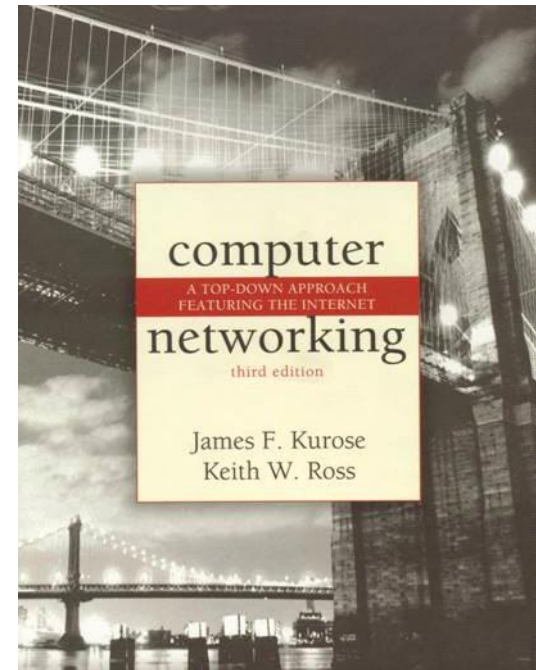


第二章 应用层



第二章 应用层

目的:

- ❑ 网络应用及应用层协议的概念、实现
 - 传输层服务模型
 - 客户机/服务器模式
 - 对等模式 **peer-to-peer**

- ❑ 通过对主流应用层协议分析来掌握应用层协议
 - HTTP
 - FTP
 - SMTP / POP3 / IMAP
 - DNS

第二章：应用层

- **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 因特网中的电子邮件**
 - **SMTP, POP3, IMAP**
- **2.5 DNS：因特网的目录服务**

一些网络应用

- 电子邮件
- **Web**
- 即时讯息
- 远程登陆
- **P2P**文件共享
- 在两台计算机之间的两个帐户之间的文件传输
- 多用户网络游戏
- 流式存储视频片段
- 因特网电话
- 实时视频会议

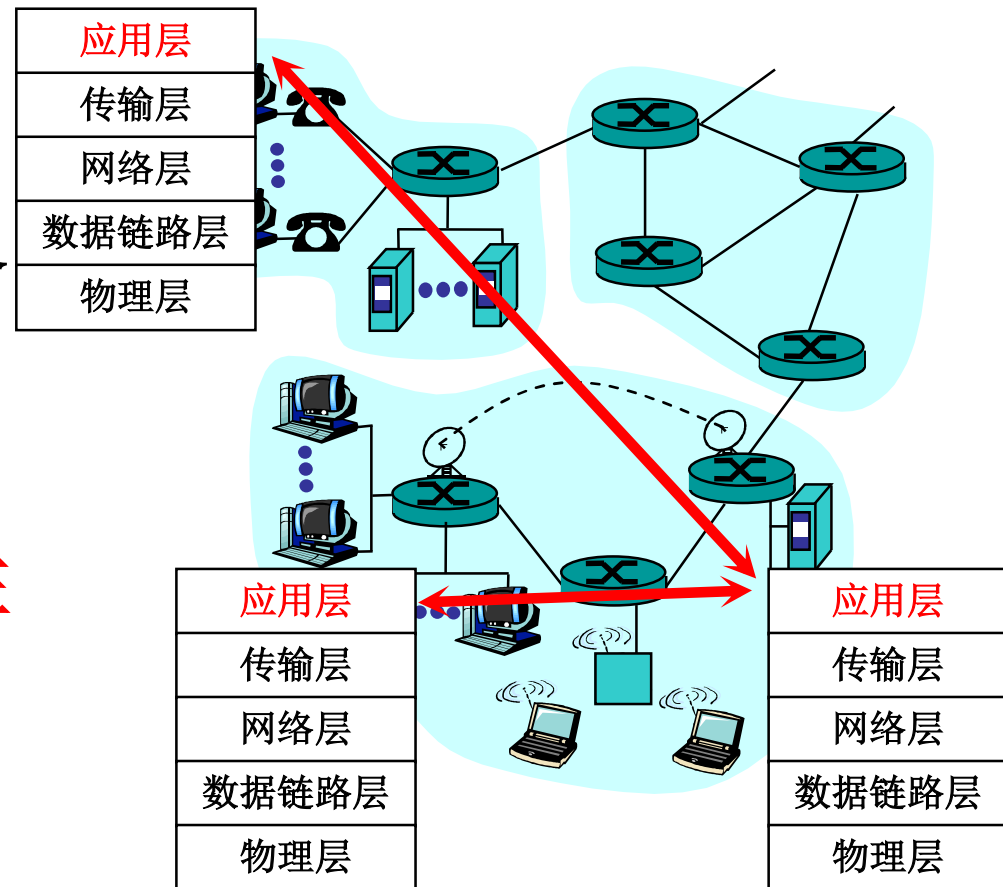
研发网络应用程序

研发网络应用程序的核心：

- 写出能够运行在不同的端系统并通过网络彼此通信的程序
- 例如，**Web**：Web服务器软件和浏览器软件通信

没有应用程序软件运行在网络核心设备上

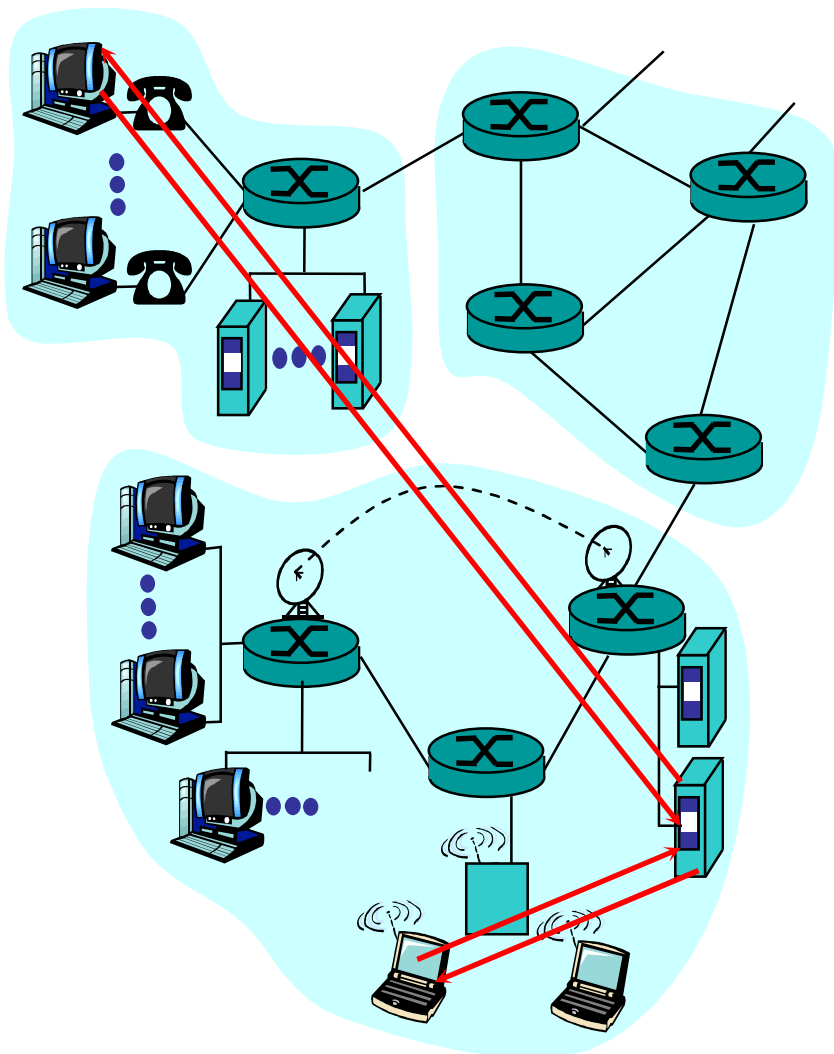
- 网络核心设备不在应用层起作用
- 这种设计方法促进了应用程序的研发



网络应用程序体系结构

- ❑ 客户机/服务器 (**Client/Server, C/S**) 体系结构
- ❑ 对等 (**Peer to Peer, P2P**) 体系结构
- ❑ 客户机/服务器和P2P混合的体系结构

客户机/服务器体系结构



服务器:

- 总是打开的主机，等待客户的请求
- 具有固定的、众所周知的**IP**地址
- 主机集群（数据中心）常被用于创建强大的虚拟服务器

客户机:

- 同服务器端通信
- 可以间断的同服务器连接
- 可以拥有动态**IP**地址
- 客户机相互之间不直接通信

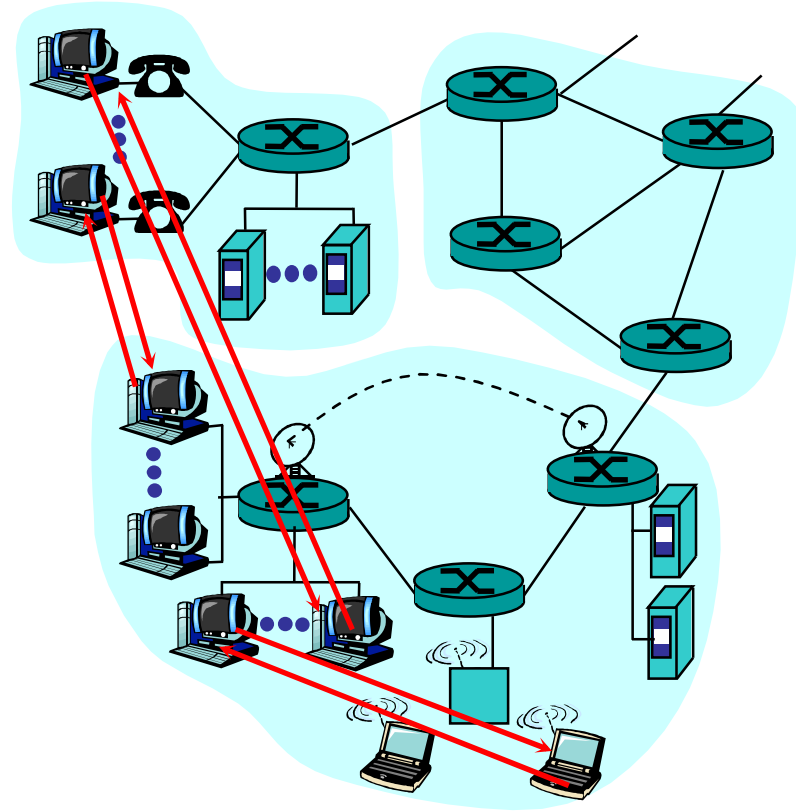
纯P2P体系结构

- ❑ 没有总是打开的服务器
- ❑ 任意一对主机直接相互通信
- ❑ 对等方间歇连接并且可以改变IP地址
- ❑ 例如: **Bittorrent**

优点: 自扩展性

面临挑战:

- ❑ **ISP友好**
- ❑ **安全性**
- ❑ **激励机制**



客户机/服务器和P2P混合的体系结构

Napster

- 文件直接在对等方之间交换
- 文件搜索通过服务器
 - 中心服务器记录对等方内容
 - 对等方查询中心服务器来决定要求的文件位置

即时讯息

- 两个聊天用户之间是P2P
- 注册、查询通过服务器
 - 用户上线时要在中心服务器上注册
 - 用户与中心服务器联系以找出在线伙伴

2.1.2进程通信

进程：运行在端系统中的程序

- 同一主机上的两个进程通过内部进程通信机制进行通信
- 不同主机上的进程通过交换报文相互通信

客户机进程：

发起通信的进程

服务器进程：

等待联系的进程

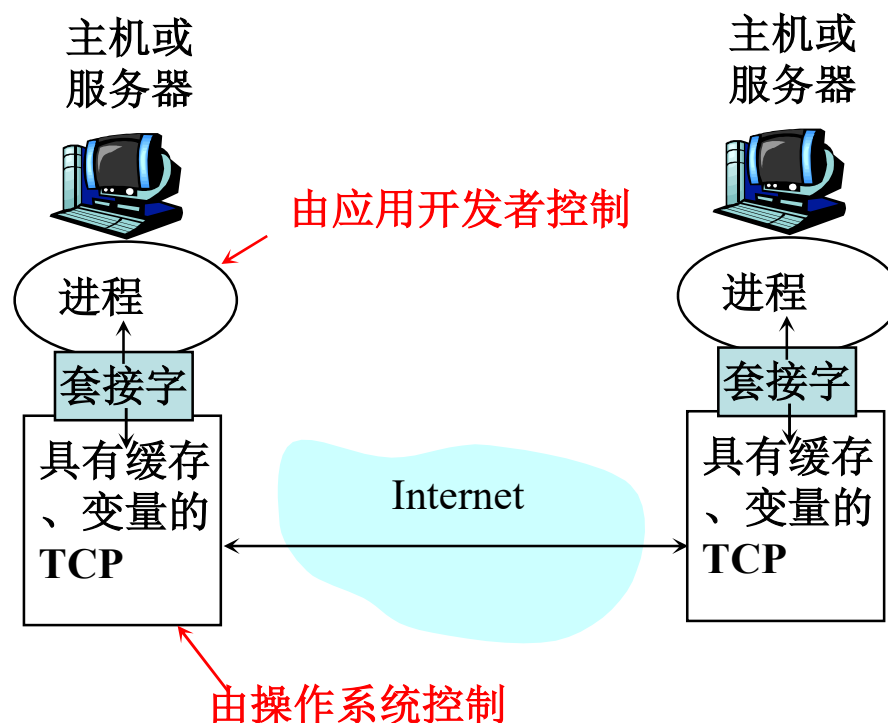
- 注意：具有P2P体系结构的应用程序有客户机进程和服务进程。

套接字

- 进程通过它的套接字在网络上发送和接收报文

- 套接字类比于门户

- 发送进程把报文推出门户
- 发送进程假定门户到另外一侧之间有运输设施，该设施可以传送报文到接收进程



- 套接字又叫做应用程序编程接口 **API**

- 用户通过 **API** 对传输层的控制仅限于：

(1) 选择传输协议；(2) 能设定几个参数

2: 应用层

进程寻址

- ❑ 为了一个进程能接收报文，它需要一个标识
- ❑ 主机有唯一的**32位IP地址**
- ❑ **问：**主机的**IP**地址足够标识进程吗？
- ❑ **答：**不能。因为一台主机上能够运行许多进程。
- ❑ 主机上的进程标识包括**IP地址**和**端口号**
- ❑ 常用应用程序的端口号：
 - **Web服务：80**
 - **邮件服务：25**
- ❑ 将在第三章详细分析端口号

2.1.3可供应用程序使用的服务？

可靠数据传输

- 一些应用（如：实时音频）能容忍一定程度的数据丢失
- 另一些应用（如：文件传输，**telnet**）需要**100%**可靠的数据传输

定时

- 一些应用（如：因特网电话、多方游戏）要求低时延

吞吐量

- 一些应用（如：多媒体）必须要达到所需带宽，带宽敏感
- 另一些应用（弹性应用）根据需要充分利用可供使用的带宽

安全

- 加密解密
- 机密性
- 完整性

常见应用的传输服务需求

应用	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	不能丢失	弹性	不
实时音频/视频	容忍丢失	音频: 几kbps-1Mbps 视频:10kbps-5Mbps	是,100 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	是和不是

2.1.4因特网提供的运输服务

TCP服务:

- ❑ **面向连接的服务:** 在客户机程序和服务器程序之间必须建立连接
- ❑ **可靠的传输服务:** 接收和发送进程间
- ❑ **流量控制:** 发送方不会淹没接收方
- ❑ **拥塞控制:** 网络出现拥塞时抑制发送进程
- ❑ **没有提供:** 时延保证, 最小带宽保证
- ❑ **SSL**

UDP 服务:

- ❑ 不可靠数据传输
- ❑ 没有提供: 建立连接, 可靠性, 流量控制, 拥塞控制, 时延和带宽保证

因特网应用：应用层协议，传输协议

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
流媒体	通常专用 (e.g. RealNetworks)	TCP or UDP
因特网电话	通常专用 (e.g., Skype)	典型用 UDP

2.1.5应用层协议

- 定义不同网络端系统的进程如何传递报文
 - 交换的报文类型，如请求报文和应答报文
 - 报文类型的语法：报文中的各个字段及其详细描述
 - 字段的语义，即包含在字段中的信息的含义
 - 进程何时、如何发送报文及对报文进行响应

公共领域协议：

- 由**RFC**文档定义
- 可供大家使用
- 例如：**HTTP**，**SMTP**

专用协议：

- 例如：**KaZaA**

第二章：应用层

- 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 因特网中的电子邮件
 - SMTP, POP3, IMAP
- 2.5 DNS：因特网的目录服务

Web 和HTTP

常用术语

- 网页（**Web**页，或称文档）由许多**对象**组成。
- 对象就是文件，可以是**HTML**文件，**JPEG**图像，**Java applet**，音频文件...
- 多数网页由单个基本**HTML**文件和若干个所引用的对象构成
- 每个对象被一个**URL**(统一资源定位符)寻址
- 举例**URL**:

`http://www.someschool.edu/someDept/pic.gif`

↑
协议

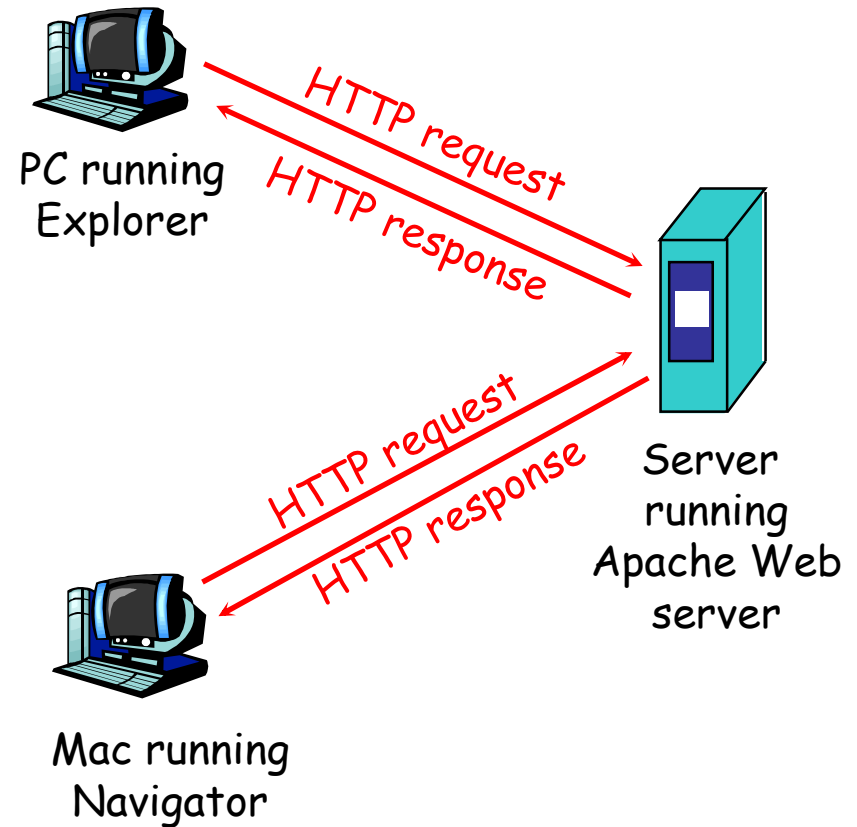
└──────────┬──────────┬──────────┘
主机名 路径名

HTTP概述

HTTP: 超文本传输协议

hypertext transfer protocol

- Web的应用层协议
- client/server模式
 - **client**: 浏览器**browser**请求, 接收, “解释显示” Web对象
 - **server**: Web服务器响应请求, 发送 Web对象
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2616



2. HTTP概述(续)

使用TCP:

- ❑ 客户初始化一个与HTTP服务器80端口的TCP连接 (创建套接字)
- ❑ HTTP服务器接受来自客户的TCP连接请求, 建立连接
- ❑ Browser (HTTP client)和Web服务器 (HTTP server) 交换HTTP消息(应用层协议消息)包括HTTP请求和响应消息
- ❑ 最后结束(或叫关闭)TCP连接

HTTP是无状态协议

- ❑ HTTP服务器不维护客户先前的状态信息

补充

维护状态的协议非常复杂!

- ❑ 必须维护过去历史 (状态信息)
- ❑ 如果server/client崩溃, 它们各自的状态视图可能不一致, 因此必须保持协调一致。

3. HTTP连接

非持久HTTP连接

- 每个TCP连接上只传送一个对象
- HTTP/1.0使用非持久HTTP连接

持久HTTP连接

- 一个TCP连接上可以传送多个对象
- HTTP/1.1默认使用持久HTTP连接

(1). 非持久HTTP连接

网页由1个HTML文件,
和10个jpeg图像构成

假设用户输入URL

`http://www.someSchool.edu/someDepartment/home.index`

1a. HTTP客户初始化1个与服务器主机
`www.someSchool.edu`中HTTP服务
器的TCP连接

1b. `www.someSchool.edu`服务器主
机中的HTTP服务器在80端口监
听来自HTTP客户的TCP连接请求。
收到连接请求, 接受, 建立连接,
通知客户。

2. HTTP客户发送1个HTTP请求消
息 (*request message*) 包含
URL到TCP连接套接字。消息指
出客户要Web对象—
`someDepartment/home.index`

3. HTTP服务器接收请求消息, 产
生1个响应消息*response
message* 包含被请求对象, 并发
送这个消息到自身TCP连接套接
字

time
↓

(1). 非持久HTTP连接(续.)

time
↓

4. HTTP服务器结束TCP连接.

5. HTTP 客户接收包含
html文件的响应消息, 显
示html. 解析html文件,
找出10个引用 jpeg对象

6. 对10个引用jpeg对象的
每1个重复步骤1-5

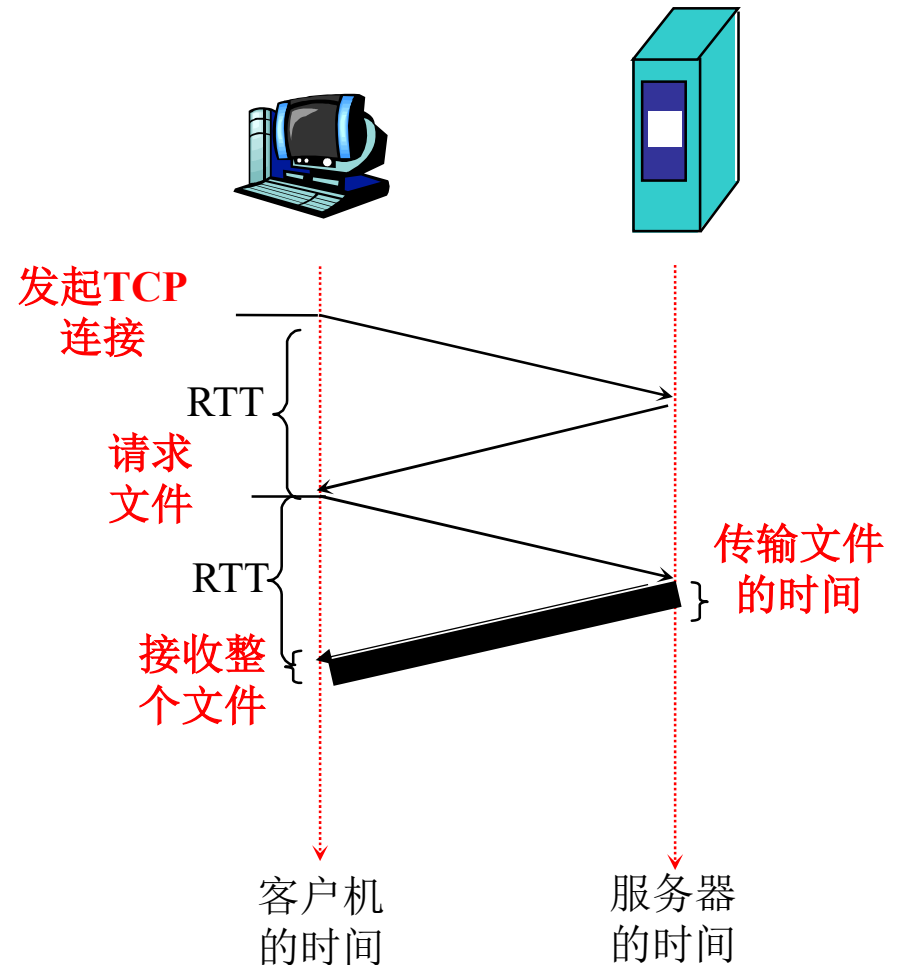
响应时间模型

定义往返时间**RTT**: 1个小分组从客户主机到服务器再到客户主机所花费的时间.

响应时间:

- 1个RTT用于建立TCP连接
- 1个RTT用于HTTP请求/响应消息的交互
- Html文件传输时间

total = 2RTT + transmit time



持久HTTP连接

非持久HTTP连接的问题:

- ❑ 每个对象需要2个RTT
- ❑ 操作系统必须为每个TCP连接分配主机资源
- ❑ 大量客户的并发TCP连接形成服务器的严重负担

持久HTTP连接

- ❑ 服务器发送响应消息后保持连接
- ❑ 同1TCP连接上会话的后续HTTP
- ❑ 消息继续在该连接上传送

不带流水线的持久HTTP连接:

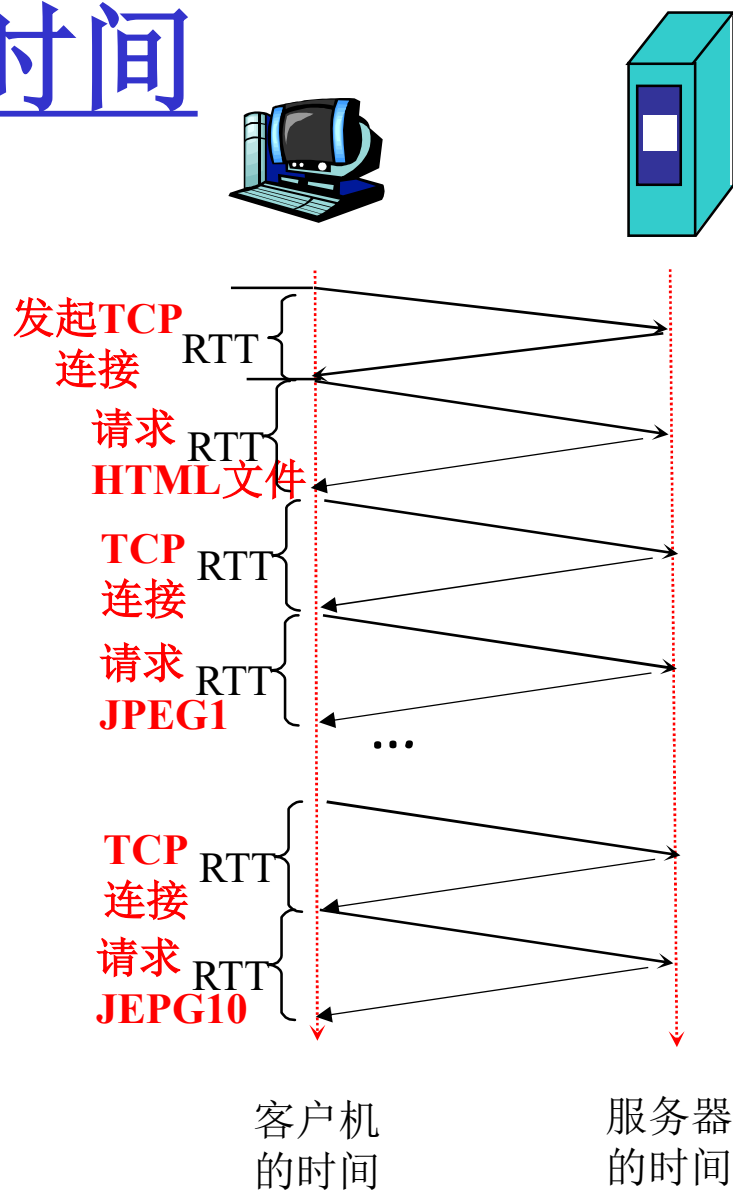
- ❑ 客户先前响应消息收到,才发出新的请求消息
- ❑ 每个引用对象经历1个RTT

带流水线的持久HTTP连接:

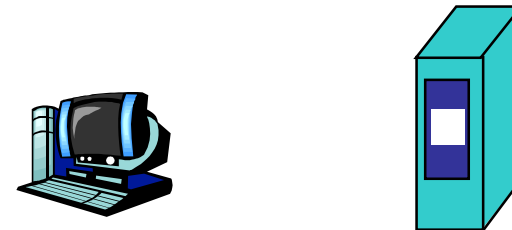
- ❑ 客户遇到1个引用对象就发送请求消息
- ❑ 所有引用对象只经历1个RTT
- ❑ HTTP/1.1默认使用

响应时间

- 网页由1个基本HTML文件和10个JPEG图像组成
- 非持久HTTP连接
 - 每个URL对象需要建立TCP连接，需要一个RTT
 - 每个URL对象请求和响应需要一个RTT
 - 11个URL对象共需22个RTT = $11 * 2RTT$



响应时间

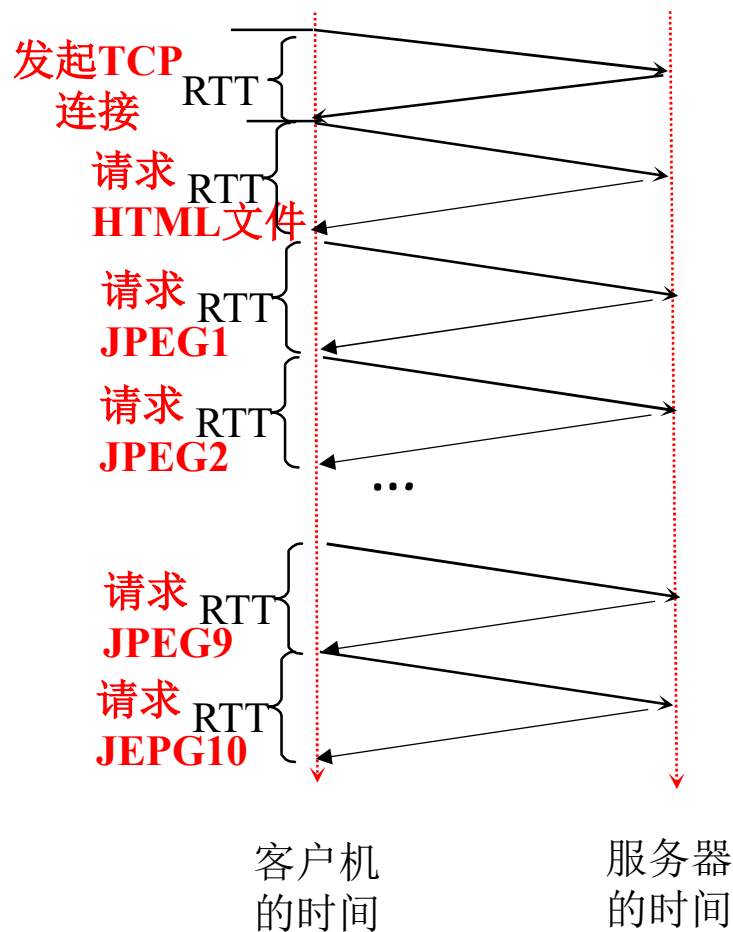


□ 不带流水线的持久

HTTP连接

- 只需建立一个TCP连接，需要一个RTT
- 每个URL对象的请求在前一个URL对象响应返回后才能发出
- 11个URL对象共需12个RTT

$1\text{RTT (TCP)} + 11\text{RTT (每个URL对象)} = 12\text{RTT}$

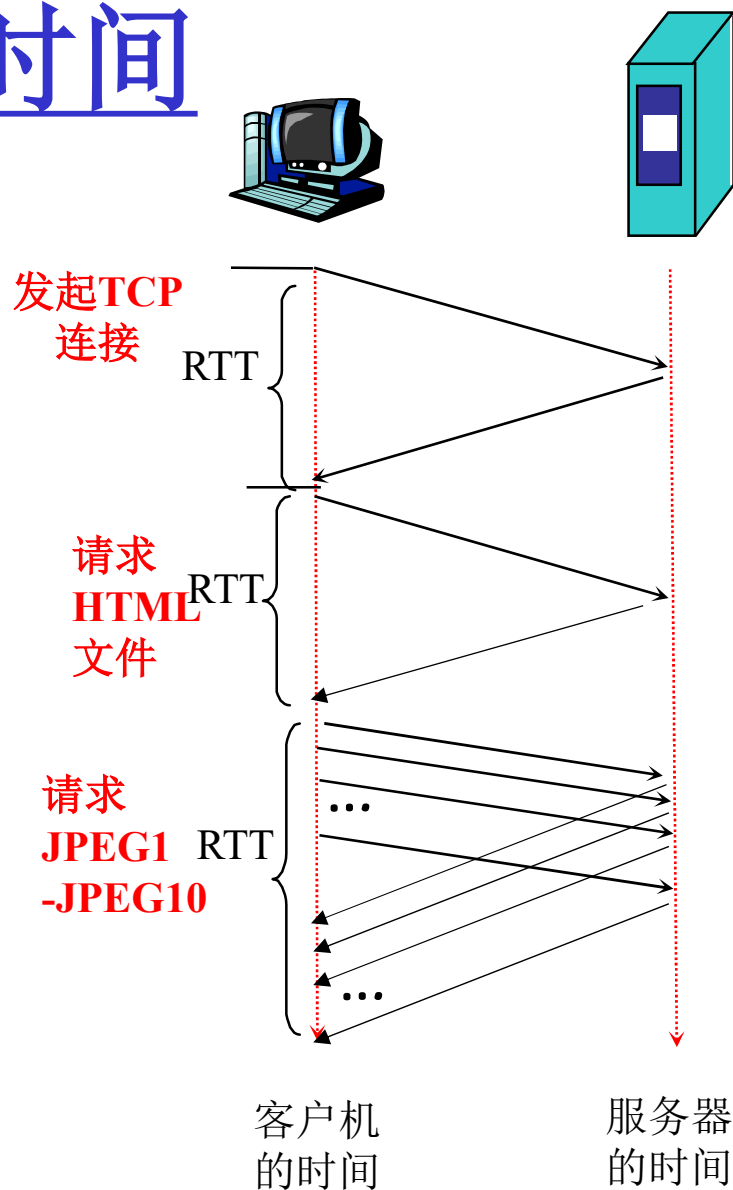


响应时间

□ 带流水线的持久HTTP连接

- 只需建立一个TCP连接，需要一个RTT
- 每个URL对象的请求无需前一个URL对象响应返回后才能发出，可连续发出
- 1个HTML基本文件请求响应需要1个RTT
- 10个JPEG对象“同时”请求，响应信息在1个RTT内返回

$1\text{RTT (TCP)} + 1\text{RTT (HTML)} + 1\text{RTT (10JPEG)}$
 $= 3\text{RTT}$



HTTP报文格式

- 2类HTTP报文:请求报文`request`, 响应报文`response`
- HTTP请求报文:
 - ASCII文本 (易于人读格式)

请求行
(GET, POST, HEAD)

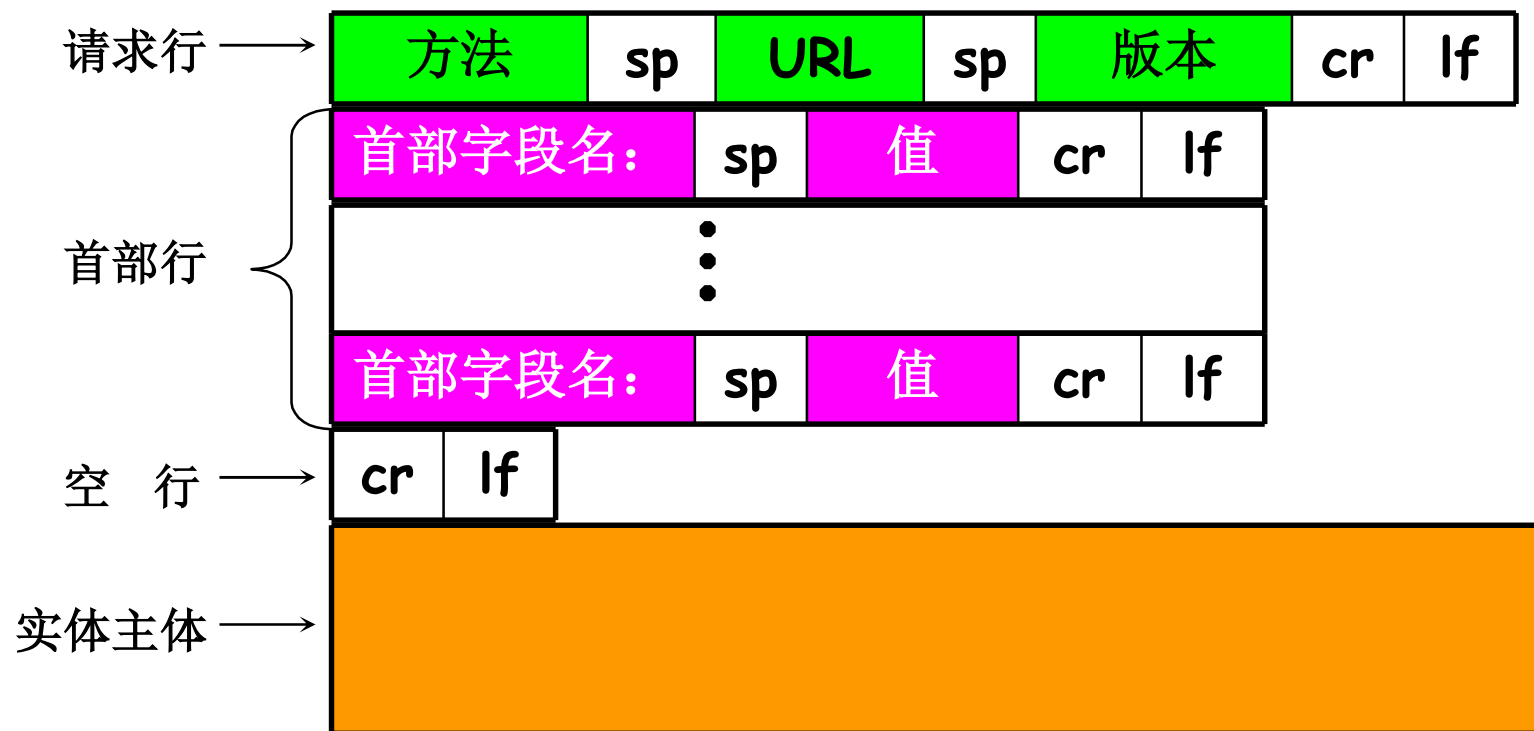
头部行

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0 //该代理类型的对象版本
Connection: Close //不使用持久连接
Accept-language: zh-cn //中文版本
```

回车换行
指示结束

(额外的 回车换行)

HTTP请求报文：通用格式



方法类型

HTTP/1.0

□ GET

□ POST

□ HEAD

- 服务器收到请求时，用HTTP报文进行响应，但不返回请求对象

HTTP/1.1

□ GET, POST, HEAD

□ PUT

- 文件在实体主体中被上载到URL字段指定的路径

□ DELETE

- 删除URL字段指定的文件

上载表单(各字段)输入值

Post方法:

- 网页时常包含表单输入
- 输入值在请求报文的实体主体中被上载到服务器

URL方法:

- 使用**GET**方法
- 表单(各字段)输入值被上载,以URL请求行的字段:
www.somesite.com/animalsearch?monkeys&banana

HTTP响应消息

状态行
(版本、状态编
码、短语)

HTTP/1.1 200 OK

首部行

Connection close

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998

Content-Length: 6821

Content-Type: text/html

数据, e.g.,
被请求的
HTML文件

data data data data data ...

HTTP 响应的状态码

位于服务器响应客户的响应消息的第一行。

几个常见的样本状态码：

200 OK

- 请求成功，所请求信息在响应消息中返回

301 Moved Permanently

- 所请求的对象已永久迁移，新的URL在本响应消息的（**location:** ）头部指出

400 Bad Request

- 该请求不能被服务器解读

404 Not Found

- 服务器上不存在所请求文档

505 HTTP Version Not Supported

HTTP客户与服务器交互实验

1. Telnet 到一个Web服务器:

```
telnet mail.uestc.edu.cn  
80
```

建立同`mail.uestc.edu.cn`主机的HTTP服务器(80端口为默认端口)的TCP连接。
键盘输入发送给`mail.uestc.edu.cn` 主机的80端口。

2. 键盘输入GET方法的HTTP请求消息:

```
GET / HTTP/1.1  
Host: mail.uestc.edu.cn
```

键盘输入后,两次敲回车,你可以发送这个最小限度的 (但完整) **GET**方法的请求消息到HTTP服务器

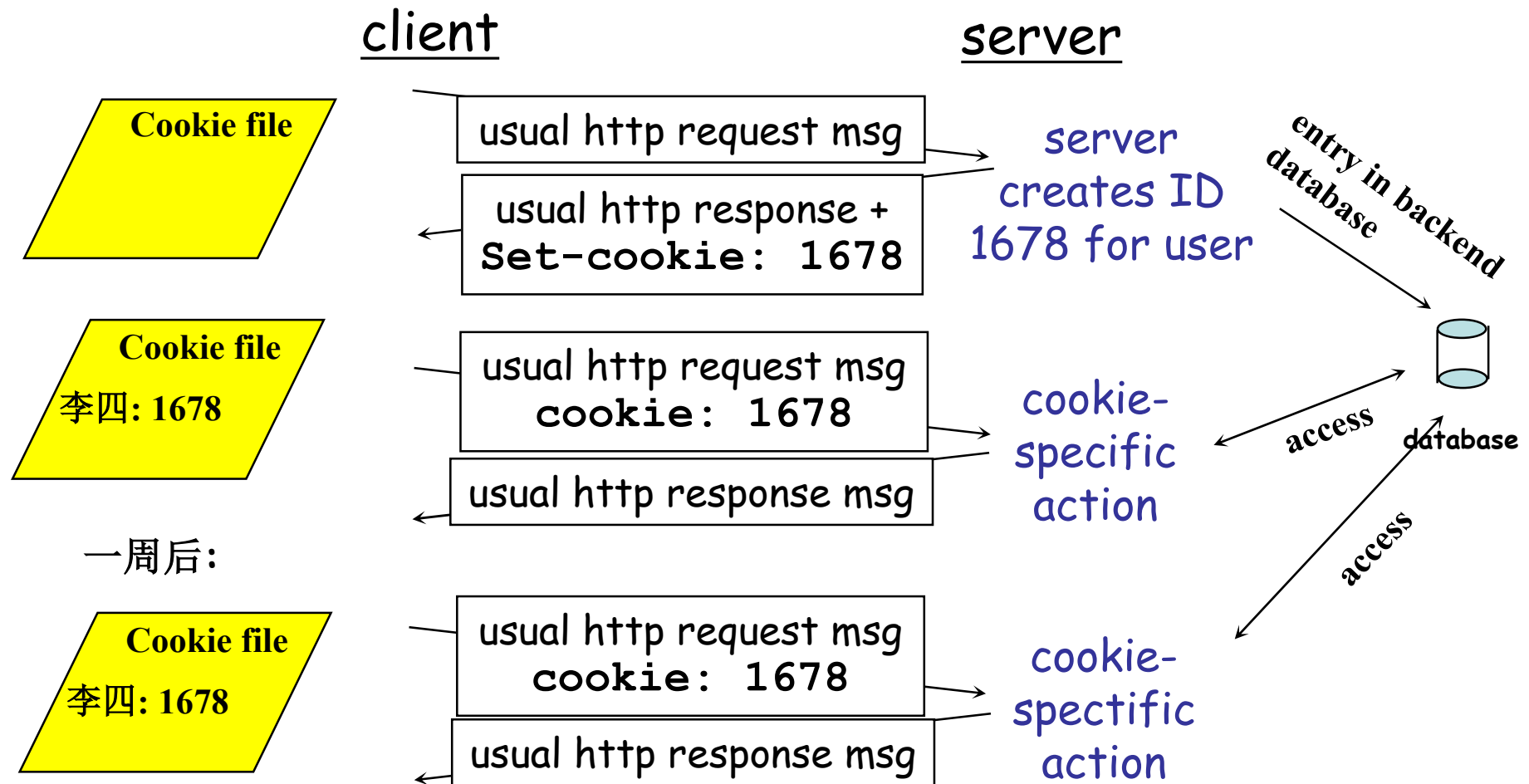
3. 观察HTTP服务器返回的响应消息 !

用户与服务器交互: Cookies

举例:

- 李四总是使用同台**PC**访问**Internet**
- 他首次访问**1**个电子商务网站
- 当他最初发出**HTTP**请求访问该站点时,该站点创建一个唯一的 **ID**,并在后端数据库创建一个响应于该**ID**表项
- 该信息返回给浏览器,并存储在本地——**cookies**文件

Cookies: 跟踪用户(续.)



用户与服务器交互: Cookies

- 1) **cookie**头部行在**HTTP**响应消息中: **Set-cookie**
- 2) **cookie**头部行在**HTTP**请求消息中: **Cookie**
- 3) **cookie**文件保存在用户主机中并被用户浏览器管理
- 4) **cookie**也保存在**Web**站点的后端数据库

Cookies: 跟踪用户(续.)

补充

Cookies可以带来什么 ?

- ❑ 身份认证
- ❑ 虚拟购物车(跟踪用户购买的物品)
- ❑ 推荐广告
- ❑ 用户会话状态 (Web e-mail)

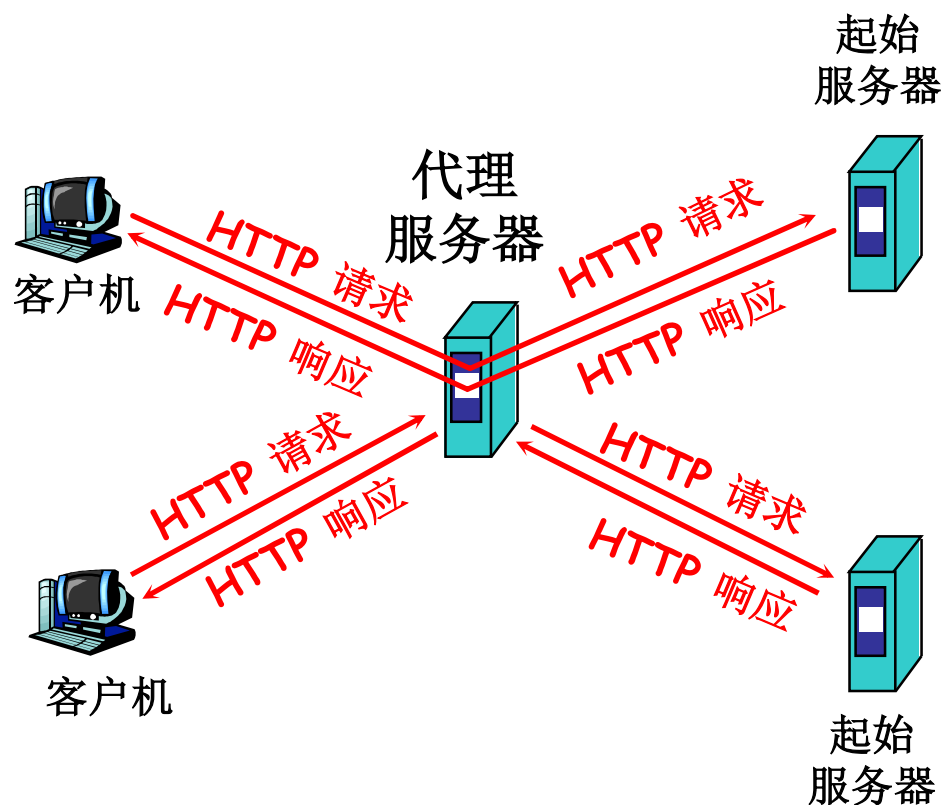
Cookies 和隐私:

- ❑ **cookies**允许网站更加了解你
- ❑ 你提供名字和**e-mail**给网站
- ❑ 广告公司通过网站获得信息
- ❑ **Cookies**不适合游动用户

Web 缓存 (代理服务器)

目标：代表起始服务器满足HTTP请求。

- 用户配置浏览器：
Web 访问经由缓存
- 所有HTTP请求指向缓存
 - 对象在缓存中：缓存器返回对象
 - 否则缓存器向起始服务器发出请求，接收对象后转发给客户机



Web缓存（续）

- 一般的，**Web**缓存器既是服务器又是客户机

为什么需要**Web**缓存器？

- 减少对客户机请求的响应时间
- 减少内部网络与接入链路上的通信量
- 能从整体上大大降低因特网上的**Web**流量

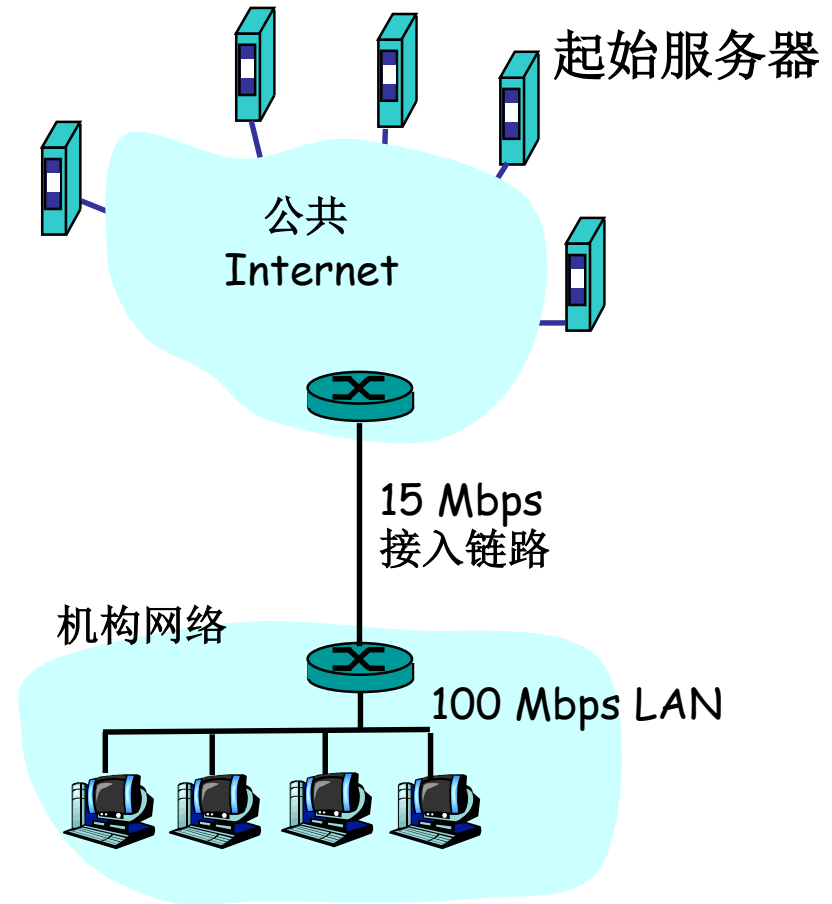
缓存器举例

假设

- ❑ 对象平均长度 = 1M bits
- ❑ 浏览器对对象的平均访问速率 = 15/sec
- ❑ 因特网时延 = 2 sec

结论

- ❑ 局域网上的流量强度 = 0.15
- ❑ 链路上的流量强度 = 1
- ❑ 总延时 = 因特网时延 + 接入时延 + 局域网时延
= 2 秒 + 数分钟 + 数毫秒



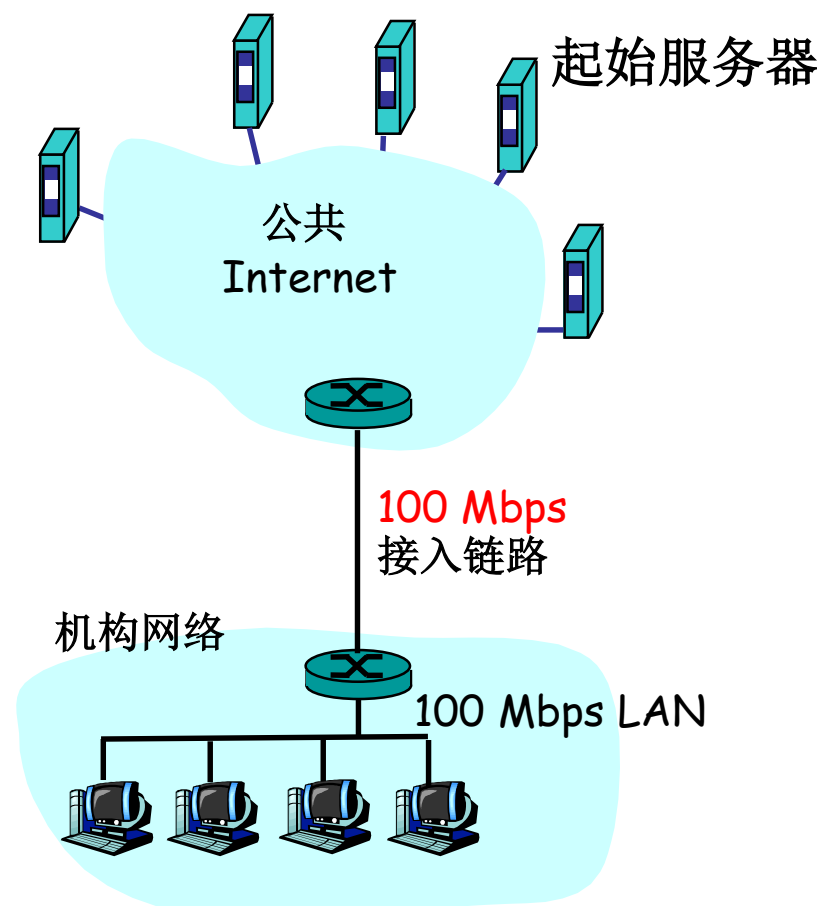
缓存器举例（续）

可能的解决办法

- 增加接入链路的速率

结论

- 局域网上的流量强度 = 15%
- 链路上的流量强度 = 15%
- 总时延 = 因特网时延 + 接入时延 + 局域网时延
= 2 秒 + 数毫秒 + 数毫秒
- 这种方案需要较大的投资



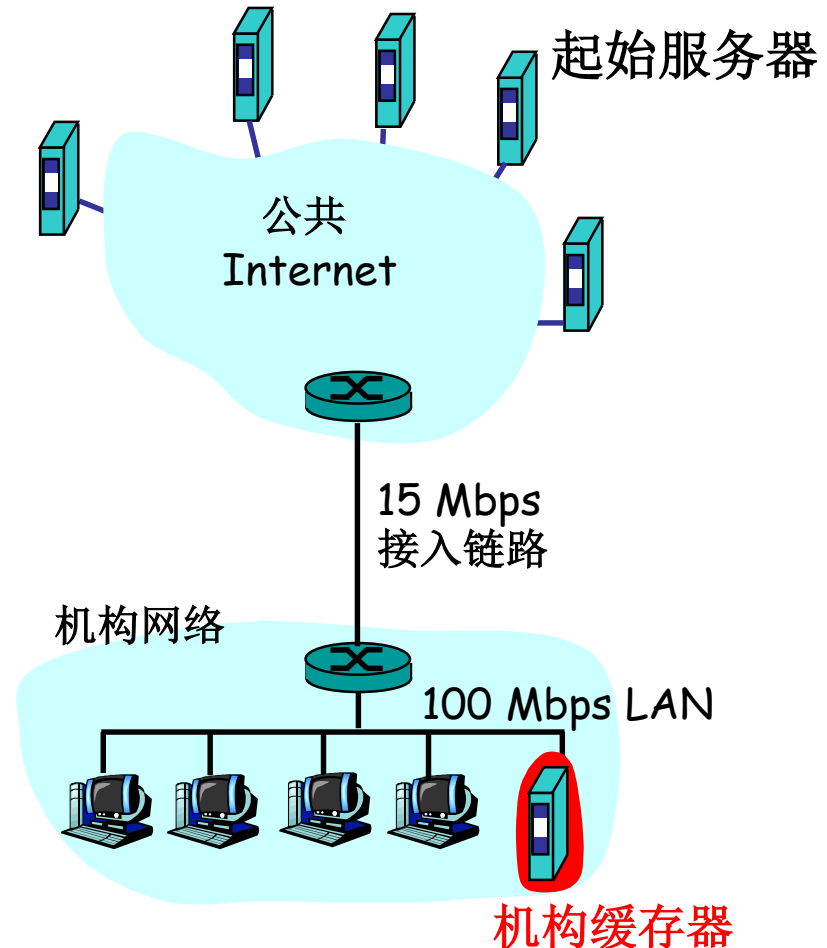
缓存器举例（续）

安装缓存器

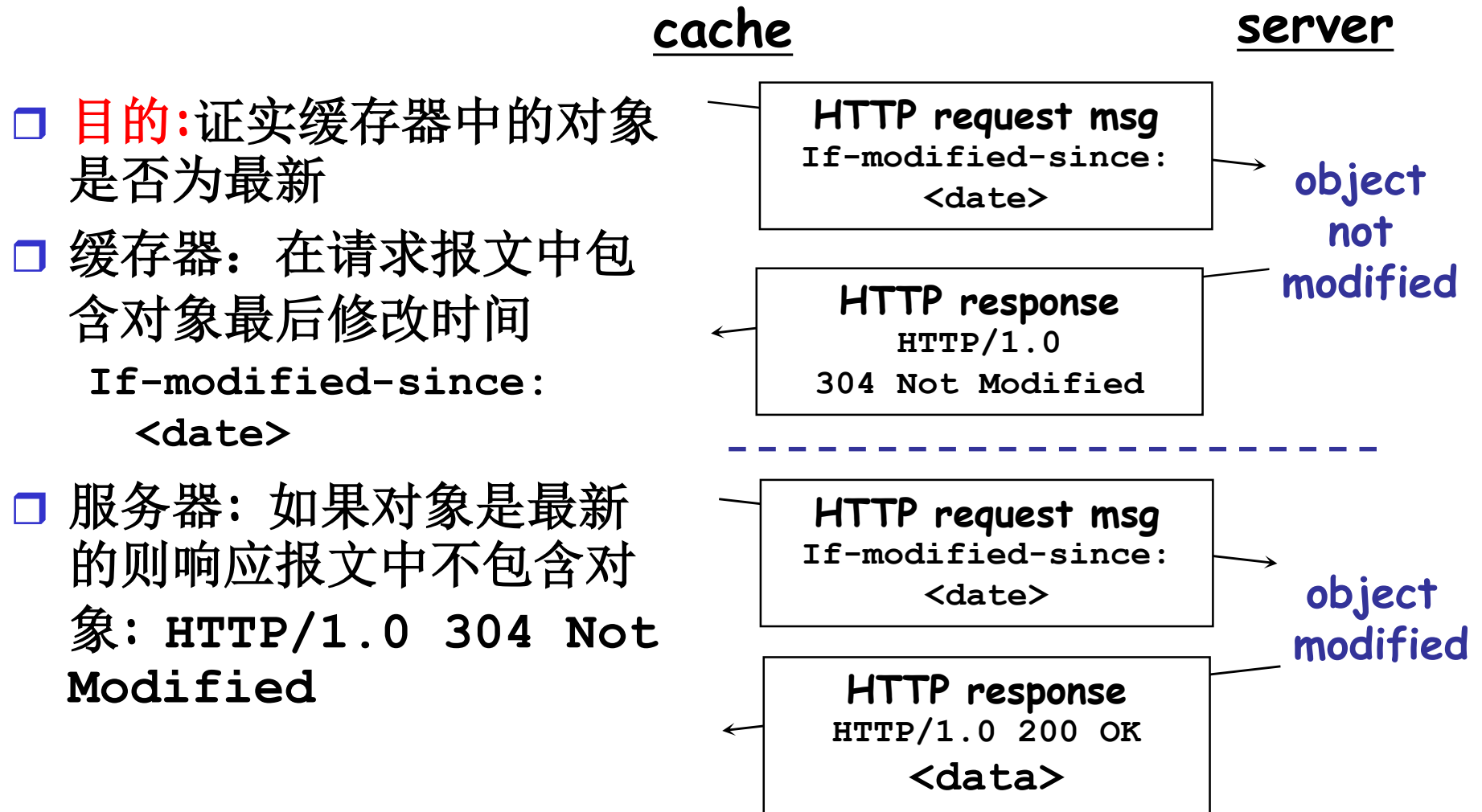
- 假设缓存器命中率为0.4

结论

- 40%的请求立即会得到响应，0.01秒
- 60 %的请求通过访问起始服务器满足
- 链路上的流量强度
 $15\text{Mb} \times 0.6 / 15\text{Mbps} = 0.6$ ，延迟较小
- 总的加权平均延时 = $0.6 \times$
（因特网时延 + 接入时延）
+ $0.4 \times$ 局域网时延
= $0.6 \times (2\text{秒} + \text{数毫秒}) +$
 $0.4 \times \text{数毫秒} < 1.4 \text{ 秒}$



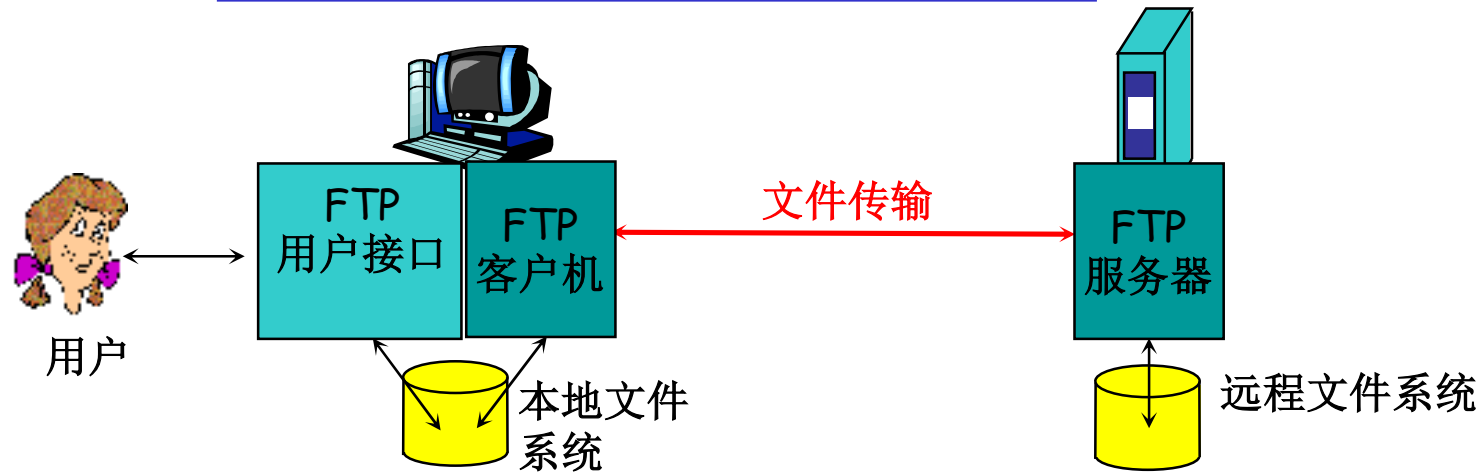
条件GET方法



第二章：应用层

- 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 因特网中的电子邮件
 - SMTP, POP3, IMAP
- 2.5 DNS：因特网的目录服务

FTP: 文件传输协议



- ❑ 传输文件到远程主机/从远程主机下载文件
- ❑ **client/server**模式
 - **client**: 发起传输的一方
 - **server**: 远程主机
- ❑ **ftp**: RFC 959
- ❑ **ftp**服务器: 端口号 21

FTP: 独立的控制连接, 数据连接

- ❑ FTP客户首先发起建立1个与FTP服务器端口号21之间的TCP控制连接, 指定TCP作为传输层协议
- ❑ 客户在建立的控制连接上获得身份认证
- ❑ 客户在建立的控制连接上发送命令来浏览远程主机的目录.
- ❑ 当服务器接收到1个文件传输命令时, 在服务器端口号20创建1个与客户的TCP数据连接
- ❑ 1个文件传输后, 服务器结束这个TCP数据连接.



- ❑ 服务器创建第2个TCP与客户的数据连接来传输下一个文件.
- ❑ 控制连接: 带外发送控制信息
- ❑ FTP 服务器要维护用户状态信息: 当前目录, 先前的身份认证

FTP命令和应答

常见命令:

- ❑ 在控制连接上发送**ASCII**文本
- ❑ **USER** *username*
- ❑ **PASS** *password*
- ❑ **LIST**: 返回当前远程目录的文件列表
- ❑ **RETR filename**: 获取远程主机当前目录下的1个文件(**get**)
- ❑ **STOR filename**: 存放1个文件到远程主机当前目录下(**put**)

常见应答:

- ❑ 状态码及其相应短语 (同**HTTP**)
- ❑ **331 Username OK, password required**
- ❑ **125 data connection already open; transfer starting**
- ❑ **425 Can't open data connection**
- ❑ **452 Error writing file**

第二章：应用层

- 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 因特网中的电子邮件
 - SMTP, POP3, IMAP
- 2.5 DNS：因特网的目录服务
- 2.6 P2P 文件共享

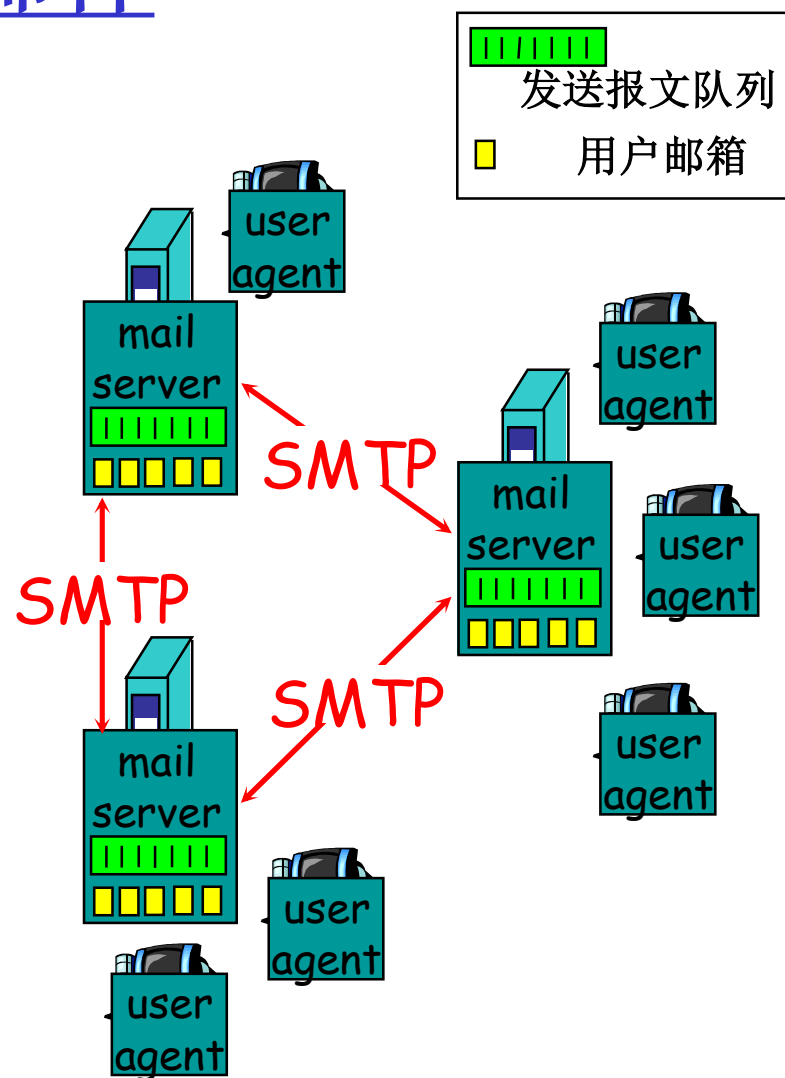
电子邮件

3个主要组成部分:

- ❑ 用户代理user agents
- ❑ 邮件服务器mail servers
- ❑ 简单邮件传送协议simple mail transfer protocol: SMTP

用户代理:

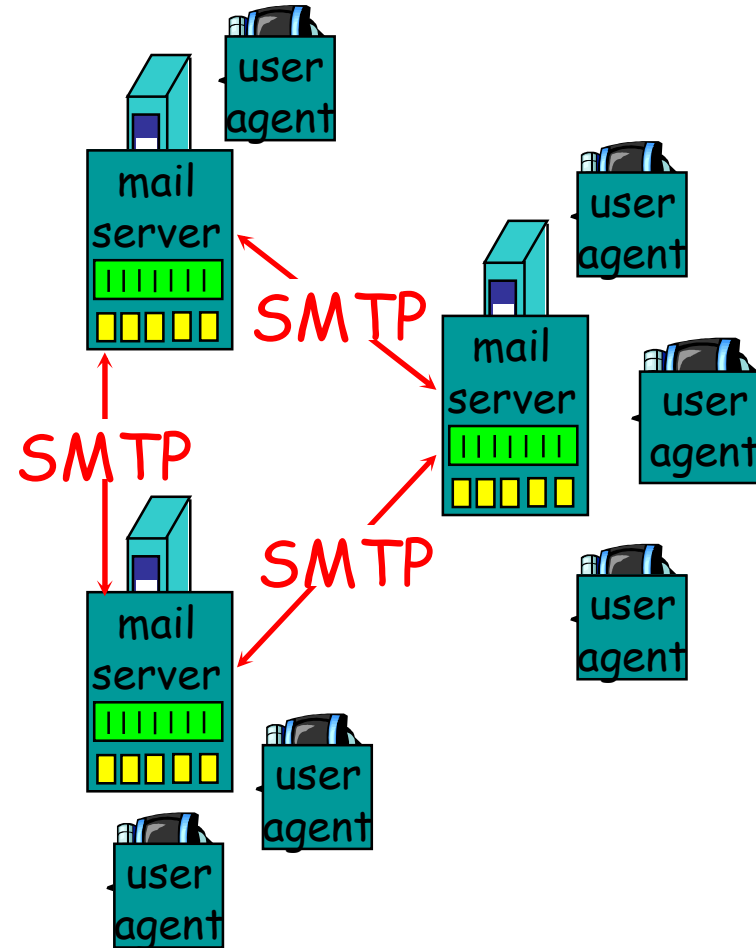
- ❑ 允许用户阅读,回复,转发,保存,编辑邮件消息
- ❑ 例如: Outlook, foxmail等
- ❑ 发送邮件消息到服务器/从服务器接收邮件消息



电子邮件：邮件服务器

邮件服务器

- ❑ 邮箱**mailbox** 存放用户接收的邮件消息
- ❑ 外出报文队列**outgoing message queue**
- ❑ **SMTP**协议在邮件服务器间发送邮件消息
 - **client**: 运行在发送邮件消息的服务器上
 - “**server**”: 运行在接收邮件消息的服务器上

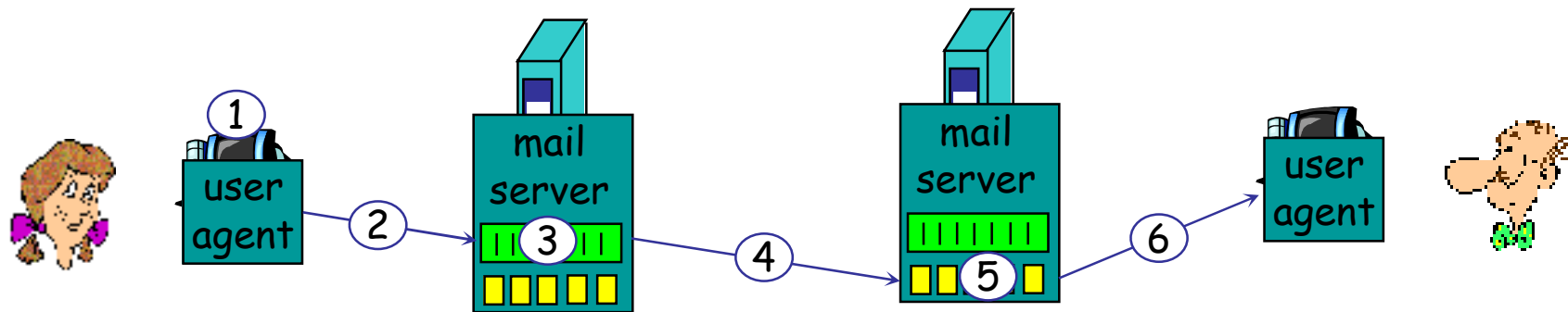


电子邮件: SMTP [RFC 2821]

- ❑ 客户使用**TCP**来可靠传输邮件消息到服务器端口号**25**
- ❑ 直接传送: 发送服务器到接收服务器
- ❑ 传输的**3**个阶段
 - 握手 (问候)
 - 邮件消息的传输
 - 结束
- ❑ 命令/应答的交互
 - 命令: **ASCII**文本格式
 - 应答: 状态码及其短语
- ❑ 邮件消息必须是**7-bit ASCII**

例子: Alice发送邮件消息到Bob

- 1) **Alice**使用用户代理编写邮件消息(给**bob@someschool.edu**的)
- 2) **Alice**的用户代理发送邮件消息到她的邮件服务器;邮件消息存放在邮件消息队列
- 3) **Alice**邮件服务器的**SMTP**客户端发起建立一个到**Bob**的邮件服务器的**SMTP**服务器端的**TCP**连接,经过应用层握手.
- 4) **SMTP**客户在这个**TCP**连接上发送**Alice**的邮件消息
- 5) **Bob**服务器存放邮件消息存到**Bob**的邮箱
- 6) **Bob**调用他的用户代理读邮件消息



SMTP客户与服务器交互实验:

- ❑ `telnet servername 25`
- ❑ 看到服务器的应答220
- ❑ 键盘输入HELO, MAIL FROM, RCPT TO, DATA, QUIT 命令

以上可以让你不使用邮件阅读器发送邮件消息

SMTP客户和服务器的命令交互

```
telnet hamburger.edu 25
```

```
S: 220 hamburger.edu
```

```
C: HELO crepes.fr
```

```
S: 250 Hello crepes.fr, pleased to
```

```
C: MAIL FROM: <alice@crepes.fr>
```

```
S: 250 alice@crepes.fr... Sender ok
```

```
C: RCPT TO: <bob@hamburger.edu>
```

```
S: 250 bob@hamburger.edu ... Recipient ok
```

```
C: DATA
```

```
S: 354 Enter mail, end with "." on a line by itself
```

```
C: Do you like ketchup?
```

```
C: How about pickles?
```

```
C: .
```

```
S: 250 Message accepted for delivery
```

```
C: QUIT
```

```
S: 221 hamburger.edu closing connection
```

C:auth login

S:334 VXNlcm5hbWU6

C:用户名 (base64编码)

S:334 UGFzc3dvcmQ6

C:密码 (base64编码)

S:235 go ahead

□ 传输阶段

□ 结束阶段

SMTP: 小结

- ❑ SMTP使用持久连接
- ❑ SMTP 要求邮件消息 (header & body)必须是 7-bit ASCII
- ❑ SMTP服务器使用 CRLF.CRLF 来判断邮件消息的结束

与HTTP的比较:

- ❑ HTTP: 拉协议
- ❑ SMTP: 推协议
- ❑ 都有ASCII 命令/应答交互, 状态码
- ❑ HTTP: 每个对象封装在它各自的HTTP响应消息中发送
- ❑ SMTP: 一个邮件内各个对象置于同一个邮件消息的多目部分发送

邮件消息的格式

SMTP: 用来交换邮件消息的协议

RFC 822: 文本邮件消息格式标准:

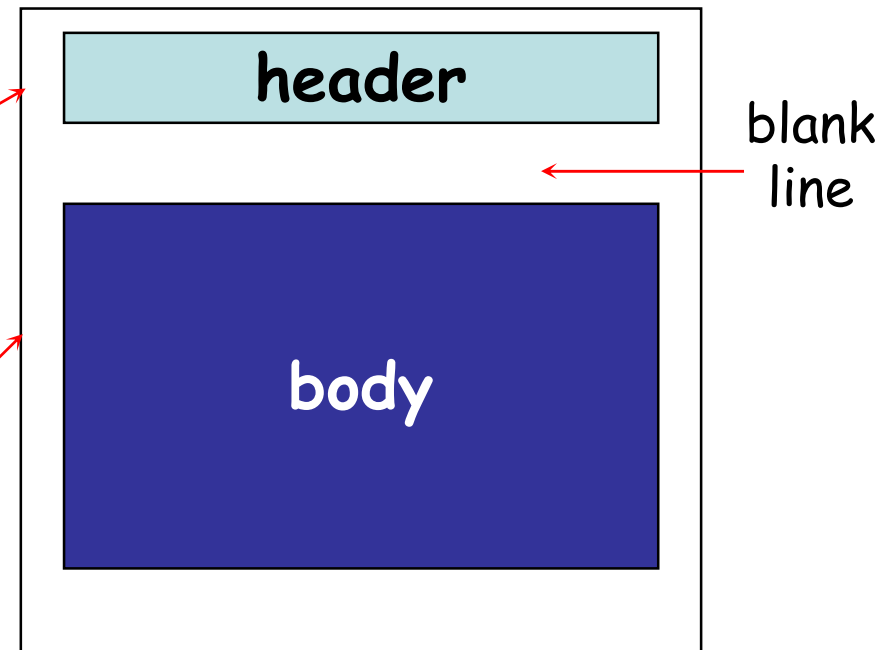
□ 信头—头部行。如:

- To:
- From:
- Subject:

这些头部不同于SMTP命令!

□ 信体

- 邮件消息也必须是**ASCII**字符

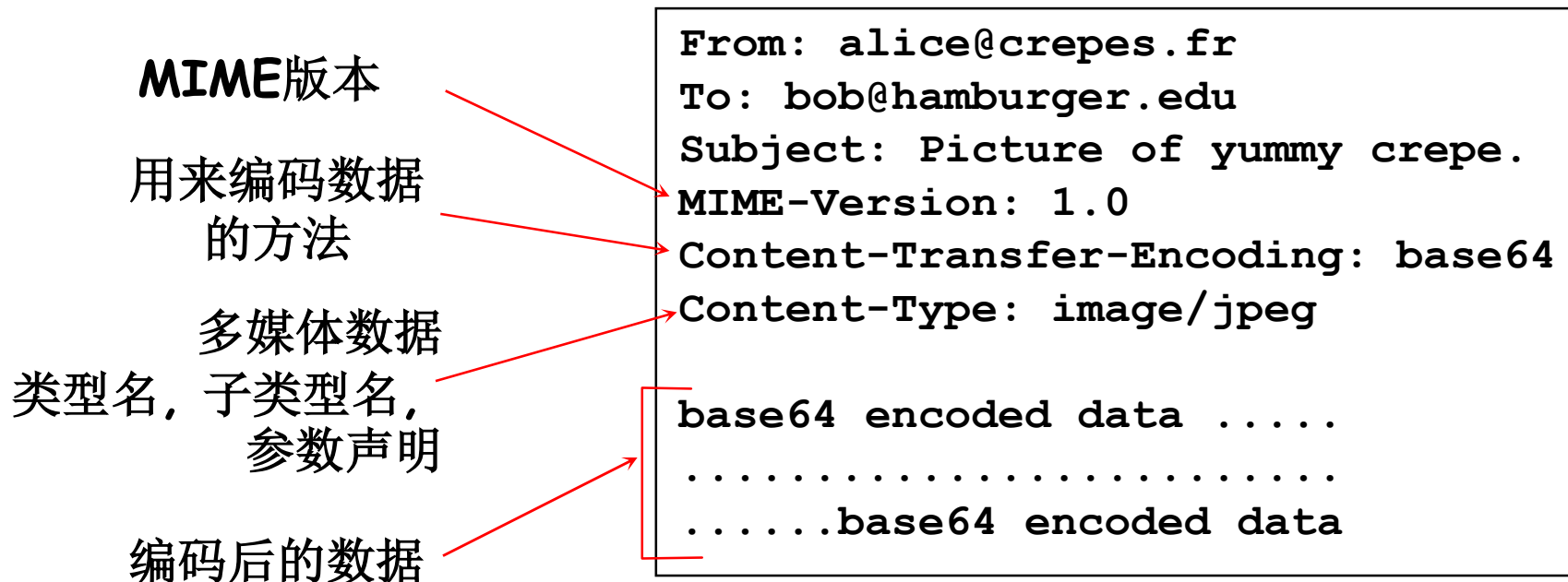


邮件消息的格式：多媒体扩展

- **SMTP**协议具有以下缺点
 - **SMTP**不能传送可执行该文件或其他的二进制对象。
 - **SMTP**限于传送7位的**ASCII**码。
 - **SMTP**服务器会拒绝超过一定长度的邮件。
- **MIME: Multipurpose Internet mail Extensions** 多用途因特网邮件扩展, **RFC 2045, 2046**
 - **MIME**并没有改动或取代**SMTP**。
 - **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。
 - **MIME**邮件可在现有的电子邮件程序和协议下传送。

邮件消息的格式：多媒体扩展

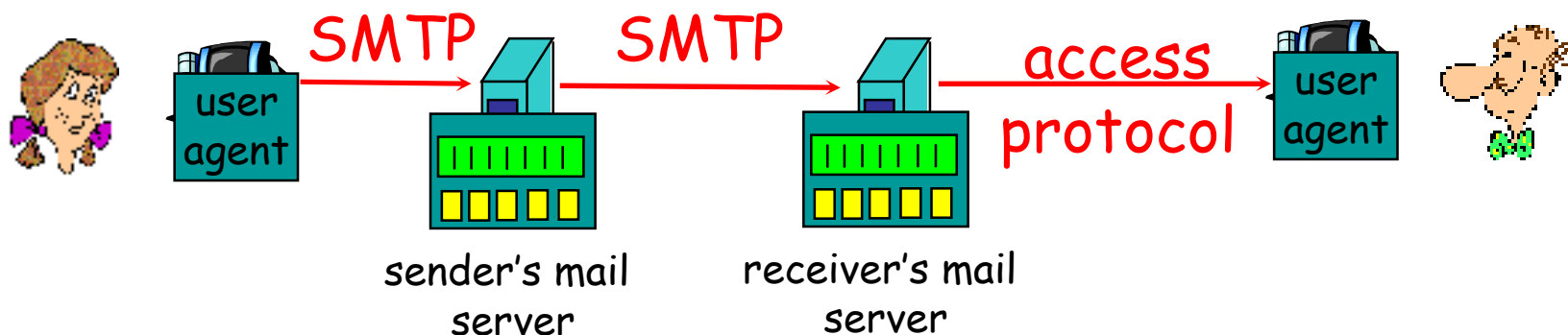
- ❑ 5个新邮件首部字段，它们可包含在RFC 822首部中。这些字段提供了有关邮件主体的信息。**MIME Version, Content Type, Content-transfer-encoding**
- ❑ 定义了许多邮件内容的格式，对多媒体电子邮件的表示方法进行了标准化。
- ❑ 定义了传送编码，可对任何内容格式进行转换，而不会被邮件系统改变。



MIME 和 SMTP 的关系



邮件访问协议



- ❑ **SMTP**: 递送/存储邮件消息到接收者邮件服务器
- ❑ 邮件访问协议: 从服务器获取邮件消息
 - **POP: Post Office Protocol** 邮局协议[RFC 1939]110端口号
 - 身份认证 (代理 <--> 服务器) 并 下载邮件消息
 - **IMAP: Internet Mail Access Protocol [RFC 1730]**
 - 更多功能特征 (更复杂!)
 - 允许用户像对待本地邮箱那样操纵远程邮箱的邮件
 - **HTTP: Hotmail , Yahoo! Mail, etc.**

POP3协议

身份认证阶段

authorization phase

❑ 客户命令:

- user username

- pass password

❑ 服务器响应

- +OK

- -ERR

传输阶段

transaction phase, client:

❑ list: 列出邮件编号

❑ retr: 按编号取邮件

❑ dele: 删除

更新阶段

❑ quit

S: +OK POP3 server ready

C: user bob

S: +OK

C: pass hungry

S: +OK user successfully logged on

C: list

S: 1 498

S: 2 912

S: .

C: retr 1

S: <message 1 contents>

S: .

C: dele 1

C: retr 2

S: <message 1 contents>

S: .

C: dele 2

C: quit

S: +OK POP3 server signing off

POP3协议

POP3的更多细节

- ❑ 先前例子使用
“Download-and-delete”.
- ❑ Bob换客户端后不能再读邮件
- ❑ “Download-and-keep”模式：在不同客户机上的邮件拷贝
- ❑ POP3的会话是无状态的

IMAP协议

IMAP

- ❑ **RFC3501，端口143**
- ❑ 保存所有邮件消息在一个位置：服务器
- ❑ 允许用户在服务器的各文件夹中管理邮件消息
 - 每个报文和一个文件夹联系起来
 - 创建文件夹
 - 移动邮件，查询邮件
- ❑ **IMAP跟踪用户会话的状态信息：**
 - 文件夹和邮件消息**IDs**与文件夹名字的映射
- ❑ 允许获邮件组件

基于Web的电子邮件

- ❑ Hotmail、Gmail...
- ❑ 用户代理是浏览器，通过**HTTP**和邮件服务器通信
- ❑ 邮件服务器之间仍然使用**SMTP**协议

第二章：应用层

- 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 因特网中的电子邮件
 - SMTP, POP3, IMAP
- 2.5 DNS：因特网的目录服务

DNS: 域名系统Domain Name System

人: 很多标识符:

- ID, name, passport #

Internet主机, 路由器:

- IP address (32 bit) - 用于分组寻址
- “主机名”, e.g., `gaia.cs.umass.edu` - 用于人记忆识别

Q: 可以在IP地址和主机名之间建立映射吗 ?

Domain Name System:

域名系统

- 分布式数据库: 一个由分层DNS服务器实现的分布式数据库
- 应用层协议: DNS服务器实现域名转换 (域名/地址转换)
- C/S, UDP, 端口53

DNS

DNS服务器提供的功能:

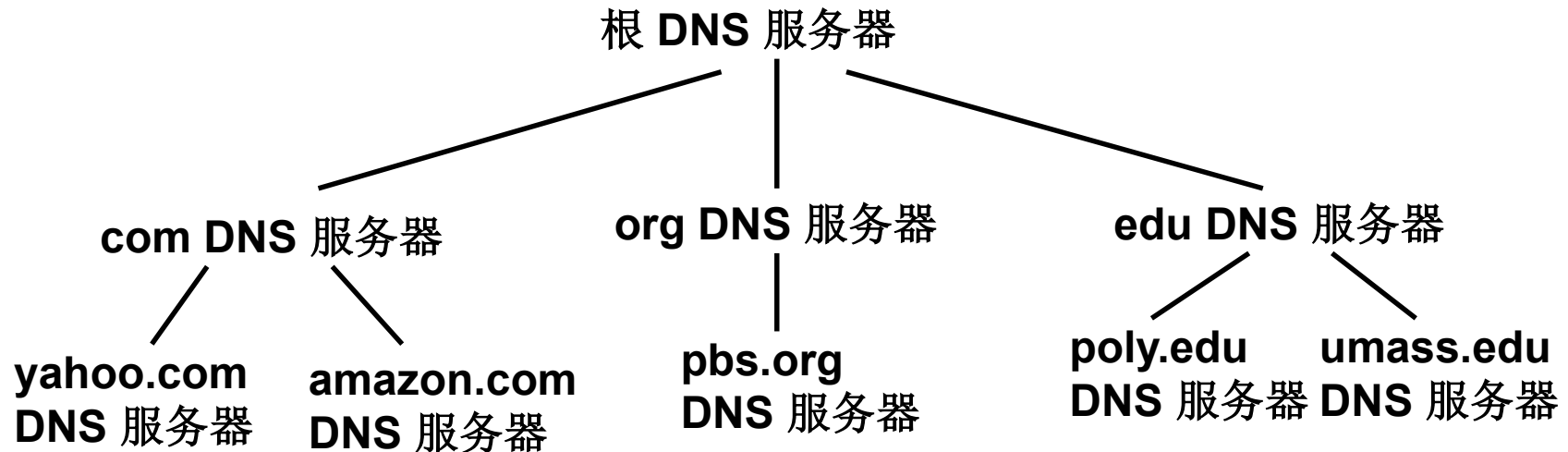
- ❑ 主机名到**IP**地址的转换
- ❑ 主机别名
 - 一个主机可以有一个规范主机名和多个主机别名
- ❑ 邮件服务器别名
- ❑ 负载分配
 - **DNS**实现冗余服务器: 一个**IP**地址集合可以对应于同一个规范主机名。

为什么不集中式**DNS**?

- ❑ 单点故障
- ❑ 巨大访问量
- ❑ 远距离集中式数据库
- ❑ 维护

不可扩展!

分布式、层次数据库

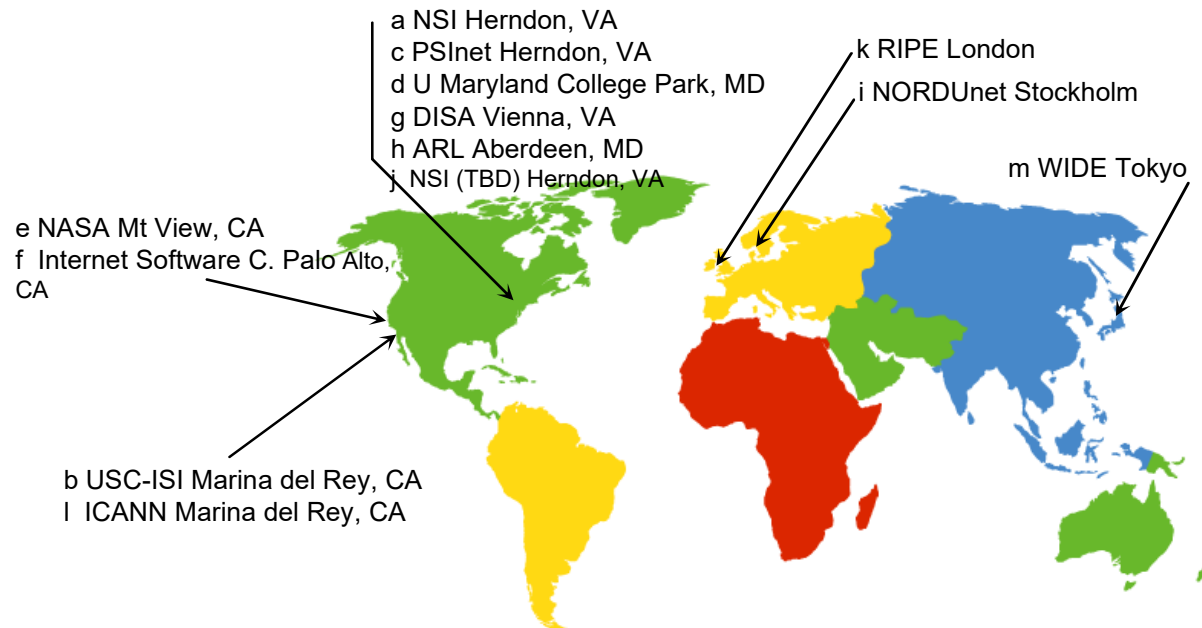


客户机怎样决定主机名**www.amazon.com**的IP地址？

- ❑ 客户机查询根服务器得到**com DNS**服务器
- ❑ 客户机查询**com DNS**服务器得到**amazon.com DNS**服务器
- ❑ 客户机查询**amazon.com DNS**服务器得到**www.amazon.com**的IP地址

DNS: 根名字服务器Root name servers

❑ 根名字服务器负责记录顶级域名服务器的信息



**13 root name
servers
worldwide**

顶级域服务器和权威DNS服务器

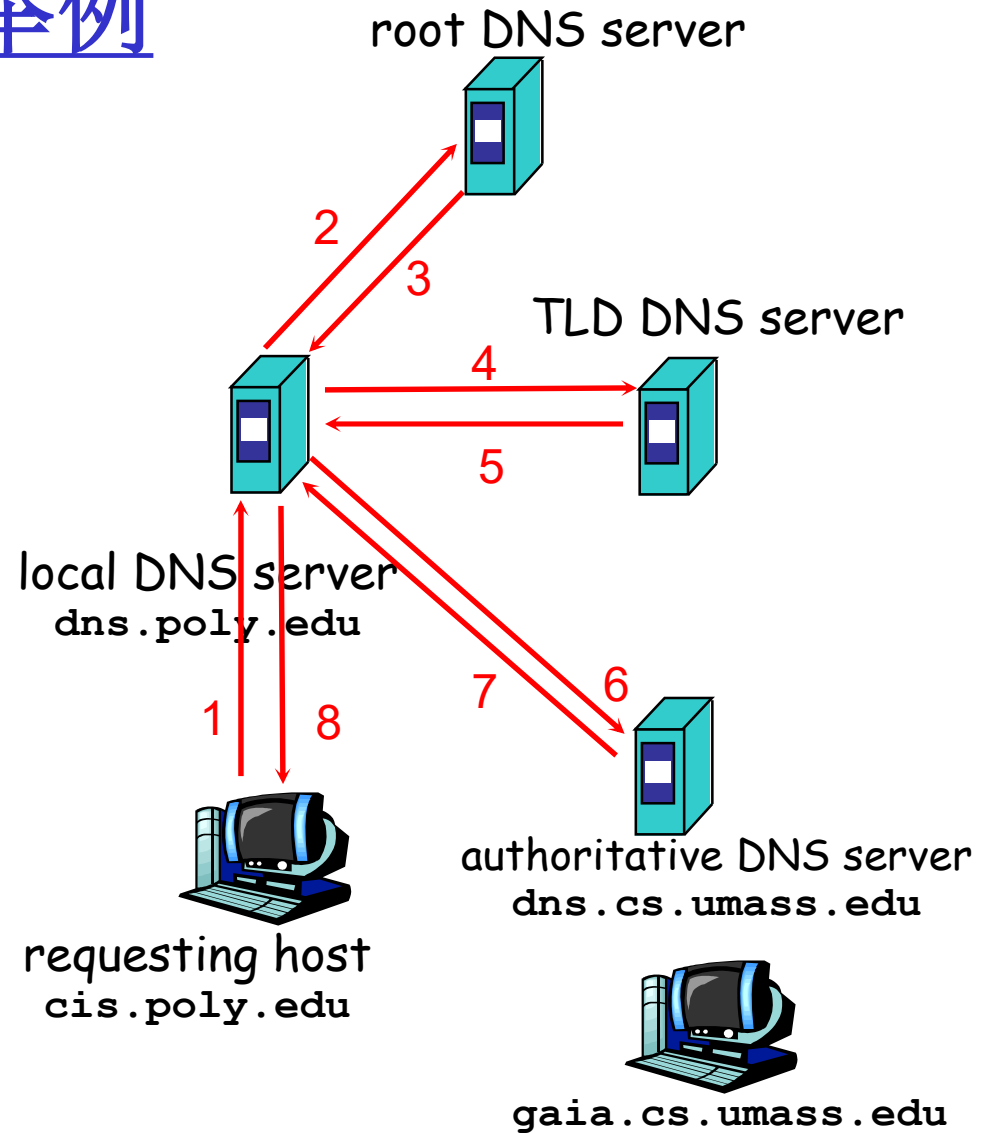
- **顶级域服务器：** 负责通用顶级域名 `com`, `org`, `net`, `edu`, `etc`, 和所有国家的顶级域名 `uk`, `fr`, `ca`, `jp`.
 - **Network solutions** 公司维护`com`顶级域的TLD服务器
 - **Educause** 公司维护`edu`顶级域的 TLD服务器
- **权威DNS服务器：** 在因特网上具有公共可访问主机（如Web服务器和邮件服务器）的每个组织机构必须提供公共可访问的**DNS**记录，这些记录将这些主机的名字映射为**IP**地址。组织机构的权威**DNS**服务器负责保存这些**DNS**记录。
 - 多数大学和公司维护它们的基本权威**DNS**服务器
 - 例如电子科技大学维护权威**DNS**服务器`dns.uestc.edu.cn`，保存了所有`uestc.edu.cn`域的**dns**记录

本地DNS服务器

- ❑ 严格来说不属于该服务器的层次结构
- ❑ 每个**ISP**（如居民区**ISP**、公司、大学）都有一个本地**DNS**
 - 也叫默认服务器
- ❑ 当主机发出**DNS**请求时，该请求被发往本地**DNS**服务器。
 - 起着代理的作用，转发请求到层次结构中。

举例

- 主机 `cis.poly.edu` 想知道主机 `gaia.cs.umass.edu` 的 IP 地址



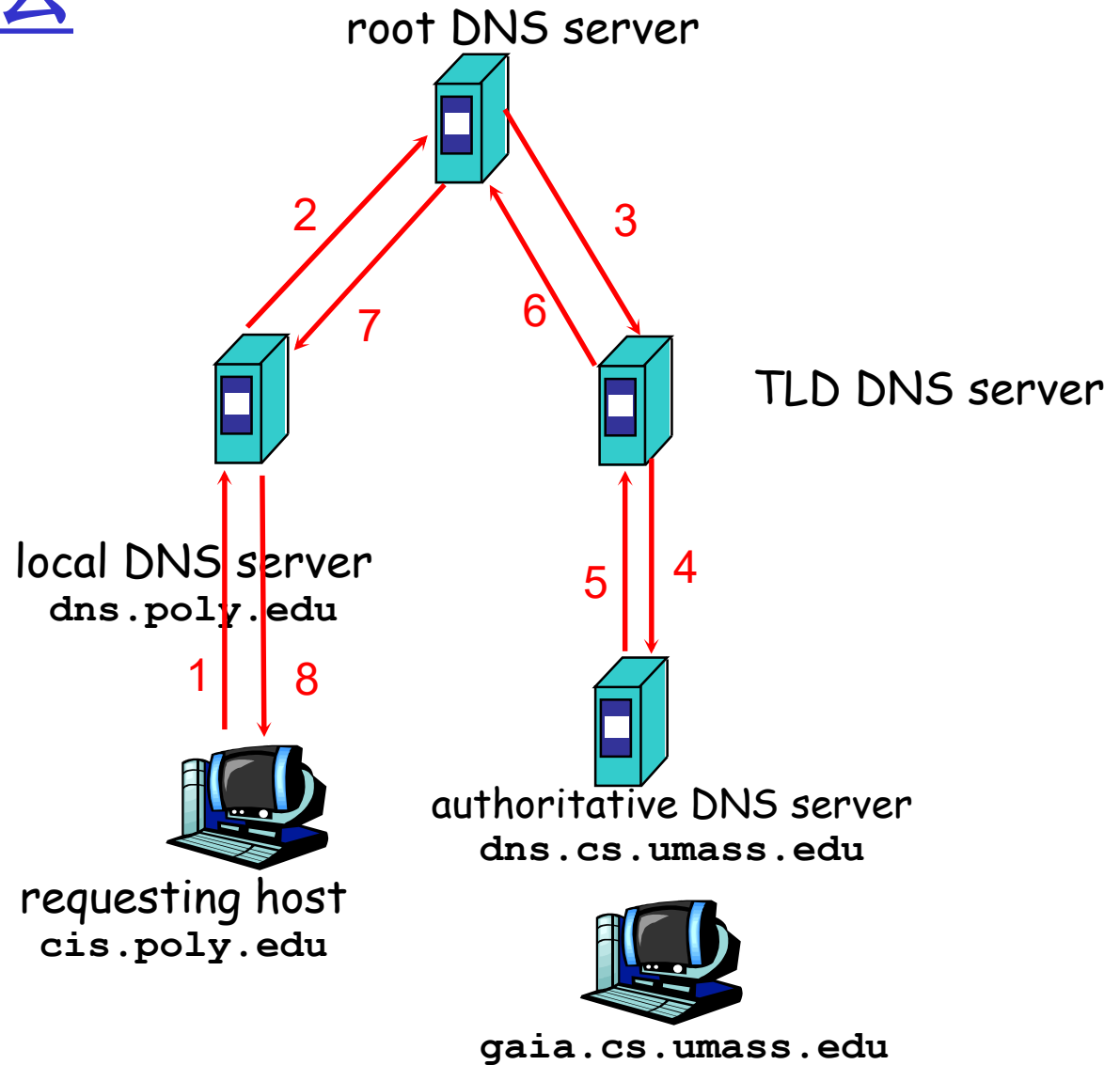
DNS查询方法

递归查询:

- ❑ 名字解析的负担交给被查询的名字服务器
- ❑ 被查询的名字服务器负载重?

迭代查询:

- ❑ 被查询的名字服务器 回复可以被查询的名字服务器的IP地址
- ❑ “我不知道它的名字, 但是可以问服务器”



DNS缓存和记录更新

- 一旦名字服务器获得**DNS**映射，它将缓存该映射到局部内存
 - 服务器在一定时间后将丢弃缓存的信息
 - 本地**DNS**服务器可以缓存**TLD**服务器的**IP**地址
 - 因此根**DNS**服务器不会被经常访问
- **IETF**设计的动态更新/通报机制
 - **RFC 2136**

DNS记录

DNS: 存储资源记录(RR)的分布式数据库

RR 格式: (name, value, type, ttl)

□ Type=A

○ name = 主机名

○ value = IP地址

(www.uestc.edu.cn, 202.112.14.178, A) 存放在dns.uestc.edu.cn上

(dns.uestc.edu.cn, 202.112.14.151, A) 存放在.edu.cn域名服务器上

□ Type=NS

○ name = 域名 (如foo.com, uestc.edu.cn)

○ value = 该域权威名字服务器的主机名

(www.uestc.edu.cn, dns.uestc.edu.cn, NS) 存放在.edu.cn域名服务器上

DNS记录

□ Type=CNAME

- name = 主机别名

`www.ibm.com`的真名为`servereast.backup2.ibm.com`

- value = 真实的规范主机名

- 例：多个域名指向同一个服务器IP

主机为`host.myweb.com`,同时提供WWW和MAIL服务, 可设置两个别名`www.myweb.com`和`mail.myweb.com`

(`www.myweb.com`, `host.myweb.com`, CNAME)

(`mail.myweb.com`, `host.myweb.com`, CNAME)

□ Type=MX (Mail Exchange)

- name = 邮件服务器的主机别名

- value = 邮件服务器的真实规范主机名

`test@uestc.edu.cn`

(`uestc.edu.cn`, `mail.uestc.edu.cn`, MX)

DNS协议，消息

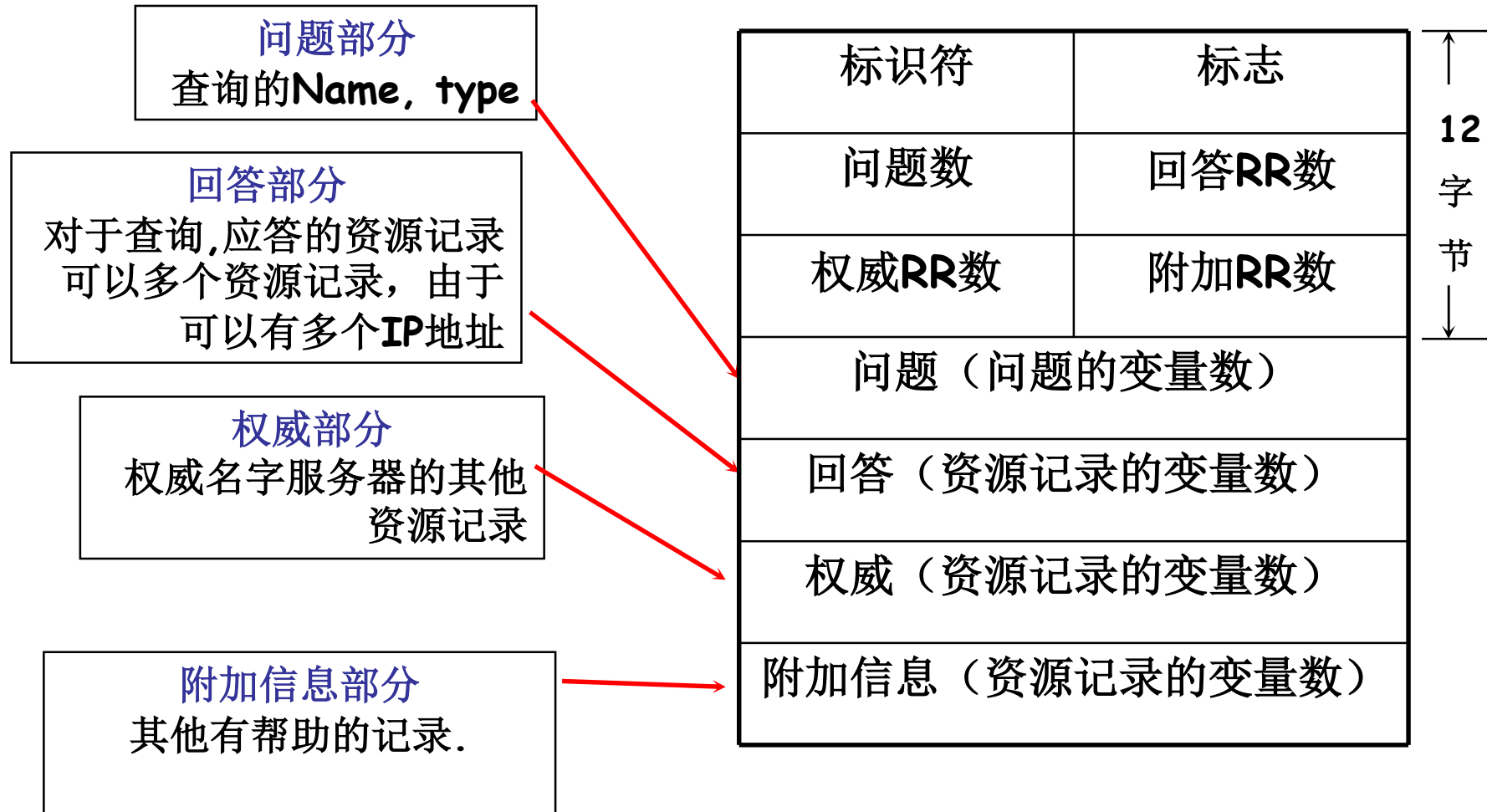
DNS协议：查询报文与应答报文，但具有同样的报文格式

报文头部

- ❑ 标识符：16位，查询和应答报文使用相同的标识符
- ❑ 标志：有若干个标志构成，分别标识不同的功能
 - 查询/应答—0/ 1
 - 查询希望是/非递归查询—1/0
 - 应答可/否获得(支持)递归查询—1/0
 - 应答是/否来自权威名字服务器—1/ 0

标识符	标志	↑ 12 字 节 ↓
问题数	回答RR数	
权威RR数	附加RR数	
问题（问题的变量数）		
回答（资源记录的变量数）		
权威（资源记录的变量数）		
附加信息（资源记录的变量数）		

DNS协议，消息



在DNS数据库中插入记录

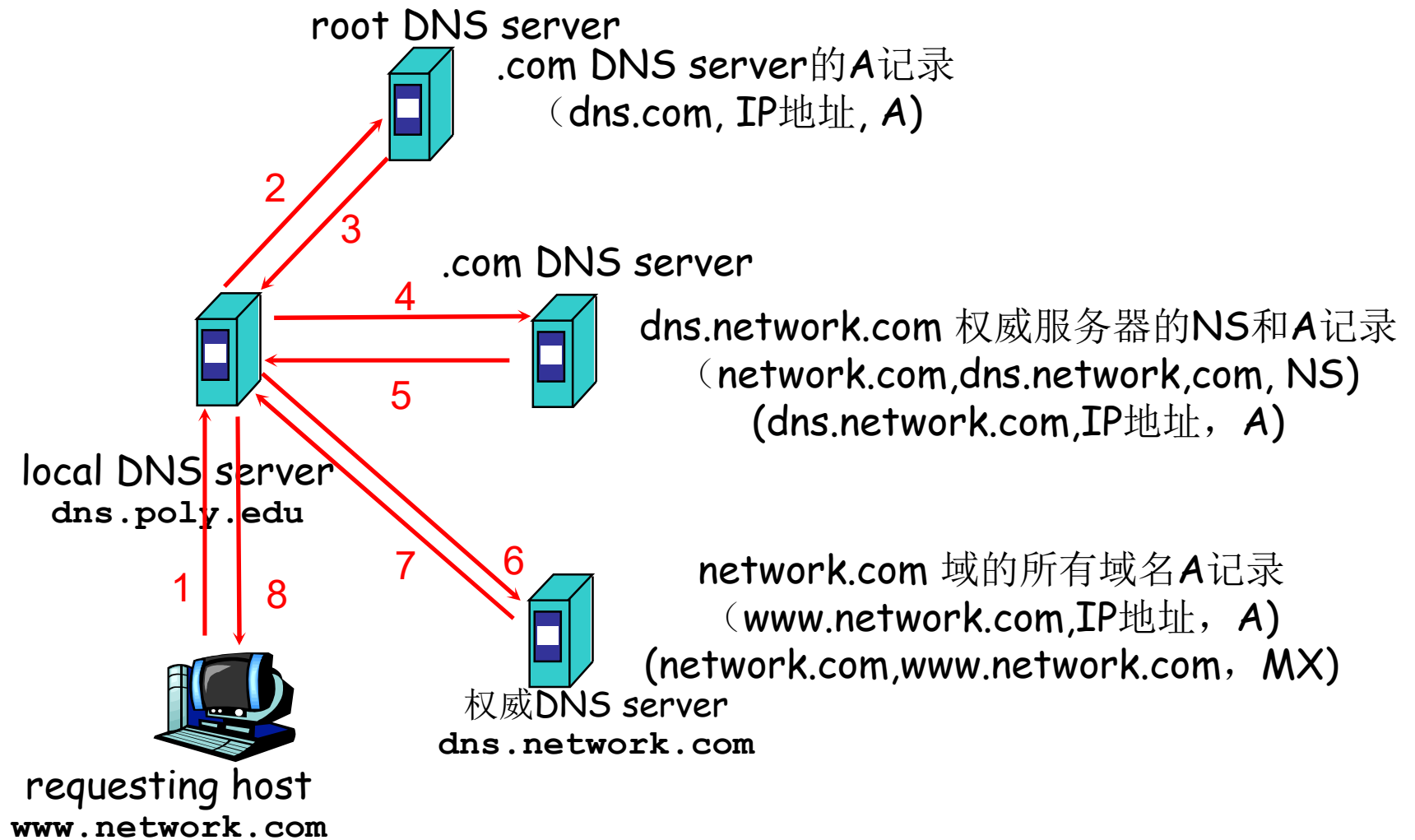
例子：刚刚创建一个“网络乌托邦”公司

- ❑ 如果你想在注册登记机构注册你的域名**network.com**，则
 - 需要提供你自己的基本权威**DNS**服务器和辅助权威**DNS**服务器的名字和**IP**地址
 - 该注册登记机构将下列两条资源记录插入注册机构的**DNS**系统中：

(**network.com**, **dns.network.com**, **NS**)

(**dns.network.com**, **212.212.212.1**, **A**)

- ❑ 如果你想建立一个网站，则可以将网址**www.network.com**以类型**A**的方式记录到你的权威**DNS**服务器**dns.network.com**中。
- ❑ 如果你想建一个邮件服务器，则可以将**mail.network.com**以类型**MX**的方式记录到你的权威**DNS**服务器**dns1.network.com**中。
- ❑ 人们怎样得到你的**Web**站点的**IP**地址呢？



第二章：复习大纲

- ❑ 网络应用程序体系结构
- ❑ Web应用和HTTP协议
 - 基本术语（网页、URL等）
 - HTTP的特性及其区别（无状态、非持久和持久等）
 - 请求和响应报文
 - COOKIE技术
 - Web缓存
- ❑ 文件传输协议FTP：两种连接
- ❑ 电子邮件：组成及其使用的协议
- ❑ DNS的功能和实现

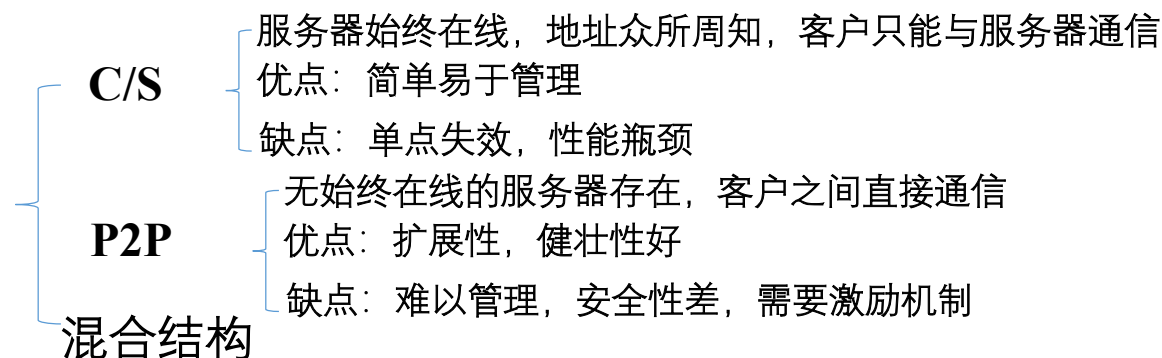
小结

- ❑ 应用程序体系结构
 - 客户机/服务器
 - P2P
 - 混合
- ❑ 应用服务器需要的服务
 - 可靠，带宽，延时
- ❑ 网络运输层协议
 - 面向连接，可靠：TCP
 - 不可靠，数据报：UDP
- ❑ 特殊协议：
 - HTTP
 - FTP
 - SMTP, POP, IMAP
 - DNS
- ❑ 套接字编程

第二章 应用层

- 1.应用层原理

- 应用体系结构*



- 确定进程：端口号，对进程的标识
 - 确定应用需求：可靠性、时延、带宽、安全
 - 选用互联网服务：TCP、UDP
 - 确定应用协议：报文类型、报文语法、语义、同步

2.HTTP协议

- WEB应用
- C/S结构
- Web服务器缺省（默认）端口号80
- TCP服务
- 无状态协议：服务器不保存客户之前的状态信息
- 两种工作模式*
 - 非持久HTTP连接：每个TCP连接上只传送一个对象
 - 持久HTTP连接
 - 不带流水线的持久HTTP连接：URL对象请求在前一请求的响应收到后才能发出
 - 带流水线的持久HTTP连接：多个URL对象请求无需等待之前的响应到达即可发出
- HTTP协议（ASCII文本）
 - 请求报文：请求行、首部行、空行、（实体）
请求方法GET、POST、HEAD，常见首部
 - 应答报文：状态行、首部行、空行、（实体）
常见响应状态

2.HTTP协议

- Cookies技术

- 服务器通过响应报文中的首部行set-cookie给用户cookie信息，并保存在后端数据库
- 浏览器通过请求报文中的首部行cookie表明身份，服务器访问后端数据库验证
- 给网上冲浪带来便利、牺牲了隐私

- Web缓存技术

- 减少响应时间，减少接入链路流量和Internet流量
- 位于客户端和起始服务器之间
- 即作为服务器（若客户请求URL对象命中）又可作为客户机（未命中）
- 命中率：客户端请求的URL对象在web缓存中存在的概率

- 条件GET

3.FTP协议

- 文件传输协议
- C/S模式
- TCP服务，FTP服务器缺省端口号21（持久连接）
- 带外控制*
 - 服务器端口21建立的TCP连接为控制连接，传送控制信息，整个会话期间保持连接
 - 传送数据信息时，服务器端口20将与客户端建立另一个TCP连接，传送完成后断开
- FTP服务器是有状态协议，维护用户状态
- FTP协议
 - 基于ASCII文本的命令应答式

4.SMTP协议*

- 简单邮件传送协议
- C/S模式
- TCP服务，SMTP服务器缺省端口号25（持久连接）
- 直接传送：发送方SMTP服务器将邮件直接传送到接收方SMTP服务，不经由第三方SMTP服务器
- SMTP协议
 - 基于ASCII文本的命令应答式
 - 三个阶段：握手、邮件传输、结束
 - 推（push）协议，HTTP是拉（pull）协议
 - 邮件消息格式:信头、空行和信体
 - 不足：邮件消息必须是7bit ASCII码，不能传送非ASCII的二进制对象，长度限制
- MIME协议：增加了邮件首部，定义了邮件内容格式和传送编码
 - 没有改动或替代SMTP
 - 对SMTP功能的扩展，使之能传送任何内容

5. POP3协议

- 邮局协议：从服务器获取邮件信息
- C/S模式
- TCP服务，POP3服务器缺省端口号110（持久连接）
- POP3协议
 - 基于ASCII文本的命令应答式
 - 三个阶段：身份认证、传输阶段、更新阶段
- POP3是无状态协议
- 两种模式：download and delete, download and keep
- 其他邮件访问协议
 - IMAP协议：有状态协议，服务器使用143端口
 - 允许用户在服务器的文件夹中管理邮件
- 基于Web的邮件访问协议

6.DNS协议

- DNS系统
 - 分布式的数据库
 - 提供主机名到IP地址的转换
 - 主机别名服务
 - 邮件服务器别名服务
 - 负载分配
- 集中式DNS的缺点
 - 单点故障
 - 巨大访问量
 - 维护成本大
 - 扩展性差
- C/S结构，UDP服务，DNS服务器端使用53

DNS协议

- 层次结构*
 - 根DNS
 - 顶级域名服务器
 - 权威域名服务器
- 本地DNS服务器，起着代理作用，严格来说不属于层次结构
- DNS查询方法*
 - 递归查询
 - 迭代查询
- DNS记录
 - A记录：主机名和IP地址的映射*
 - NS记录：域名和该域权威DNS服务器的映射
 - CNAME记录：主机别名和真实规范主机名的映射
 - MX记录：邮件服务器别名和邮件服务器真实规范主机名的映射
- DNS报文：查询报文和应答报文

本章作业

□ R2, R6, R15

□ P1, P3, P8, P9, P13