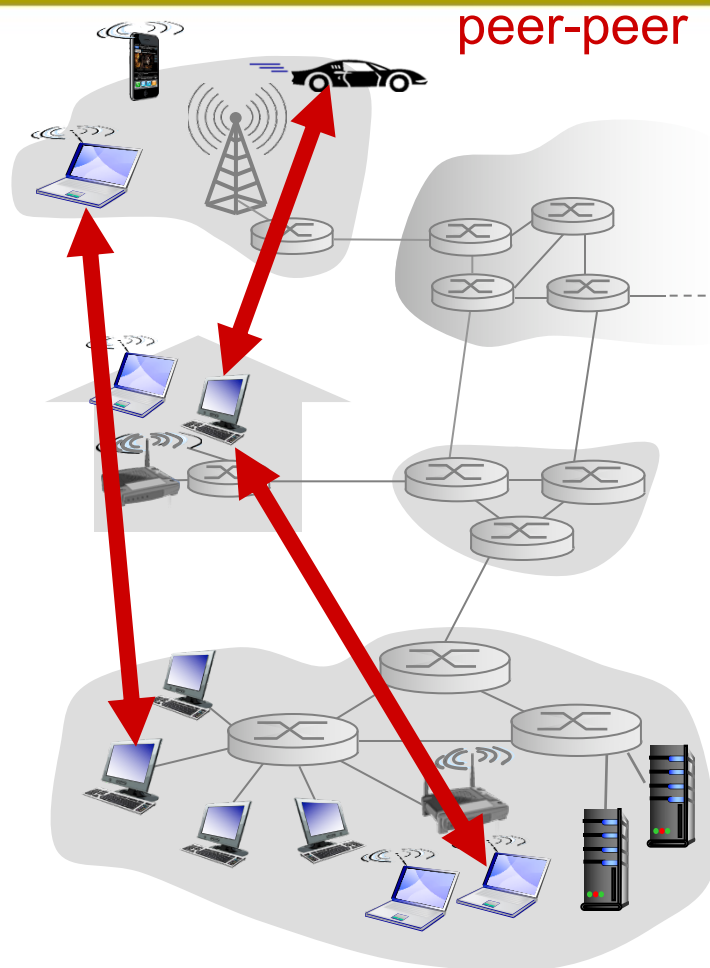
2.1应用层协议原理：体系结构

P2P模式

- 非集中结构，多对多
- 节点具备客户与服务器双重特性
- 充分利用终端资源
- 可扩展性好

类型

- 纯P2P：Gnutella
- 混合P2P：迅雷



2.1应用层协议原理：进程

进程：运行在端系统中的程序。

- 在同一台主机中：两个进程使用进程间通信 (由操作系统定义).
- 在不同的主机中：进程通过**消息交换机制**（**应用层协议**）通信

客户机进程: 发起通信的进程

服务器进程：等待联系的进程

注意：具有P2P体系结构的应用程序**同时**具有客户进程和服务器进程

2.1应用层协议原理：进程

一个主机中可能同时运行有多个进程，接收端从网络收到消息后要交给哪个进程？

- 对于接收报文的进程，必须具有一个标识
- 一台主机具有一个独特的32比特的IP地址
- **问题：**主机的IP地址足以标识该进程？
- **答案：**否，同一台主机上能够运行许多进程
- 标示符包括IP地址和与主机上该进程相关的端口号.
- 端口号例子:
 - ◆ HTTP 服务器: 80
 - ◆ 电子邮件服务器: 25

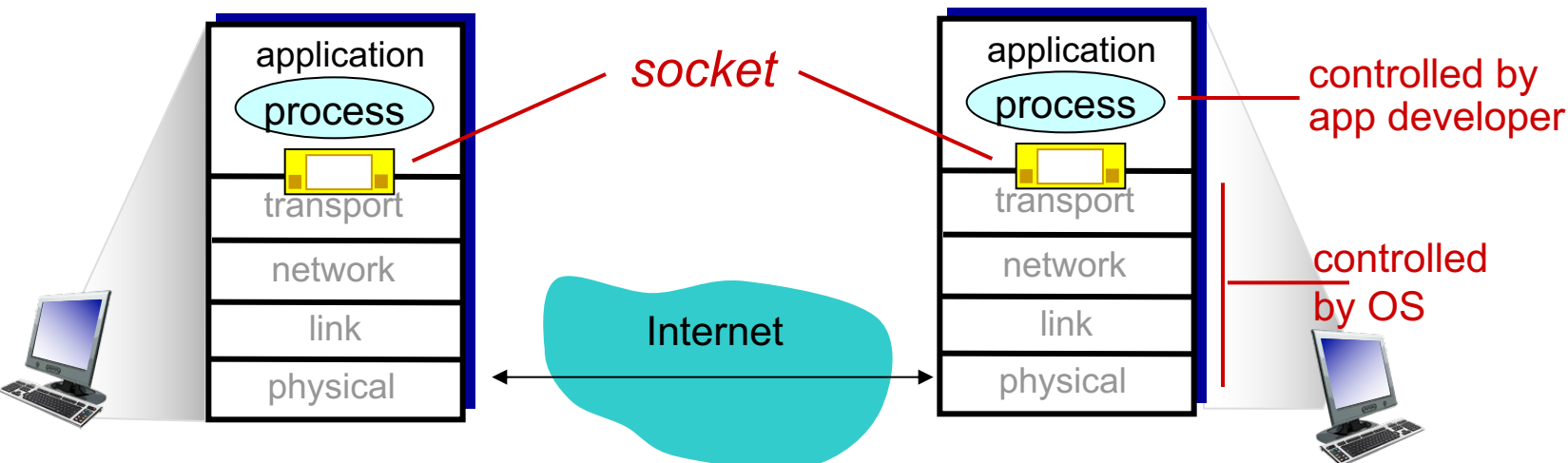
应用层需要一种机制把
上述信息交付给传输层/
运输层:Socket

2.1应用层协议原理：套接字

- Berkeley UNIX 操作系统定义了一种 API，它又称为套接字接口(socket interface)。
- 微软公司在其操作系统中采用了套接字接口 API，形成了一个稍有不同的 API，并称之为 Windows Socket。
- AT&T 为其 UNIX 系统 V 定义了一种 API，简称为 TLI (Transport Layer Interface)。

2.1应用层协议原理：套接字

套接字 (socket) 是通信的基石，是支持TCP/IP协议的网络通信的基本操作单元。它是网络通信过程中端点的抽象表示，包含进行网络通信必须的五种信息：**连接使用的协议，本地主机的IP地址，本地进程的协议端口，远地主机的IP地址，远地进程的协议端口。**



- 进程通过**套接字**在网络上发送/接收报文
- 套接字类似于门
上联应用进程，下联网络协议栈

等一下，Socket里的协议是？

2.1应用层协议原理：传输手段

可靠的数据传输

- 某些应用（如音频）能够容忍某些丢失
- 其他应用（如文件传输，Telnet）要求100%可靠数据传输

定时

- 某些应用（如因特网电话、交互式游戏）要求“有效的”低时延

吞吐量

- 某些应用（如多媒体）要求“有效的”最小量的带宽
- 其他应用（“弹性应用”）充分利用它们获得的所有带宽

安全

- 加密，数据完整性，...

2.1应用层协议原理：传输手段

| 应用程序 | 数据丢失 | 带宽 | 时间敏感 |
|---------|------|-------------------------------------|---------------|
| 文件传输 | 不能丢失 | 弹性 | 不 |
| 电子邮件 | 不能丢失 | 弹性 | 不 |
| Web 文档 | 不能丢失 | 弹性 | 不 |
| 实时音频/视频 | 容忍丢失 | 音频: 5kbps-1Mbps 视频: 10kbps-5Mbps | 是, 100's msec |
| 存储音频/视频 | 容忍丢失 | 同上 | 是, 几秒 |
| 交互式游戏 | 容忍丢失 | 几kbps以上 | 是, 100 msec |
| 即时讯息 | 不能丢失 | 弹性 | |

2.1应用层协议原理：传输手段

TCP服务（不能丢失）：

- 面向连接：客户机和服务器之间所需的建立
- 可靠传输：在发送和接收进程之间
- 流控制：发送方不会淹没接收方
- 拥塞控制：当网络过载时抑制发送方
- 并不提供：定时，最小带宽保证

UDP服务（容忍丢失）：

- 在发送进程及接收进程之间的不可靠数据传输
- 不提供：建立连接建立，可靠性，流控，拥塞控制，定时或带宽保证

问题:为什么需要UDP?

2.1应用层协议原理：传输手段

| 应用 | 应用层协议 | 下面的传输协议 |
|---------|--------------------|---------|
| 电子邮件 | SMTP [RFC 2821] | TCP |
| 远程终端访问 | Telnet [RFC 854] | TCP |
| Web | HTTP [RFC 2616] | TCP |
| 文件传输 | FTP [RFC 959] | TCP |
| 远程文件服务器 | NFS [McKusik 1996] | UDP或TCP |
| 流媒体 | 通常专用 (YouTube) | UDP或TCP |
| 因特网电话 | 通常专用 (Skype) | 典型用UDP |

- 2.1 应用层协议原理
- **2.2 Web应用和HTTP协议**
- 2.3 文件传输协议：FTP
- 2.4 电子邮件
- 2.5 域名系统DNS
- 2.6 P2P 应用

2.2 Web和HTTP



2.2 Web和HTTP

- Web : World wide Web , 万维网 , 也称Web , 是一种互联网应用。
- web : 网页 , 网站
- 发展
 - ◆ Web1.0 : HTML网页
 - ◆ Web2.0 : 社交网站、博客、 Wiki

2.2 Web和HTTP:万维网

- 万维网是**分布式超媒体**(hypermedia)系统，它是**超文本**(hypertext)系统的扩充。
- 一个超文本由多个信息源链接成。利用一个链接可使用户找到另一个文档。这些文档可以位于世界上任何一个接在因特网上的超文本系统中。超文本是万维网的基础。
- 超媒体与超文本的区别是文档内容不同。超文本文档仅包含文本信息，而超媒体文档还包含其他表示方式的信息，如图形、图像、声音、动画，甚至活动视频图像。

2.2 Web和HTTP：万维网

- 万维网以客户-服务器方式工作。
- 浏览器就是在用户计算机上的万维网客户程序。万维网文档所驻留的计算机则运行服务器程序，因此这个计算机也称为万维网服务器。
- 客户程序向服务器程序发出请求，服务器程序向客户程序送回客户所要的万维网文档。
- 在一个客户程序主窗口上显示出的万维网文档称为页面(page)。

2.2 Web和HTTP：万维网

万维网面对的4大挑战：

- (1) 怎样标志分布在整個因特网上的万维网文档？
- 使用**统一资源定位符** URL (Uniform Resource Locator)来标志万维网上的各种文档。
 - 使每一个文档在整个因特网的范围内具有唯一的标识符 URL。

2.2 Web和HTTP：万维网

(2) 用何协议实现万维网上各种超链的链接？

- 在万维网客户程序与万维网服务器程序之间进行交互所使用的协议，是超文本传送协议 HTTP (HyperText Transfer Protocol)。
- HTTP 是一个应用层协议，它使用 TCP 连接进行可靠的传送。

2.2 Web和HTTP：万维网

(3) 怎样使各种万维网文档都能在因特网上的各种计算机上显示出来，同时使用户清楚地知道在什么地方存在着超链？

➤ **超文本标记语言** HTML (HyperText Markup Language)使得万维网页面的设计者可以很方便地用一个超链从本页面的某处链接到因特网上的任何一个万维网页面，并且能够在自己的计算机屏幕上将这些页面显示出来。

2.2 Web和HTTP：万维网

(4) 怎样使用户能够很方便地找到所需的信息？

- 为了在万维网上方便地查找信息，用户可使用各种的搜索工具（即搜索引擎）。

2.2 Web和HTTP：万维网

- 统一资源定位符 URL 是对可以从因特网上得到的资源的位置和访问方法的一种简洁的表示。
- URL 给资源的位置提供一种抽象的识别方法，并用这种方法给资源定位。
- 只要能够对资源定位，系统就可以对资源进行各种操作，如存取、更新、替换和查找其属性。
- URL 相当于一个文件名在网络范围的扩展。因此 URL 是与因特网相连的机器上的任何可访问对象的一个指针。

2.2 Web和HTTP：万维网

- 由以冒号隔开的两大部分组成，并且在 URL 中的字符对大写或小写没有要求。
- URL 的一般形式是：

<协议>://<主机>:<端口>/<路径>

ftp —— 文件传送协议 FTP

http —— 超文本传送协议 HTTP

News —— USENET 新闻

2.2 Web和HTTP：万维网

- 由以冒号隔开的两大部分组成，并且在 URL 中的字符对大写或小写没有要求。
- URL 的一般形式是：

<协议>://<主机>:<端口>/<路径>

<主机> 是存放资源的主机
在因特网中的域名

2.2 Web和HTTP：万维网

- 由以冒号隔开的两大部分组成，并且在 URL 中的字符对大写或小写没有要求。
- URL 的一般形式是：

<协议>://<主机>:<端口>/<路径>

有时可省略

2.2 Web和HTTP : HTTP

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

这表示使用 HTTP 协议

2.2 Web和HTTP : HTTP

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

冒号和两个斜线是规定的格式

2.2 Web和HTTP : HTTP

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

这里写主机的域名

2.2 Web和HTTP : HTTP

- 使用 HTTP 的 URL 的一般形式

http://<主机> <端口>/<路径>

HTTP 的默认端口号是 80，通常可省略

2.2 Web和HTTP : HTTP

- 使用 HTTP 的 URL 的一般形式

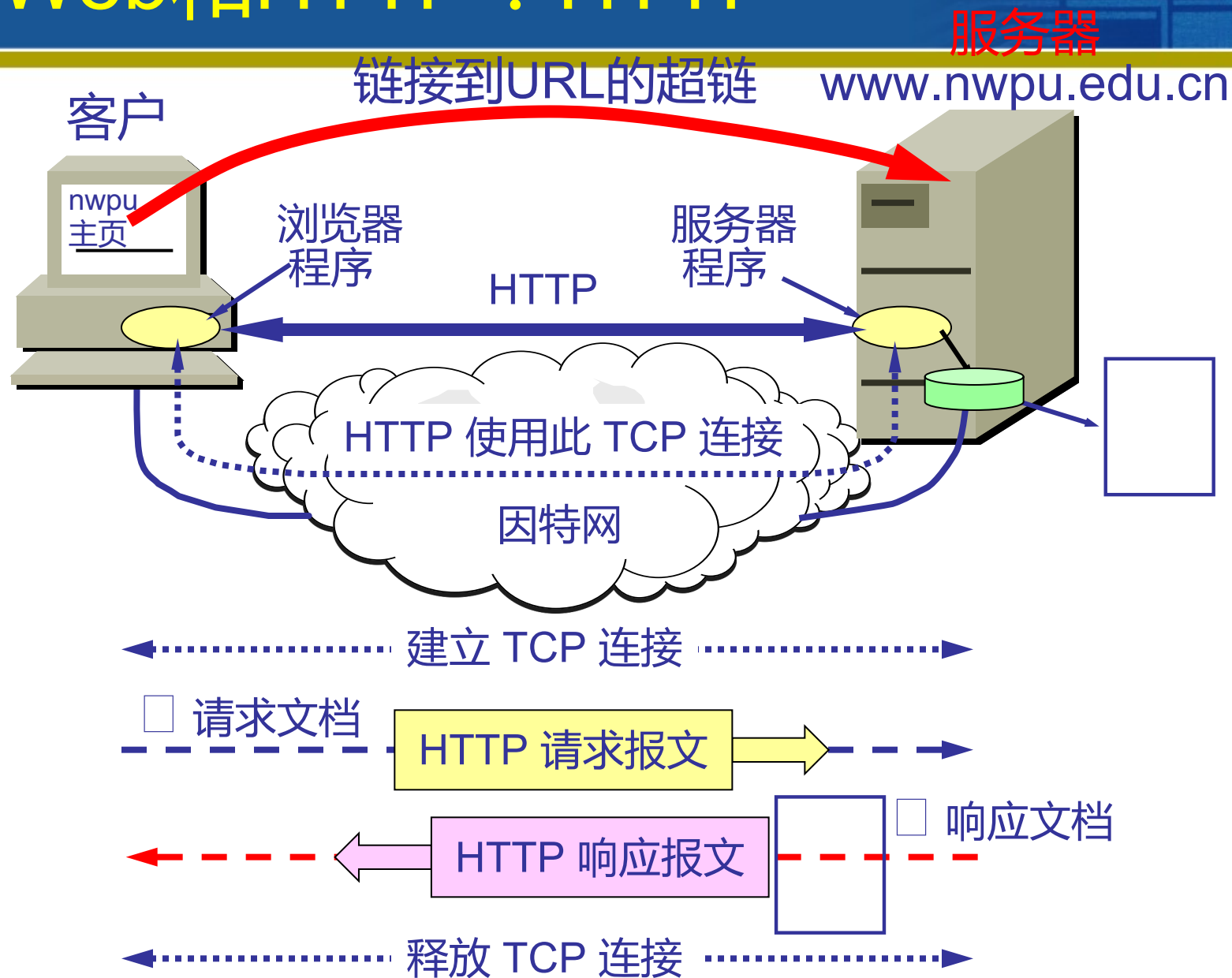
http://<主机>:<端口>/<路径>

若再省略文件的<路径>项，则 URL 就指到因特网上的某个主页(home page)。

2.2 Web和HTTP : HTTP

- 为了使超文本的链接能够高效率地完成，需要用 HTTP 协议来传送一切必须的信息。
- 从层次的角度看，HTTP 是面向事务的 (transaction-oriented) 应用层协议，它是万维网上能够可靠地交换文件（包括文本、声音、图像等各种多媒体文件）的重要基础。

2.2 Web和HTTP : HTTP



2.2 Web和HTTP：HTTP

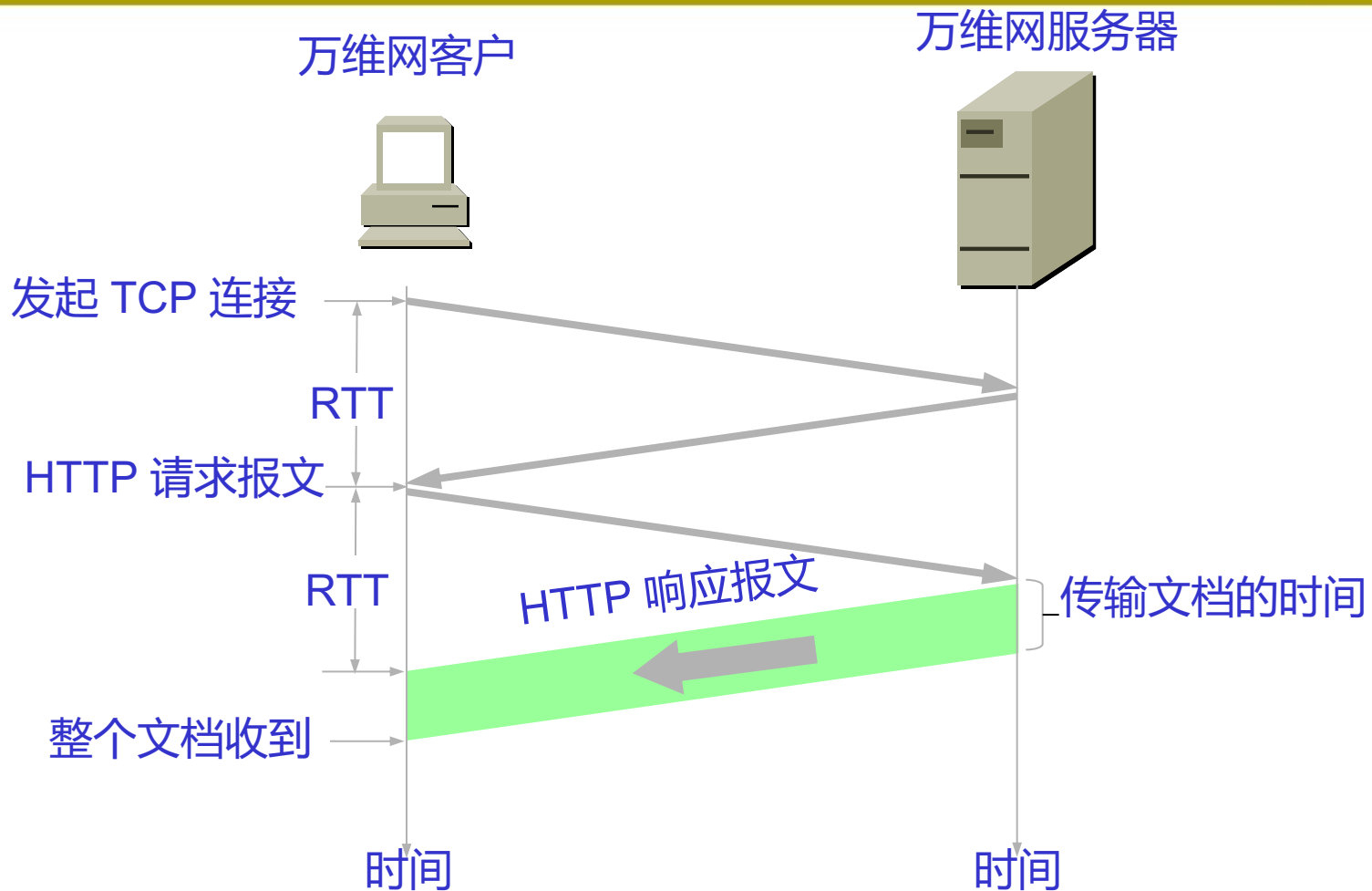
一个经典问题：当浏览网页时，点击鼠标发生了什么？

- (1) 浏览器分析超链指向页面的 URL。
- (2) 浏览器向 DNS 请求解析 `www.nwpu.edu.cn` 的 IP 地址。
- (3) 域名系统 DNS 解析出西工大服务器的 IP 地址。
- (4) 浏览器与服务器建立 TCP 连接
- (5) 浏览器发出取文件命令：
 `GET /nwpu/xyz/index.htm`。
- (6) 服务器给出响应，把文件 `index.htm` 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示 `index.htm` 中的内容。

2.2 Web和HTTP : HTTP

- HTTP 是面向事务的客户服务器协议。
- HTTP 1.0 协议是**无状态的**(stateless)。
- HTTP 协议本身也是无连接的，虽然它使用了面向连接的 TCP 向上提供的服务。

2.2 Web和HTTP : HTTP



单次请求--响应时延： $2RTT + \text{传输文档所需时间}$

如果客户发起了多次请求呢？

2.2 Web和HTTP : HTTP

- HTTP/1.1 协议使用**持续连接**。
- 万维网服务器在发送响应后仍然在一段时间内保持这条连接，使同一个客户（浏览器）和该服务器可以继续在这条连接上传送后续的 HTTP 请求报文和响应报文。
- 这并不局限于传送同一个页面上链接的文档，而是只要这些文档都在同一个服务器上就行。
- 目前一些流行的浏览器（例如，IE 6.0）的默认设置就是使用 HTTP/1.1。

2.2 Web和HTTP : HTTP

- **非流水线方式**：客户在收到前一个响应后才能发出下一个请求。这比非持续连接的两倍 RTT 的开销节省了建立 TCP 连接所需的一个 RTT 时间。但服务器在发送完一个对象后，其 TCP 连接就处于空闲状态，浪费了服务器资源。
- **流水线方式**：客户在收到 HTTP 的响应报文之前就能够接着发送新的请求报文。一个接一个的请求报文到达服务器后，服务器就可连续发回响应报文。使用流水线方式时，客户访问所有的对象只需花费一个 RTT 时间，使 TCP 连接中的空闲时间减少，提高了下载文档效率。

2.2 Web和HTTP : HTTP

➤ 两类HTTP报文：请求报文，响应报文

➤ HTTP请求报文：

◆ 例子，ASCII (人可读的格式)

The diagram shows an HTTP request in ASCII format with three annotations:

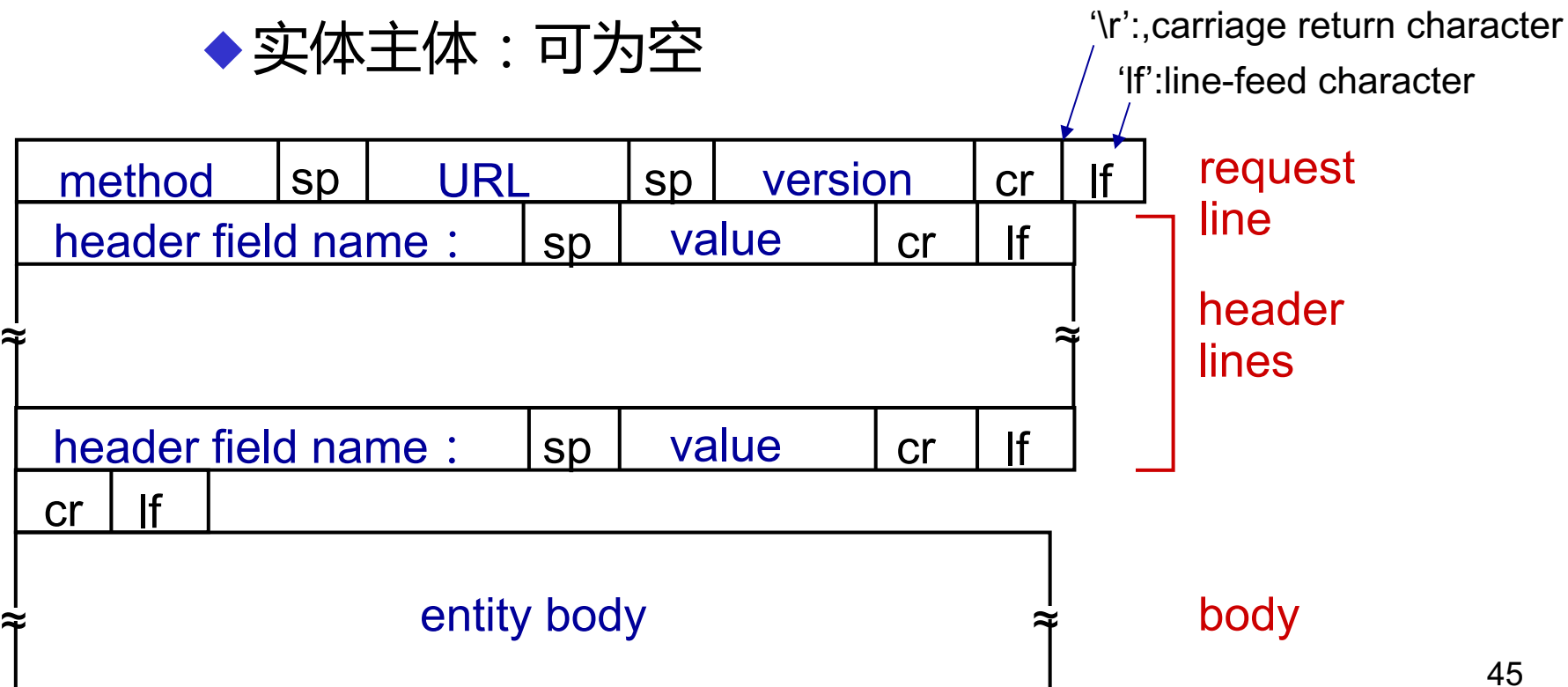
- 请求行 (GET, POST, HEAD命令)**: Points to the first line of the request: **GET /somedir/page.html HTTP/1.1**
- 首部行**: A bracket groups the following four lines: **Host: www.someschool.edu**, **Connection: close**, **User-agent: Mozilla/5.0**, and **Accept-language: fr**.
- 回车, 换行指示 报文的结束**: Points to the text **(回车, 换行)** at the end of the request.

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
(回车, 换行)
```

2.2 Web和HTTP : HTTP

➤ 通用格式

- ◆ 请求行：第1行，方法字段，URL字段，版本字段
- ◆ 首部行：一般多行
- ◆ 实体主体：可为空



2.2 Web和HTTP : HTTP

- GET : 最常用, 请求访问网页, 实体主体为空
- POST : 常用, 提交表单同时请求访问网页, 如使用搜索引擎, 实体主体中为表单输入值
- HEAD : 少用, 测试用, 与GET区别在于响应中去掉请求的对象
- PUT : 很少用, 向URL字段中定义的路径, 上载在实体主体中文件
- DELETE , 很少用, 删除在URL字段中定义的文件

2.2 Web和HTTP : HTTP

观察HTTP响应报文例子

状态行
(版本,状态码,状态短语)

首部行

数据, 如请求的
HTML文件

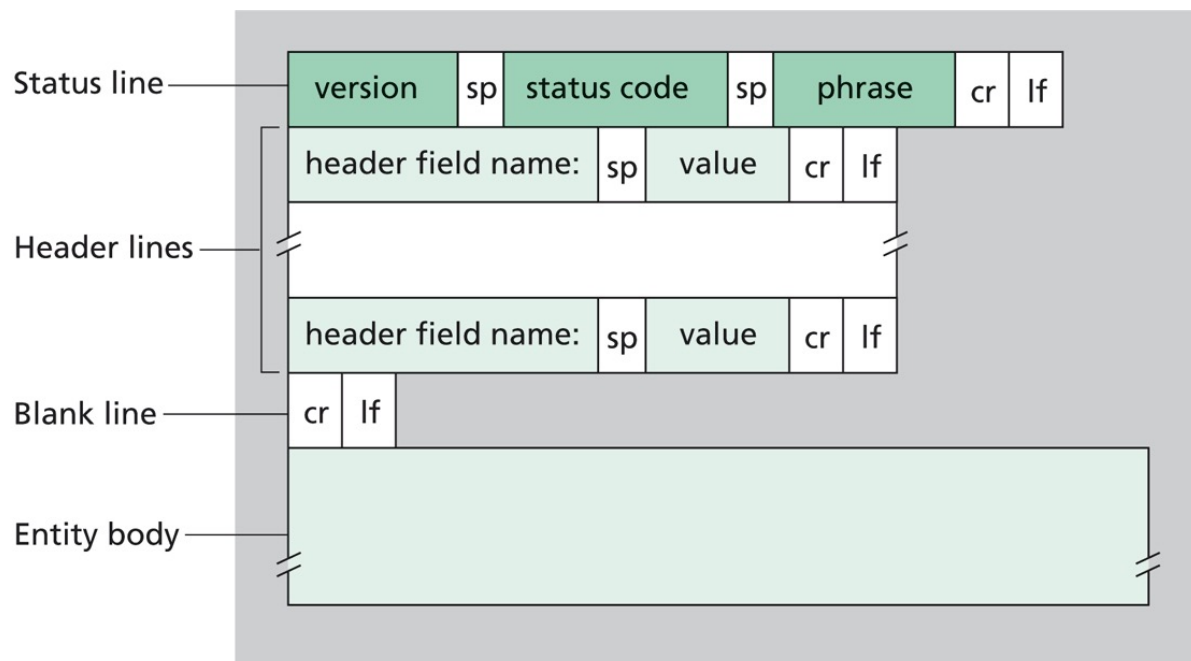
```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (Unix)
Last-Modified: Thu,02 Aug 2011 .....
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

2.2 Web和HTTP : HTTP

➤ 通用格式

- ◆ 状态行：第1行，版本字段，状态码字段，短语字段
- ◆ 首部行：一般多行
- ◆ 实体主体：报文的主体



2.2 Web和HTTP : HTTP

在服务器到客户机响应报文中的首行.

一些编码的例子:

200 OK

- ◆ 请求成功，请求的对象在这个报文后面

301 Moved Permanently

- ◆ 请求的对象已转移，新的URL在响应报文的Location:首部行中指定

400 Bad Request

- ◆ 请求报文不为服务器理解

404 Not Found

- ◆ 请求的文档没有在该服务器上发现

505 HTTP Version Not Supported

2.2 Web和HTTP : Cookie

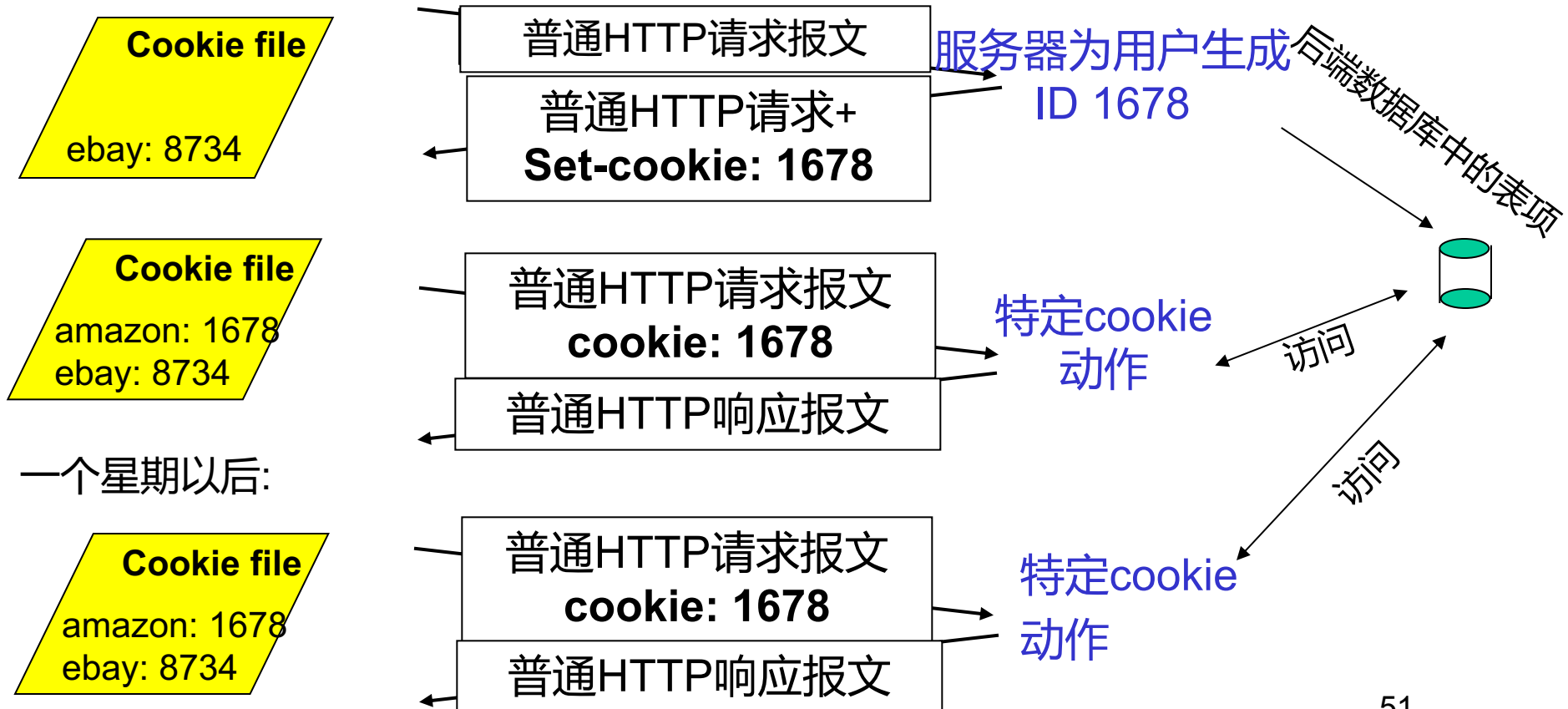
- HTTP是一种无状态协议，不能保存客户信息
- Cookie是一种在客户端保持HTTP状态信息的技术（好比：商场VIP卡）
 - ◆ 客户端访问网站时，Web服务器会查看、创建、修改Cookie资料
 - ◆ 帮助Web站点保存访问者信息：浏览历史，购物车
- cookie技术组成
 - ◆ cookie识别码（http请求报文，http响应报文）
 - ◆ 客户端保留cookie文件
 - ◆ 服务器提供后端数据库

2.2 Web和HTTP : Cookie

Susan访问Amazon网站

客户机

服务器



2.2 Web和HTTP : Cookie

Cookie的用途：

- 授权、认证
- 购物车
- 用户推荐
- 用户session状态

如何保持状态：

- 基于协议：在多个事物的发送/接收报文时维护状态信息
- cookie：通过http报文携带状态信息

Cookie：便利与隐私：

- 为用户提供便利
- 用户的个人信息泄露

2.2 Web和HTTP : HTTPs

HTTPS 协议（HyperText Transfer Protocol over Secure Socket Layer）：可以理解为HTTP+SSL/TLS，即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL，用于安全的 HTTP 数据传输。



2.2 Web和HTTP : HTTPs

- SSL (Secure Socket Layer , 安全套接字层) :
1994年为 Netscape 所研发 , SSL 协议位于 TCP/IP 协议与各种应用层协议之间 , 为数据通讯提供安全支持。
- TLS (Transport Layer Security , 传输层安全) : 其前身是 SSL , 它最初的几个版本 (SSL 1.0、SSL 2.0、SSL 3.0) 由网景公司开发 , 1999年从 3.1 开始被 IETF 标准化并改名 , 发展至今已经有 TLS 1.0、TLS 1.1、TLS 1.2 三个版本。SSL3.0和TLS1.0由于存在安全漏洞 , 已经很少被使用到。TLS 1.3 改动会比较大 , 目前还在草案阶段 , 目前使用最广泛的是 TLS 1.1、TLS 1.2。

2.2 Web和HTTP : HTTPs

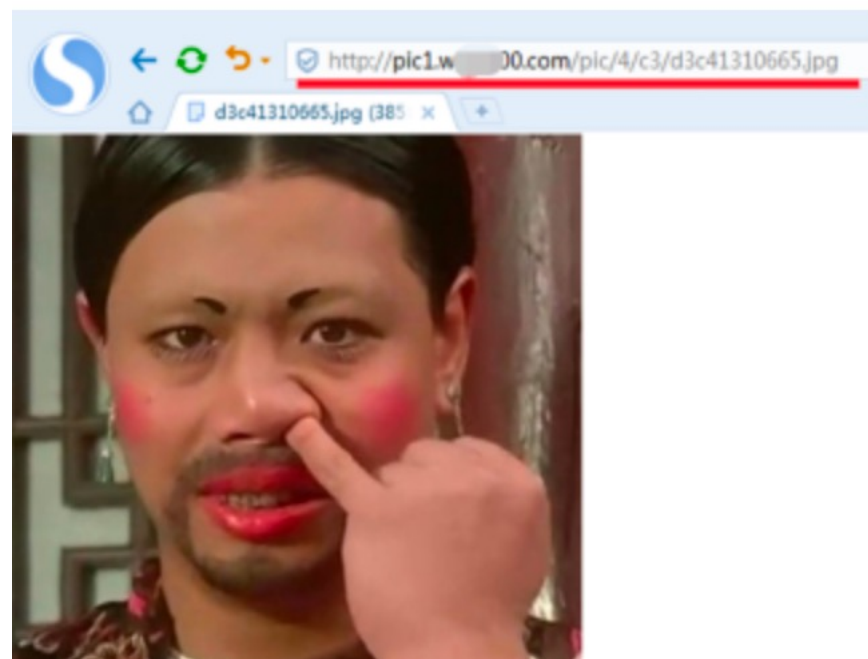
正在捕获 本地连接

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

ip.addr==183.15

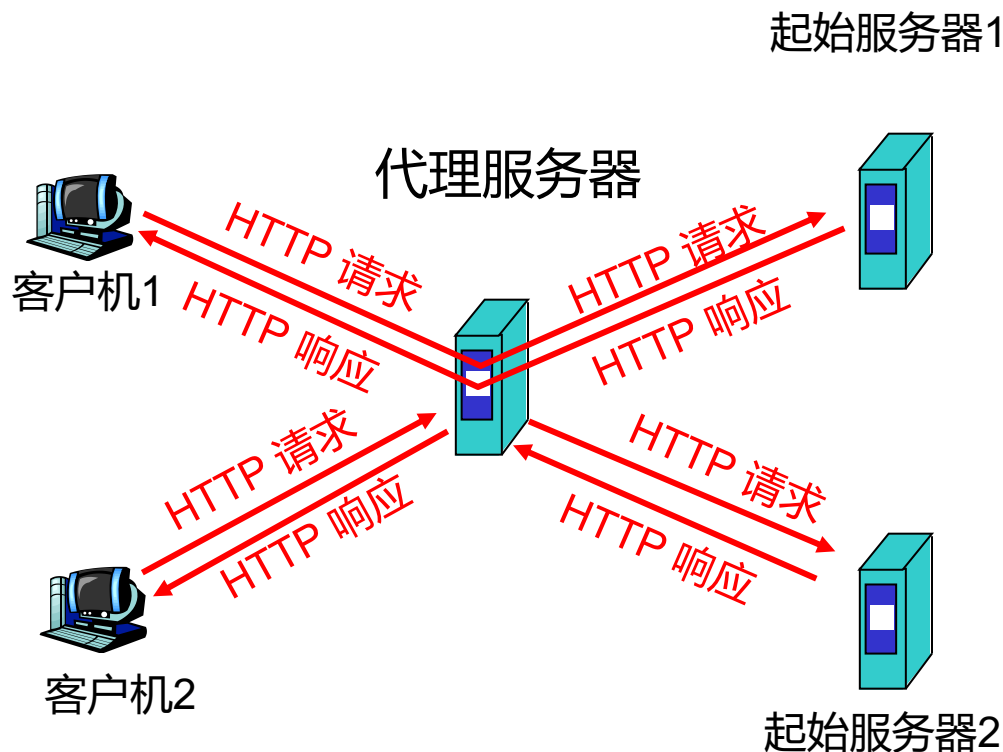
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|------------|-------------|----------|--------|---|
| 273 | 3.035865 | 10.0.2.137 | 183.158.7 | TCP | 74 | 53119→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM |
| 275 | 3.038219 | 183.158.3 | 10.0.2.137 | TCP | 74 | 80→53119 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SA |
| 276 | 3.038261 | 10.0.2.137 | 183.158.3 | TCP | 66 | 53119→80 [ACK] Seq=1 Ack=1 Win=66052 Len=0 TSval=2878487 TS |
| 277 | 3.039414 | 10.0.2.137 | 183.158.1 | HTTP | 493 | GET /pic/4/c3/d3c4 .jpg HTTP/1.1 客户端发送请求内容 |
| 278 | 3.041236 | 183.158.7 | 10.0.2.137 | TCP | 66 | 80→53119 [ACK] Seq=1 Ack=428 Win=30720 Len=0 TSval=17407877 |

TCP三次握手



2.2 Web和HTTP：Web缓存

- Web缓存器：保存最近请求过的web对象
- 浏览器向缓存发送所有HTTP请求
 - ◆ 对象在缓存中：缓存返回对象
 - ◆ 否则缓存向起始服务器请求对象，然后向客户机返回对象



好处：1) 减小客户机请求的响应时间；2) 减少机构内部网与因特网接入链路的通信量

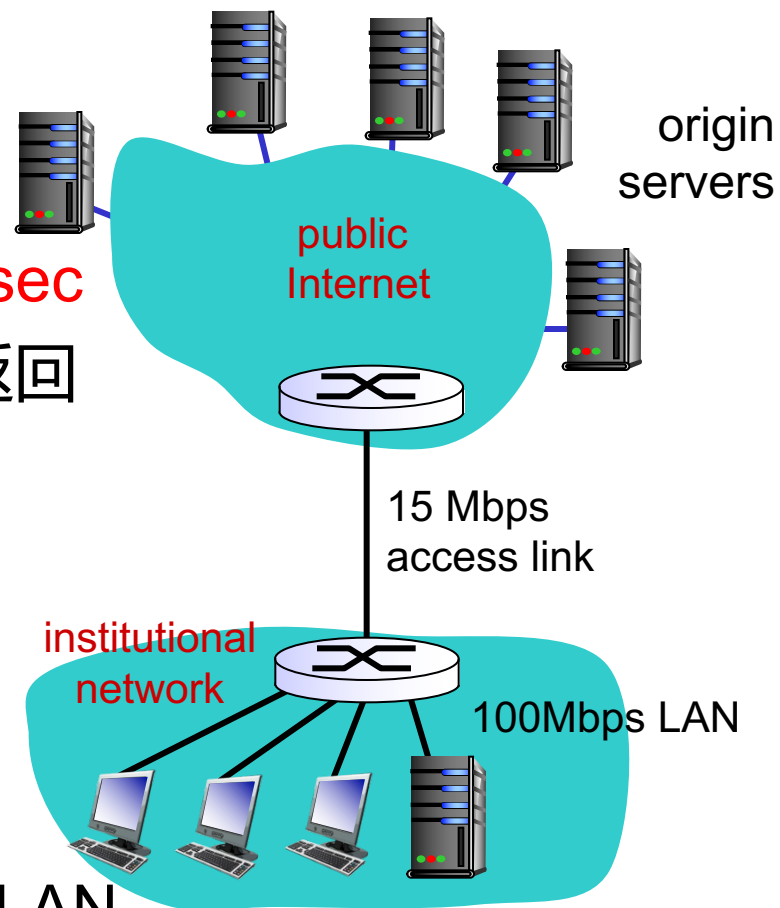
2.2 Web和HTTP：Web缓存

假定

- 平均对象长度 = 1Mb
- 来自机构的浏览器的平均请求 > 15/sec
- 从机构路由器到任何初始服务器并返回的时延 = 2 sec
- 接入网速率：15Mbps

结果

- 局域网的使用率：15%
- 接入链路使用率：100%
- 总时延 = 因特网时延 + 访问时延 + LAN时延 = 2 sec + 分钟 + 毫秒



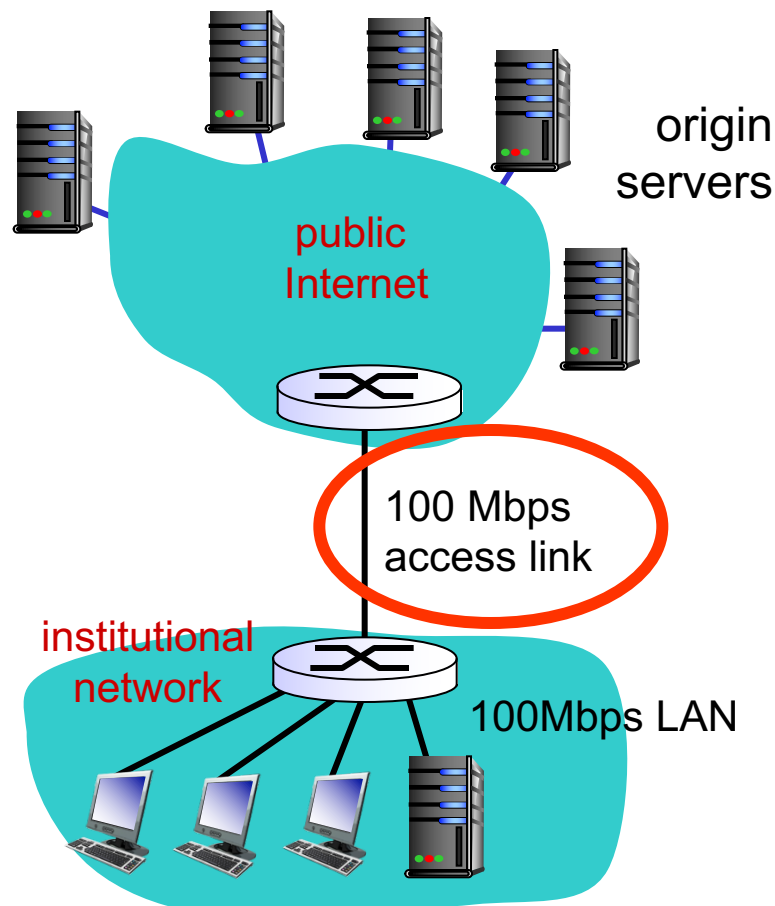
2.2 Web和HTTP : Web缓存

可能的解决方案

- 将访问链路的带宽提高到如100 Mbps

结果

- LAN利用率 = 15%
- 访问链路利用率 = 15%
- 总时延 = 因特网时延 + 访问时延 + LAN时延
= 2 sec + 毫秒 + 毫秒
- 通常升级费用可观



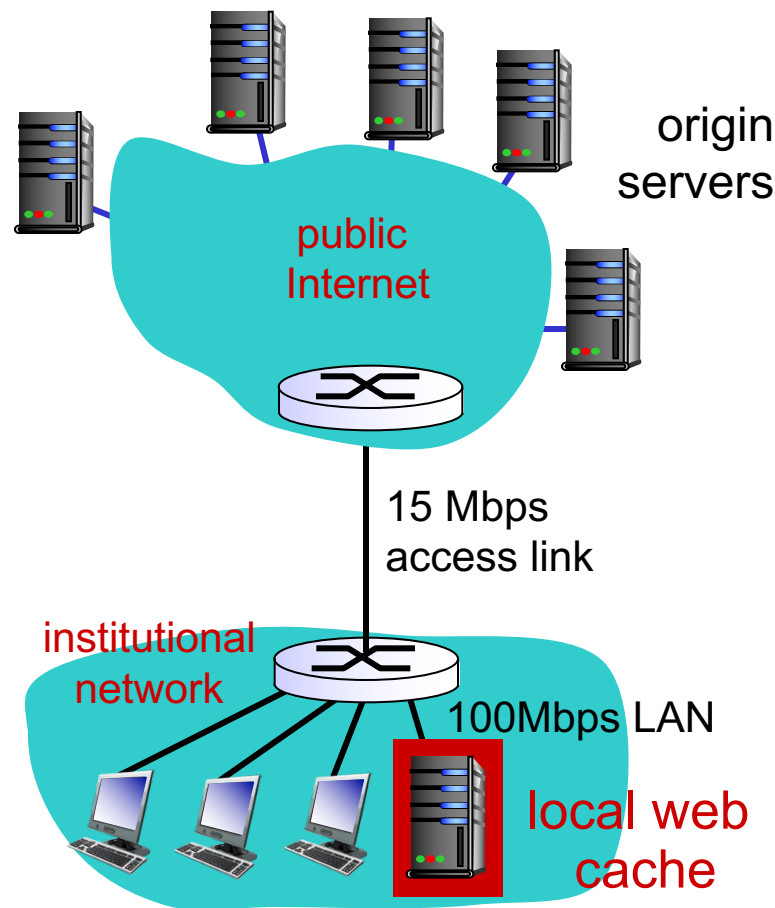
2.2 Web和HTTP : Web缓存

安装缓存器

- 假定命中率是0.4

结果

- 40%请求几乎立即得到满足
- 60%请求由初始服务器满足
- 接入链路的流量强度减少到60%, 产生可忽略不计的时延 (如10 msec)
- 总平均时延 = 因特网时延 + 访问时延 + LAN时延 =
 $0.6 * 2.01 \text{ secs} + 0.4 * 10 \text{ msec} \approx 1.2 \text{ secs}$



2.2 Web和HTTP : Web缓存

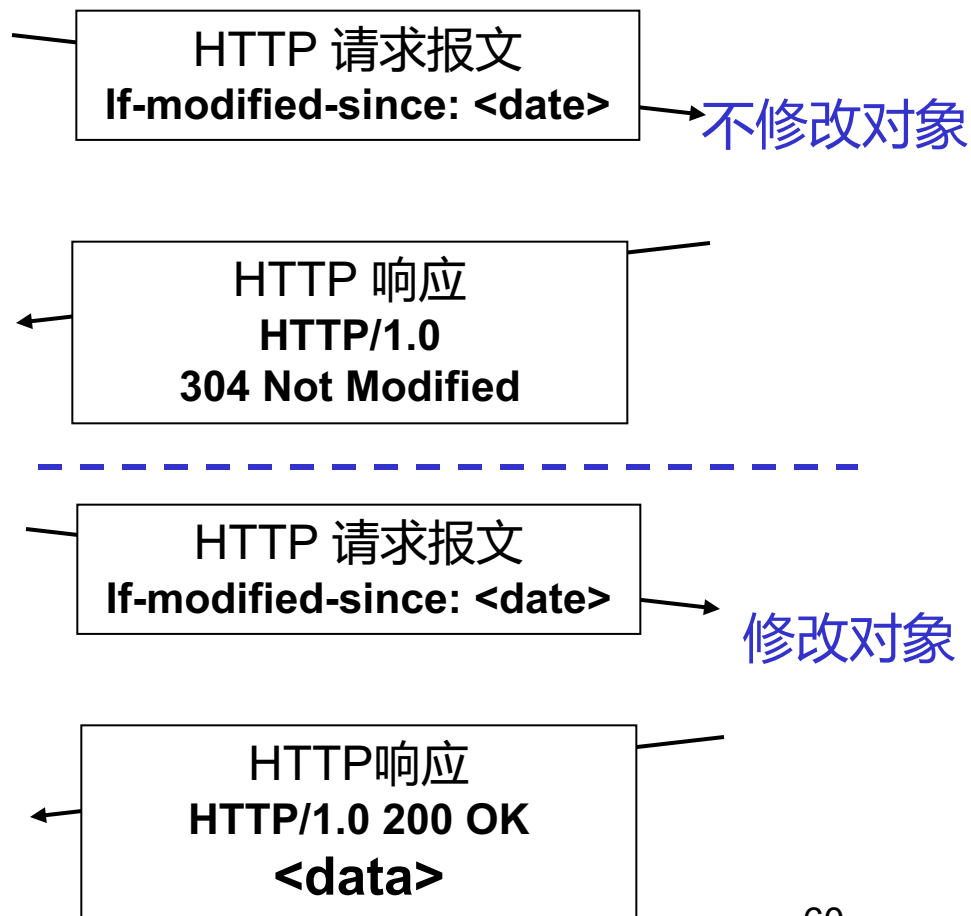
缓存中的对象可能不是最新？

解决：条件GET

- **目的**: 如果缓存中有最新缓存版本，就不发送该对象
- **缓存器**: 在HTTP请求**If-modified-since: <date>**中，指定缓存版本的日期
- **服务器**: 如果缓存的拷贝是最新，响应不包含对象:
HTTP/1.0 304 Not Modified

缓存器

服务器



2.2 Web和HTTP : Web缓存

If-Modified-Since

如果远端资源的 Last-Modified 首部标识的日期比在该首部中列出的值要更晚，表示条件匹配成功。

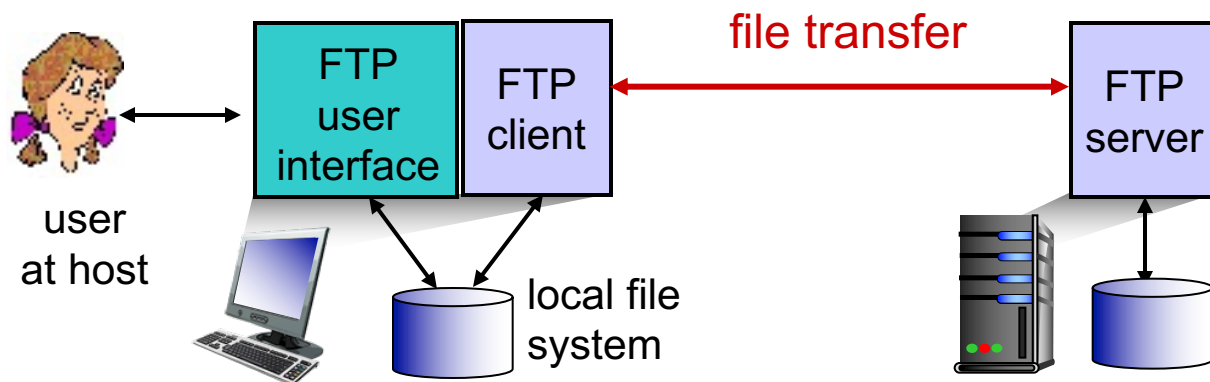
If-Unmodified-Since

如果远端资源的 HTTPHeader("Last-Modified")) 首部标识的日期比在该首部中列出的值要更早或相同，表示条件匹配成功。

- 2.1 应用层协议原理
- 2.2 Web应用和HTTP协议
- **2.3 文件传输协议：FTP**
- 2.4 电子邮件
- 2.5 域名系统DNS
- 2.6 P2P 应用

2.3 FTP工作机制

- 提供网络中的文件共享与传输
- FTP早于HTTP协议约10年，互联网还相对封闭



- 传输文件到/来自远程主机
- 客户机/服务器模型
 - ◆ 客户机: 发起传输的一侧(到/来自远程之一)
 - ◆ 服务器: 远程主机

2.3 FTP工作机制

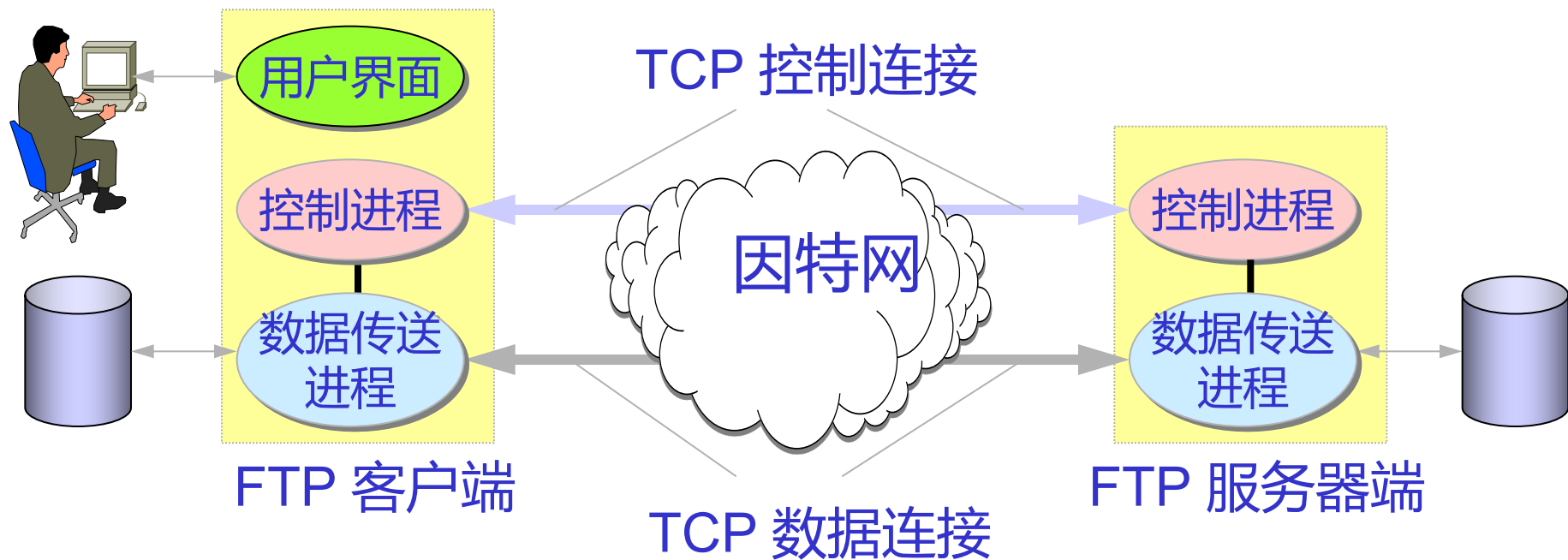
命令示例:

- 经控制信道以ASCII 文本发送
- **USER *username***
- **PASS *password***
- **LIST**返回当前目录中的文件列表
- **RETR filename**获取(get) 文件
- **STOR filename** 存储 (puts)文件到远程主机

返回码示例:

- 状态码和短语(如在HTTP中的那样)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

2.3 FTP工作机制



2.3 FTP工作机制

- 当客户进程向服务器进程发出建立连接请求时，要寻找连接服务器进程的控制端口(21)，同时还要告诉服务器进程自己的另一个端口号码，用于建立数据传送连接。
- 接着，服务器进程用自己传送数据的传输端口(20)与客户进程所提供的端口号码建立数据传送连接。
- 由于FTP使用了两个不同的端口号，所以数据连接与控制连接不会发生混乱。

2.3 FTP工作机制

- 使协议更加简单和更容易实现。
- 在传输文件时还可以利用控制连接（例如，客户发送请求终止传输）。

- 2.1 应用层协议原理
- 2.2 Web应用和HTTP协议
- 2.3 文件传输协议：FTP
- **2.4 电子邮件**
- 2.5 域名系统DNS
- 2.6 P2P 应用

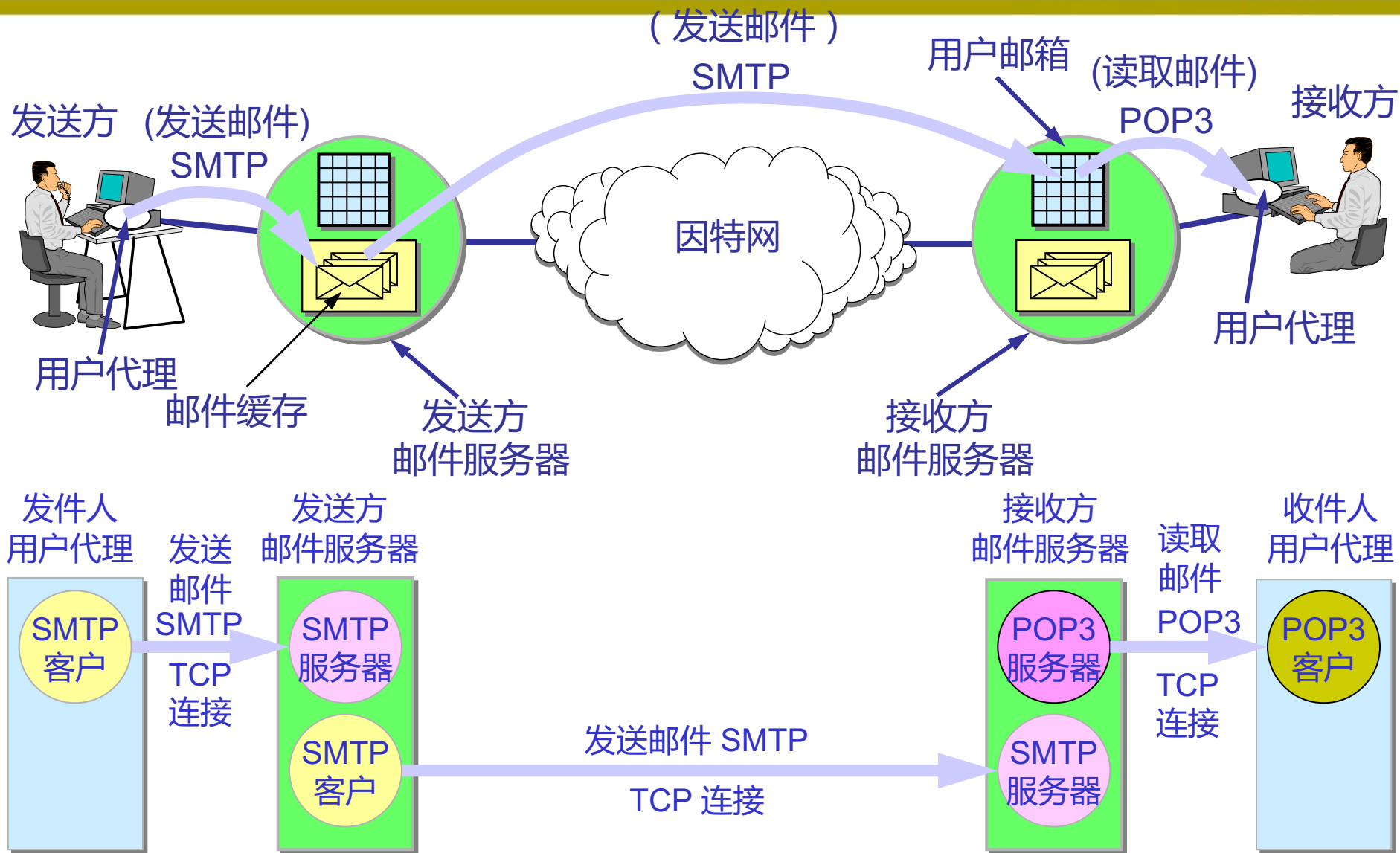
2.4 电子邮件协议族

- 电子邮件(e-mail)是因特网上使用得最多的和最受用户欢迎的一种应用。
- 电子邮件把邮件发送到收件人使用的邮件服务器，并放在其中的收件人邮箱中，收件人可随时上网到自己使用的邮件服务器进行读取。
- 电子邮件不仅使用方便，而且还具有传递迅速和费用低廉的优点。
- 现在电子邮件不仅可传送文字信息，而且还可附上声音和图像。

2.4 电子邮件协议族

- 发送邮件的协议：SMTP
- 读取邮件的协议：POP3 和 IMAP
- MIME 在其邮件首部中说明了邮件的数据类型(如文本、声音、图像、视像等)，使用 MIME 可在邮件中同时传送多种类型的数据。

2.4 电子邮件协议族：架构



2.4 电子邮件协议族：架构

- **用户代理** UA 就是用户与电子邮件系统的接口，是电子邮件客户端软件。
- 用户代理的功能：撰写、显示、处理和通信。
- 邮件服务器的功能是发送和接收邮件，同时还要向发信人报告邮件传送的情况（已交付、被拒绝、丢失等）。
- 邮件服务器按照客户-服务器方式工作。邮件服务器需要使用发送和读取两个不同的协议。

2.4 电子邮件协议族：架构

- 一个邮件服务器既可以作为客户，也可以作为服务器。
- 例如，当邮件服务器 A 向另一个邮件服务器 B 发送邮件时，邮件服务器 A 就作为 SMTP 客户，而 B 是 SMTP 服务器。
- 当邮件服务器 A 从另一个邮件服务器 B 接收邮件时，邮件服务器 A 就作为 SMTP 服务器，而 B 是 SMTP 客户。

2.4 电子邮件协议族：架构

- ① 发件人调用 PC 中的用户代理撰写和编辑要发送的邮件。
- ② 发件人的用户代理把邮件用 SMTP 协议发给发送方邮件服务器，
- ③ SMTP 服务器把邮件临时存放在邮件缓存队列中，等待发送。
- ④ 发送方邮件服务器的 SMTP 客户与接收方邮件服务器的 SMTP 服务器建立 TCP 连接，然后就把邮件缓存队列中的邮件依次发送出去。

2.4 电子邮件协议族：架构

- ⑤ 运行在接收方邮件服务器中的SMTP服务器进程收到邮件后，把邮件放入收件人的用户邮箱中，等待收件人进行读取。
- ⑥ 收件人在打算收信时，就运行 PC 机中的用户代理，使用 POP3（或 IMAP）协议读取发送给自己的邮件。
- 请注意，POP3 服务器和 POP3 客户之间的通信是由 POP3 客户发起的。

2.4 电子邮件协议族：SMTP

1. 连接建立：连接是在发送主机的 SMTP 客户和接收主机的 SMTP 服务器之间建立的。SMTP不使用中间的邮件服务器。
2. 邮件传送
3. 连接释放：邮件发送完毕后，SMTP 应释放 TCP 连接。

2.4 电子邮件协议族：SMTP

- RFC 822 只规定了邮件内容中的首部(header)格式，而对邮件的主体(body)部分则让用户自由撰写。
- 用户写好首部后，邮件系统将自动地将信封所需的信息提取出来并写在信封上。所以用户不需要填写电子邮件信封上的信息。
- 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：To 和 Subject。

2.4 电子邮件协议族：SMTP

- “To:”后面填入一个或多个收件人的电子邮件地址。用户只需打开地址簿，点击收件人名字，收件人的电子邮件地址就会自动地填入到合适的位置上。
- “Subject:”是邮件的主题。它反映了邮件的主要内容，便于用户查找邮件。
- 抄送 “Cc:” 表示应给某某人发送一个邮件副本。
- “From” 和 “Date” 表示发信人的电子邮件地址和发信日期。“Reply-To” 是对方回信所用的地址。

2.4 电子邮件协议族：POP

- 邮局协议 POP 是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 POP3。
- POP 也使用客户服务器的工作方式。
- 在接收邮件的用户 PC 机中必须运行 POP 客户程序，而在用户所连接的 ISP 的邮件服务器中则运行 POP 服务器程序。

2.4 电子邮件协议族：IMAP

- IMAP 也是按客户服务器方式工作，现在较新的是版本 4，即 IMAP4。
- 用户在自己的 PC 机上就可以操纵 ISP 的邮件服务器的邮箱，就像在本地操纵一样。
- 因此 IMAP 是一个联机协议。当用户 PC 机上的 IMAP 客户程序打开 IMAP 服务器的邮箱时，用户就可看到邮件的首部。若用户需要打开某个邮件，则该邮件才传到用户的计算机上。

2.4 电子邮件协议族：IMAP

- IMAP最大的好处就是用户可以在不同的地方使用不同的计算机随时上网阅读和处理自己的邮件。
- IMAP 还允许收件人只读取邮件中的某一个部分。例如，收到了一个带有视像附件（此文件可能很大）的邮件。为了节省时间，可以先下载邮件的正文部分，待以后有时间再读取或下载这个很长的附件。
- IMAP 的缺点是如果用户没有将邮件复制到自己的PC上，则邮件一直是存放在IMAP服务器上。因此用户需要经常与IMAP服务器建立连接。

2.4 电子邮件协议族：IMAP

- 不要将邮件读取协议 POP 或 IMAP 与邮件传送协议 SMTP 弄混。
- 发信人的用户代理向源邮件服务器发送邮件，以及源邮件服务器向目的邮件服务器发送邮件，都是使用 SMTP 协议。
- 而 POP 协议或 IMAP 协议则是用户从目的邮件服务器上读取邮件所使用的协议。

2.4 电子邮件协议族：IMAP

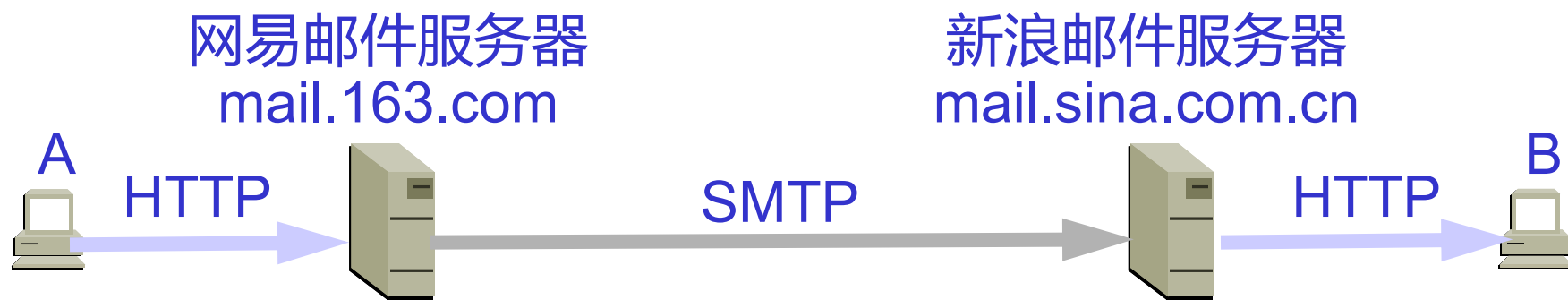
- **POP3** 协议允许电子邮件客户端下载服务器上的邮件，但是在客户端的操作（如移动邮件、标记已读等），不会反馈到服务器上。
- **IMAP** 客户端的操作都会反馈到服务器上，对邮件进行的操作，服务器上的邮件也会做相应的动作
- **IMAP** 提供摘要浏览功能，可以让你在阅读完所有的邮件到达时间、主题、发件人、大小等信息后才作出是否下载的决定。此外，**IMAP** 更好地支持了从多个不同设备中随时访问新邮件。

2.4 电子邮件协议族：IMAP

| 操作位置 | 操作内容 | IMAP | POP3 |
|-------|----------------|------------|-------|
| 收件箱 | 阅读、标记、移动、删除邮件等 | 客户端与邮箱更新同步 | 仅客户端内 |
| 发件箱 | 保存到已发送 | 客户端与邮箱更新同步 | 仅客户端内 |
| 创建文件夹 | 新建自定义的文件夹 | 客户端与邮箱更新同步 | 仅客户端内 |
| 草稿 | 保存草稿 | 客户端与邮箱更新同步 | 仅客户端内 |
| 垃圾文件夹 | 接收误移入垃圾文件夹的邮件 | 支持 | 不支持 |
| 广告邮件 | 接收被移入广告邮件夹的邮件 | 支持 | 不支持 |

2.4 电子邮件协议族：基于万维网

- 电子邮件从 A 发送到网易邮件服务器是使用 HTTP 协议。
- 两个邮件服务器之间的传送使用 SMTP。
- 邮件从新浪邮件服务器传送到 B 是使用 HTTP 协议。



2.4 电子邮件协议族：MIME

SMTP 有以下缺点：

- SMTP 不能传送可执行文件或其他的二进制对象。
- SMTP 限于传送 7 位的 ASCII 码。许多其他非英语国家的文字（如中文、俄文，甚至带重音符号的法文或德文）就无法传送。
- SMTP 服务器会拒绝超过一定长度的邮件。
- 某些 SMTP 的实现并没有完全按照[RFC 821]的 SMTP 标准。

2.4 电子邮件协议族：MIME

- MIME 并没有改动 SMTP 或取代它。
- MIME 的意图是继续使用目前的[RFC 822]格式，但增加了邮件主体的结构，并定义了传送非 ASCII 码的编码规则。



- 2.1 应用层协议原理
- 2.2 Web应用和HTTP协议
- 2.3 文件传输协议：FTP
- 2.4 电子邮件
- **2.5 域名系统DNS**
- 2.6 P2P 应用

DNS: 域名系统

➤ DNS: Domain Name System 域名系统

- ◆ 同时: Domain Name Server 域名服务器, DNS 协议
- ◆ 分布式数据库, 提供名字服务

人: 许多标识符

- ◆ 名字, 身份证号, 学号, 护照号

因特网主机、路由器:

- ◆ 主机名: www.nwpu.edu.cn
- ◆ IP地址(32 bit) : 113.107.238.154

问题: IP地址和名字之间的映射?

DNS完成主机名
到IP地址的解析

DNS服务

- 主机名到IP地址的转换

- 主机别名：

 - n主机名→1 IP

例如：邮件服务器别名Foxmail.qq.com

为何不用集中式DNS?

- 单点故障

- 通信量

- 远距离集中式数据库

- 维护

2.5 域名系统

- 许多应用层软件经常直接使用**域名系统** DNS (Domain Name System)，但计算机的用户只是间接而不是直接使用域名系统。
- 因特网采用层次结构的命名树作为主机的名字，并使用**分布式的**域名系统 DNS。
- 名字到 IP 地址的解析是由若干个域名服务器程序完成的。域名服务器程序在专设的结点上运行，运行该程序的机器称为**域名服务器**。

2.5 域名系统：结构

- 因特网采用了层次树状结构的命名方法。
- 任何一个连接在因特网上的主机或路由器，都有一个**唯一**的层次结构的**名字**，即**域名**。
- 域名的结构由标号序列组成，各标号之间用**点**隔开：

... . 三级域名 . 二级域名 . 顶级域名

- 各标号分别代表不同级别的域名。

2.5 域名系统：结构

- 域名只是个逻辑概念，并不代表计算机所在的物理地点。
- 变长的域名和使用有助记忆的字符串，是为了便于人来使用。而 IP 地址是定长的 32 位二进制数字则非常便于机器进行处理。
- 域名中的“点”和点分十进制 IP 地址中的“点”并无一一对应的关系。点分十进制 IP 地址中一定是包含三个“点”，但每一个域名中“点”的数目则不一定正好是三个。

2.5 域名系统：层次

- (1) 国家顶级域名 nTLD：如: .cn 表示中国，.us 表示美国，.uk 表示英国，等等。
- (2) 通用顶级域名 gTLD：最早的顶级域名是：
 - .com （ 公司和企业 ）
 - .net （ 网络服务机构 ）
 - .org （ 非赢利性组织 ）
 - .edu （ 美国专用的教育机构 ）
 - .gov （ 美国专用的政府部门 ）
 - .mil （ 美国专用的军事部门 ）
 - .int （ 国际组织 ）

2.5 域名系统：层次

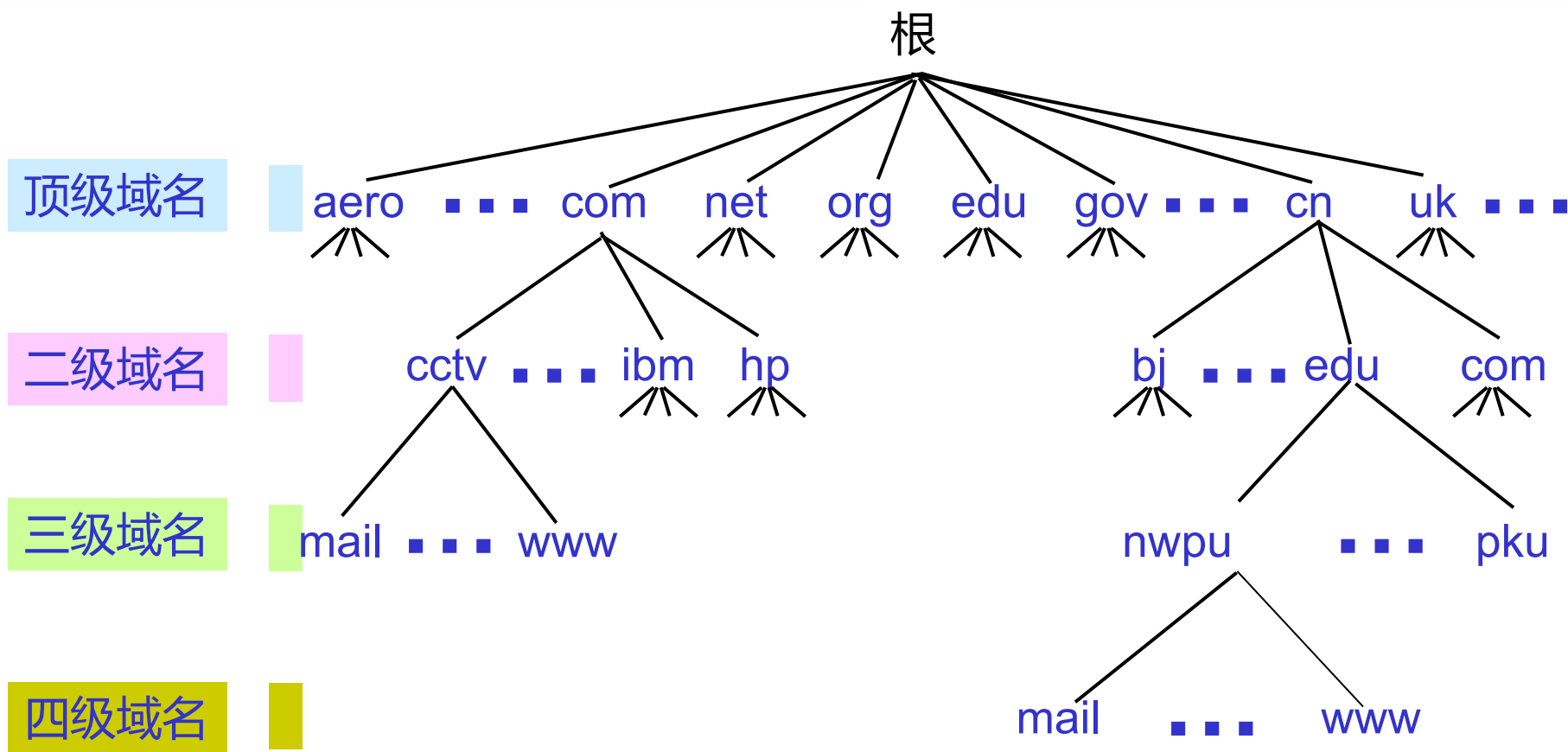
(3) 基础结构域名(infrastructure domain)：这种顶级域名只有一个，即 arpa，用于反向域名解析，因此又称为反向域名。

2.5 域名系统：层次

新增顶级域名

- .aero （ 航空运输企业 ）
- .biz （ 公司和企业 ）
- .cat （ 加泰隆人的语言和文化团体 ）
- .coop （ 合作团体 ）
- .info （ 各种情况 ）
- .jobs （ 人力资源管理者 ）
- .mobi （ 移动产品与服务的用户和提供者 ）
- .museum （ 博物馆 ）
- .name （ 个人 ）
- .pro （ 有证书的专业人员 ）
- .travel （ 旅游业 ）

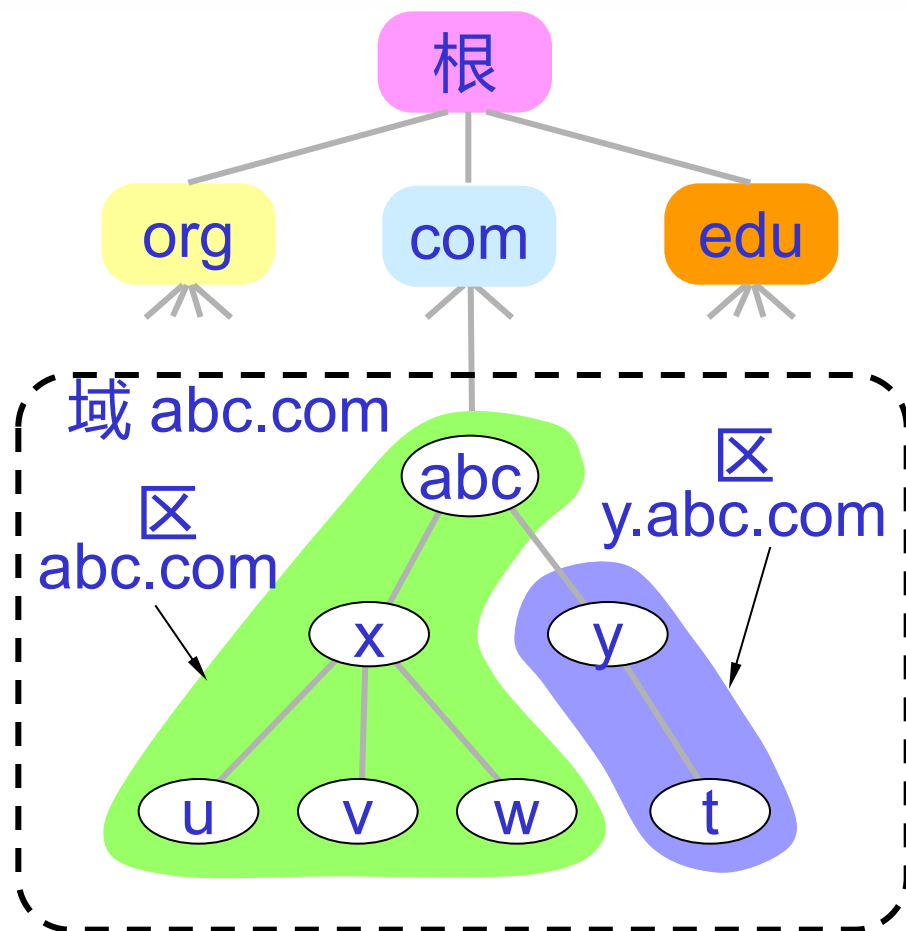
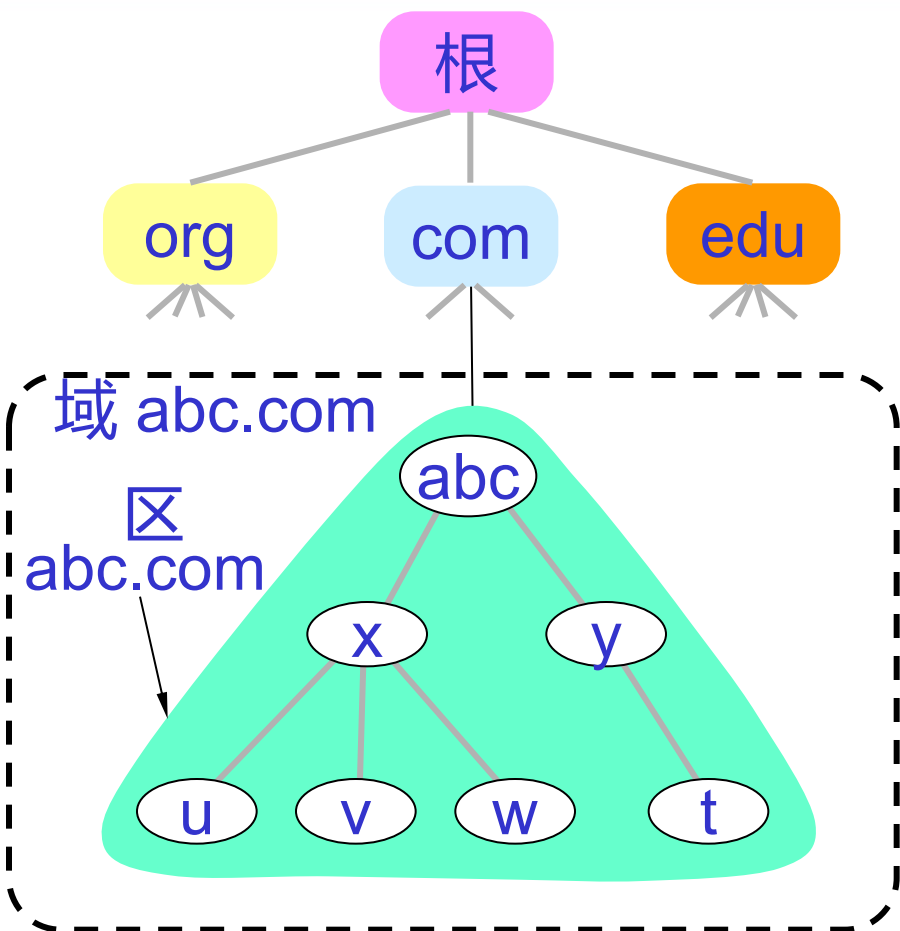
2.5 域名系统：层次



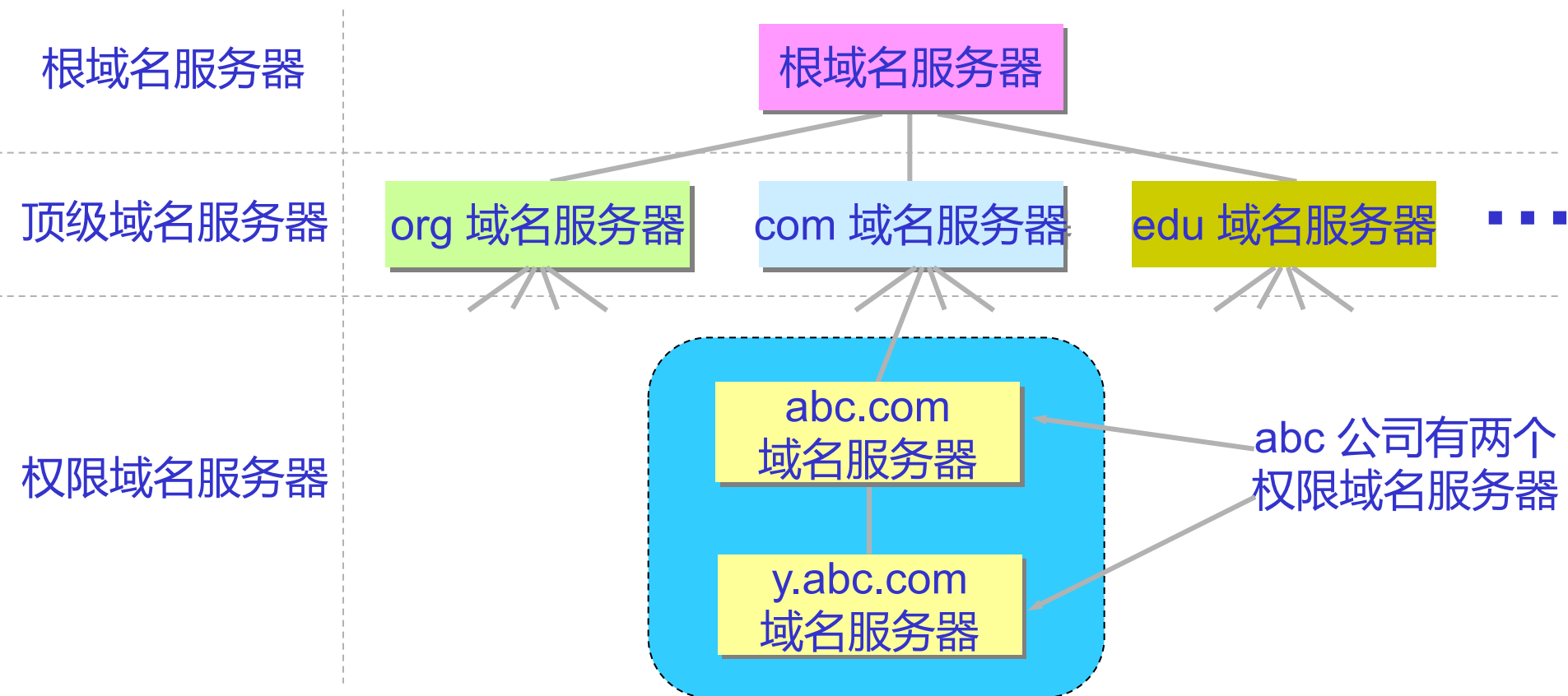
2.5 域名系统：域名服务器

- 一个服务器所负责管辖的（或有权限的）范围叫做区(zone)。
- 各单位根据具体情况来划分自己管辖范围的区。但在一个区中的所有节点必须是能够连通的。
- 每一个区设置相应的权限域名服务器，用来保存该区中的所有主机的域名到IP地址的映射。
- DNS 服务器的管辖范围不是以“域”为单位，而是以“区”为单位。

2.5 域名系统：域名服务器



2.5 域名系统：域名服务器



2.5 域名系统：域名服务器

- 根域名服务器
- 顶级域名服务器
- 权限域名服务器
- 本地域名服务器

2.5 域名系统：根域名服务器

- **根域名服务器**是最重要的域名服务器。所有的根域名服务器都知道所有的顶级域名服务器的域名和 IP 地址。
- 不管是哪一个本地域名服务器，若要对因特网上任何一个域名进行解析，只要自己无法解析，就首先求助于根域名服务器。
- 在因特网上共有13 个不同 IP 地址的根域名服务器，它们的名字是用一个英文字母命名，从 a 一直到 m（前13 个字母）。

2.5 域名系统：根域名服务器

- 这些根域名服务器相应的域名分别是
 - a.rootservers.net
 - b.rootservers.net
 - ...
 - m.rootservers.net
- 到 2006 年底全世界已经安装了一百多个根域名服务器机器，分布在世界各地。
- 这样做的目的是为了更方便用户，使世界上大部分 DNS 域名服务器都能就近找到一个根域名服务器。

2.5 域名系统：根域名服务器



根域名服务器 f 共有 49 个机器

- 根域名服务器并不直接把域名直接转换成 IP 地址。
- 在使用迭代查询时，根域名服务器把下一步应当找的顶级域名服务器的 IP 地址告诉本地域名服务器。

2.5 域名系统：顶级域名服务器

- 这些域名服务器负责管理在该顶级域名服务器注册的所有二级域名。
- 当收到 DNS 查询请求时，就给出相应的回答（可能是最后的结果，也可能是下一步应当找的域名服务器的 IP 地址）。

2.5 域名系统：权限域名服务器

- 这就是前面已经讲过的负责一个区的域名服务器。
- 当一个权限域名服务器还不能给出最后的查询回答时，就会告诉发出查询请求的 DNS 客户，下一步应当找哪一个权限域名服务器。

2.5 域名系统：本地域名服务器

- 本地域名服务器对域名系统非常重要。
- 当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器。
- 每一个因特网服务提供者 ISP，或一个大学，甚至一个大学里的系，都可以拥有一个本地域名服务器，
- 这种域名服务器有时也称为默认域名服务器。

2.5 域名系统：辅助域名服务器

- DNS 域名服务器都把数据复制到几个域名服务器来保存，其中的一个是主域名服务器，其他的是辅助域名服务器。
- 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断。
- 主域名服务器定期把数据复制到辅助域名服务器中，而更改数据只能在主域名服务器中进行。这样就保证了数据的一致性。

2.5 域名系统：域名查询

- 主机向本地域名服务器的查询一般都是采用递归查询。如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。
- 本地域名服务器向根域名服务器的查询通常是采用迭代查询。当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。

2.5 域名系统：域名查询

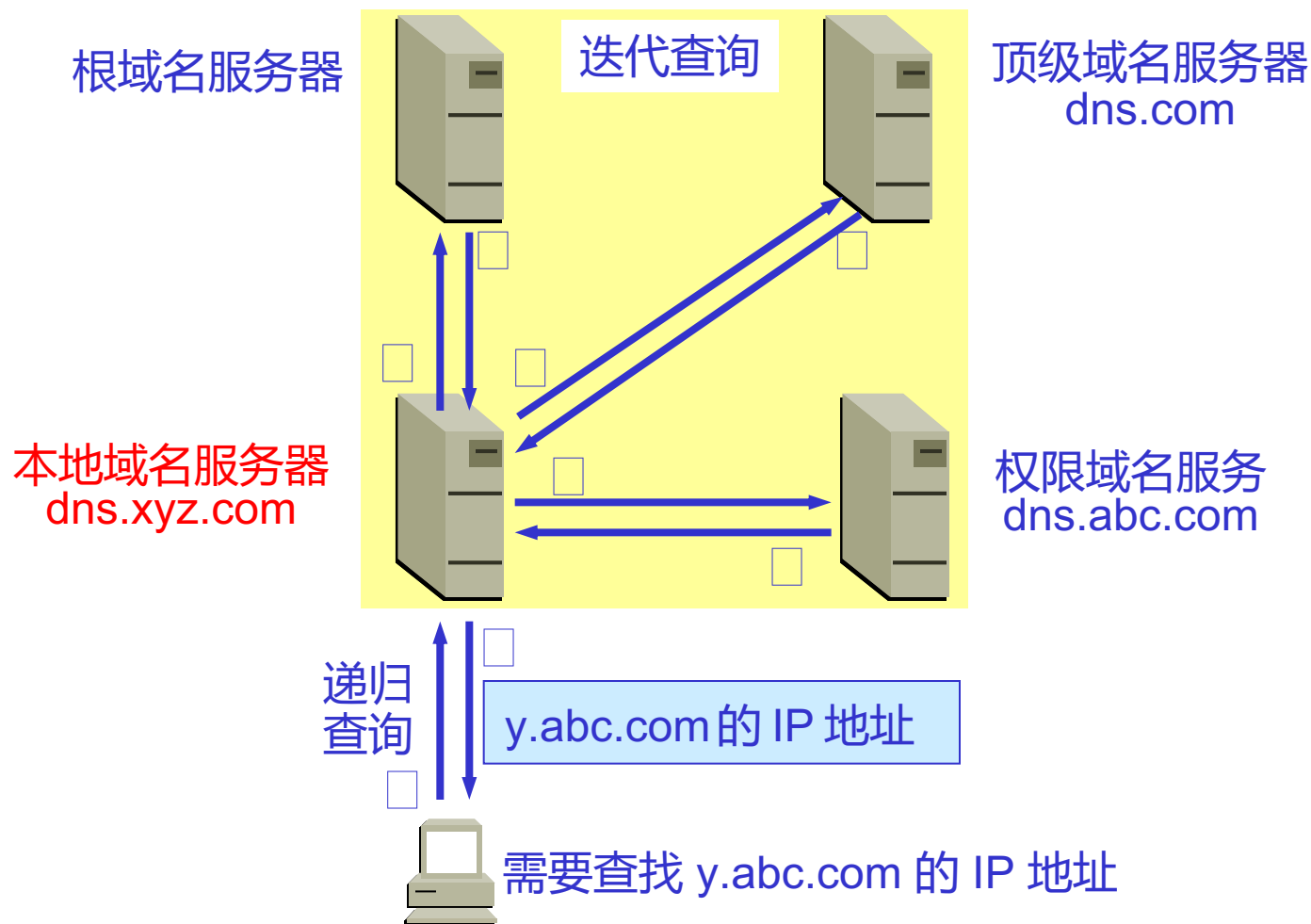
➤ 递归查询

在该模式下DNS 服务器接收到客户机请求，必须使用一个准确的查询结果回复客户机。如果DNS 服务器本地没有存储查询DNS 信息，那么该服务器会询问其他服务器，并将返回的查询结果提交给客户机。

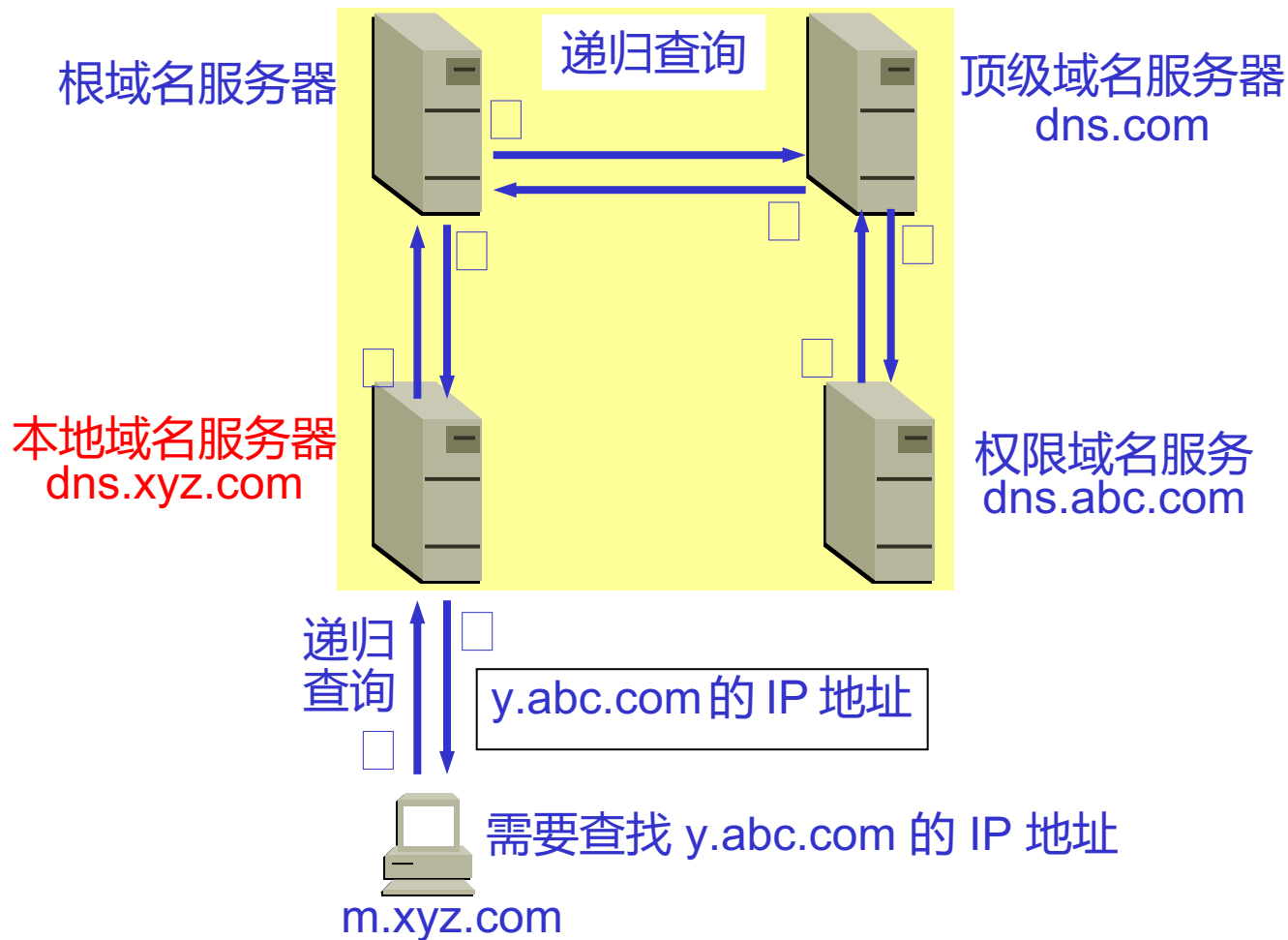
➤ 迭代查询

DNS 服务器会向客户机提供其他能够解析查询请求的DNS 服务器地址，当客户机发送查询请求时，DNS 服务器并不直接回复查询结果，而是告诉客户机另一台DNS 服务器地址，客户机再向这台DNS 服务器提交请求，依次循环直到返回查询的结果为止。

2.5 域名系统：域名查询



2.5 域名系统：域名查询



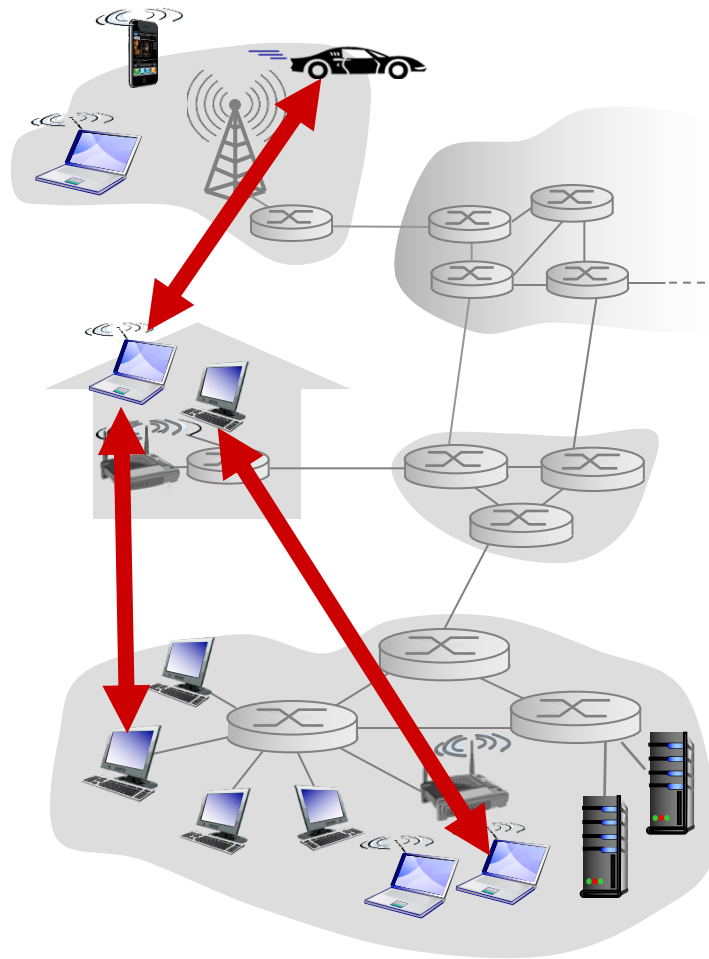
2.6 分布式文件共享：P2P

➤ P2P体系结构

- 没有始终运行着的服务器
- 任意的端系统（称之为对等方）之间直接通信
- 对等方之间间歇性的互联，其IP地址也可能改变

➤ 例如：

- 文件分发
- 流媒体
- VoIP



2.6 分布式文件共享：P2P

- 自从因特网能够提供音频/视频服务后，宽带上网用户数也急剧增长。很多用户使用宽带接入的目的就是为了更快地下载音频/视频文件。
- P2P 工作方式受到广大网民的欢迎。这种工作方式解决了集中式媒体服务器可能出现的瓶颈问题。
- 在 P2P 工作方式下，所有的音频/视频文件都是在普通的因特网用户之间传输。这是相当于有很多分散在各地的媒体服务器向其他用户提供所要下载的音频/视频文件。

2.6 分布式文件共享：P2P

Napster

- 最早出现的 P2P 技术，可提供免费下载 MP3 音乐。
- Napster 能够搜索音乐文件，能够提供检索功能。所有的[音乐文件地址](#)集中存放在一个 Napster 目录服务器中。使用者可很方便地下载需要的 MP3 文件。
- 用户要及时向 Napster 的目录服务器报告自己存有的音乐文件。当用户想下载某个 MP3 文件时，就向目录服务器发出询问。目录服务器检索出结果后向用户返回存放此文件的 PC 机的 IP 地址。Napster 的文件传输是分散的，但文件的定位则是集中的。
- 这种集中式目录服务器的缺点就是[可靠性差](#)。Napster 被判决属于“间接侵害版权”，因此在 2000 年 7 月底 Napster 网站就被迫关闭了。

2.6 分布式文件共享：P2P

Gnutella

- Gnutella 是第二代 P2P 文件共享程序，它全分布方法定位内容的P2P 文件共享应用程序。。
- Gnutella 与 Napster 最大的区别就是不使用集中式的目录服务器，而是使用洪泛法在大量 Gnutella 用户之间进行查询。
- 为了不使查询的通信量过大，Gnutella 设计了一种有限范围的洪泛查询。这样可以减少倾注到因特网的查询流量，但由于查询的范围受限，因而这也影响到查询定位的准确性。

2.6 分布式文件共享：P2P

电骡 eMule

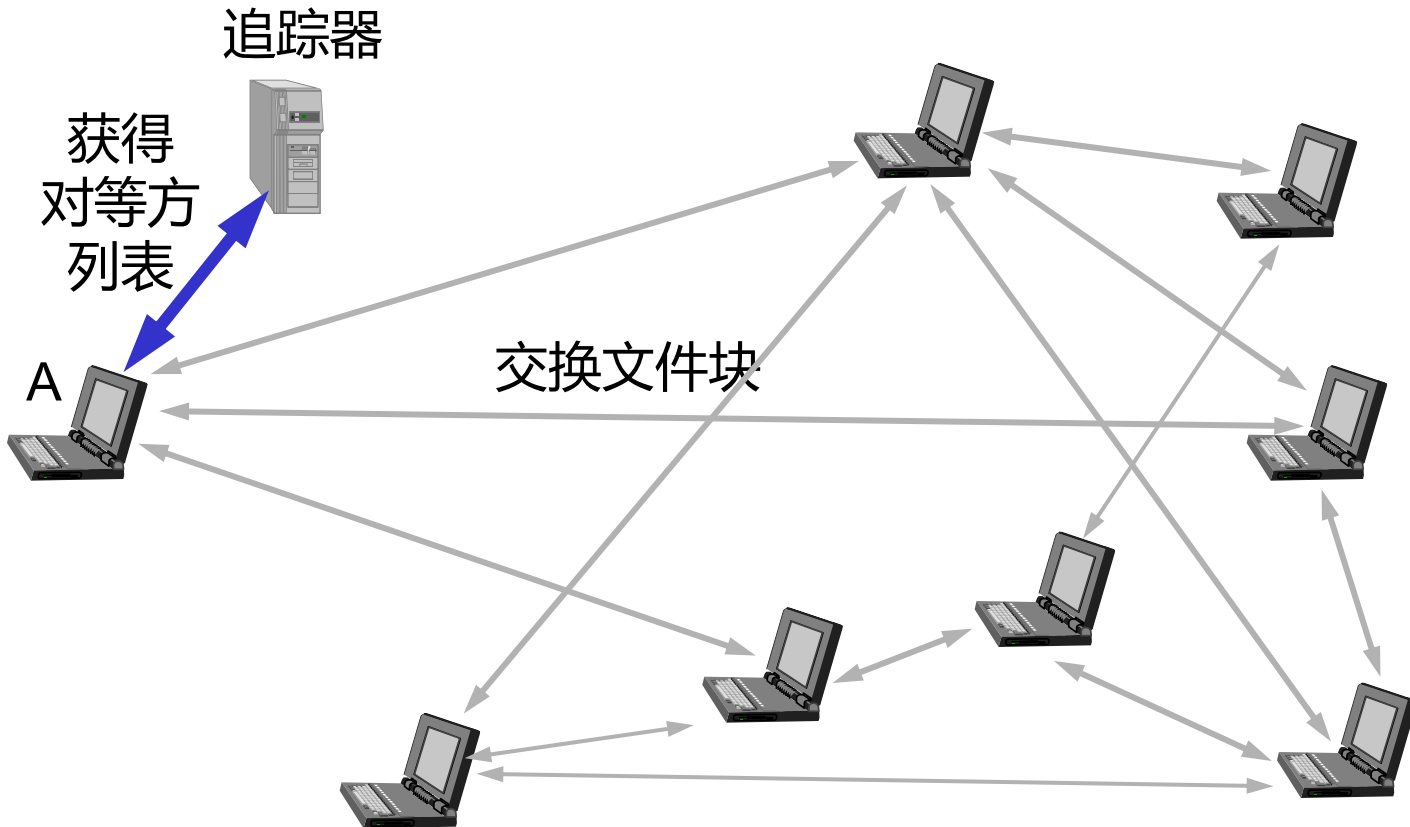
- 电骡 eMule 使用分散定位和分散传输技术，把每一个文件划分为许多小文件块，并使用多源文件传输协议 MFTP 进行传送。因此用户可以同时从很多地方下载一个文件中的不同文件块。由于每一个文件块都很小，并且是并行下载，所以下载可以比较快地完成。
- eMule 用户在下载文件的同时，也在上传文件，因此，因特网上成千上万的 eMule 用户在同时下载和上传一个个小的文件块。

2.6 分布式文件共享：P2P

eMule 的其他特点

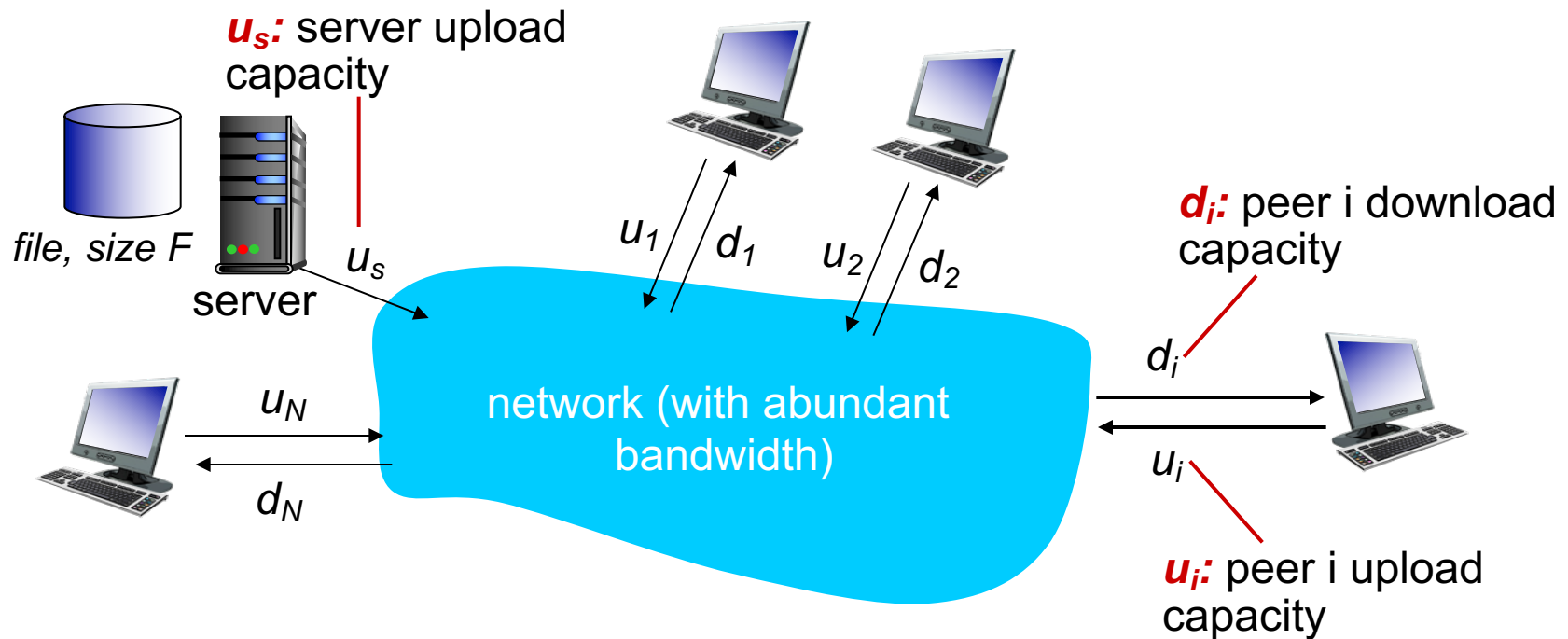
- eMule 使用了一些服务器。这些服务器并不是保存音频/视频文件，而是保存用户的有关信息，因而可以告诉用户从哪些地方可以下载到所需的文件。
- eMule 使用了专门定义的文件夹，让用户存放可以和其他用户共享的文件。
- eMule 的下载文件规则是鼓励用户向其他用户上传文件。用户上传文件越多，其下载文件的优先级就越高（因而下载就越快）。

2.6 分布式文件共享：P2P



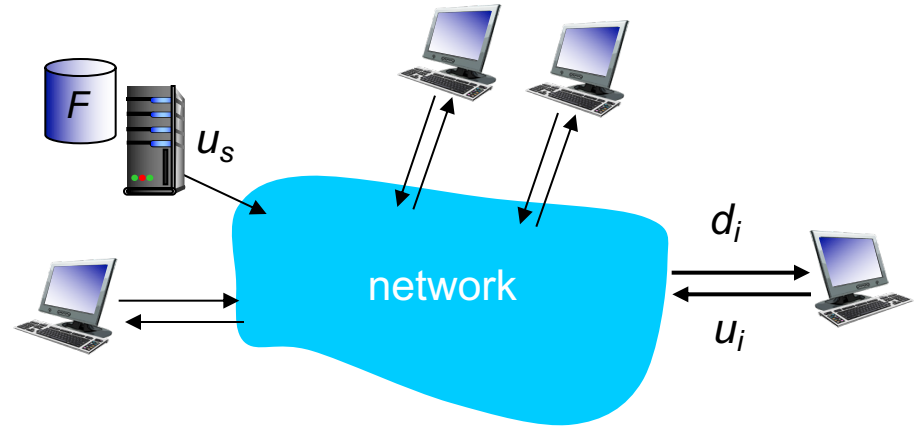
2.6 分布式文件共享：P2P

- Q：将文件（大小为 F ）从服务器分发到 N 个节点，需要多少时间？
- 每个节点上传/下载的数据容量是有限的



2.6 分布式文件共享：P2P

- 文件分发时间：客户-服务器模式
- 服务器处理：必须顺序的发送（上传） N 个拷贝： NF/u_s
- 客户端：每个客户端下载文件拷贝： F/d_{\min}



*time to distribute F
to N clients using
client-server approach*

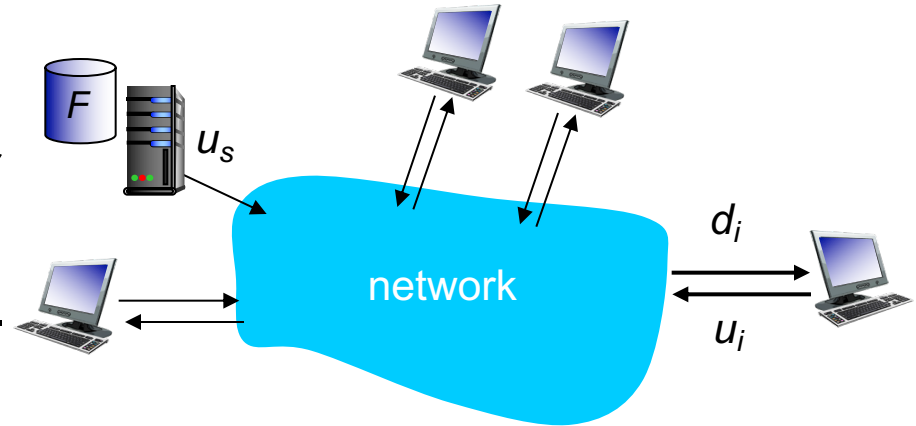
$$D_{c-s} \geq \max\{NF/u_s, F/d_{\min}\}$$

increases linearly in N

2.6 分布式文件共享：P2P

文件分发时间：P2P

- 服务器处理：至少上传一个文件拷贝
- 客户端：每个客户端下载一个文件拷贝
- 全部客户端总下载量为 NF bits



*time to distribute F
to N clients using
P2P approach*

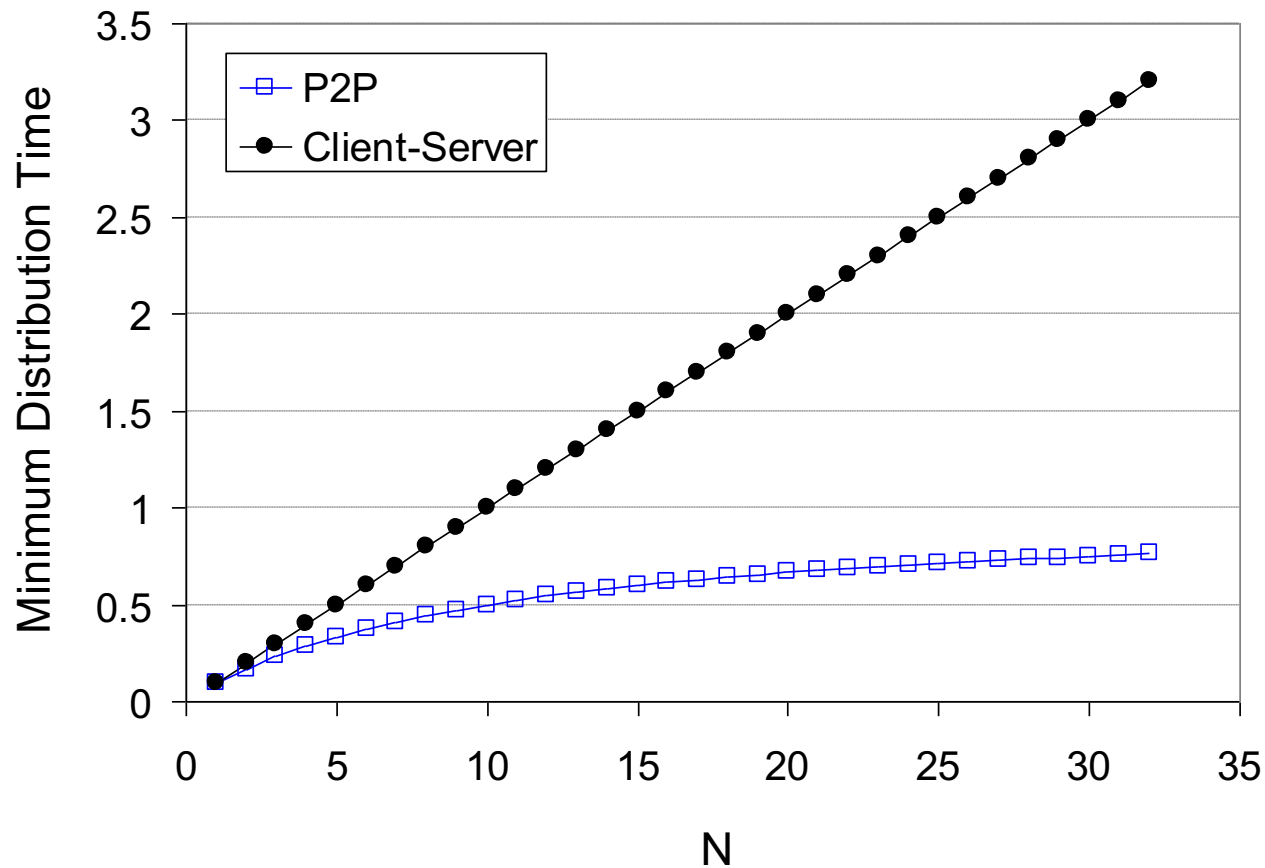
$$D_{P2P} \geq \max\{F/u_s, F/d_{\min}, NF/(u_s + \sum u_i)\}$$

increases linearly in N ...

... but so does this, as each peer brings service capacity

2.6 分布式文件共享：P2P

client upload rate = u , $F/u = 1$ hour, $u_s = 10u$, $d_{min} \geq u_s$



本章小结与作业

重要内容：

- 网络应用程序体系结构。
- 各个应用层协议（HTTP,FTP,SMTP）的作用及端口。
- HTTP的持久连接与非持久连接。
- DNS的作用，层次，域名解析过程。
- 访问网站的步骤。

作业：

- R5, R12, R15, R19, R21
- P4, P5, P22
- 思考题：R11，