

计算机网络

■ 主讲：肖林



内容



- ④ 第1章 概述
- ④ 第2章 物理层
- ④ 第3章 数据链路层
- ④ 第4章 介质访问控制子层
- ④ 第5章 网络层
- ④ 第6章 传输层
- ④ **第7章 应用层**

第7章 应用层



- ④ DNS域名系统
- ④ 电子邮件
- ④ 万维网
- ④ 流式音视频
- ④ 内容分发

7.1 DNS域名系统



- DNS由来
- DNS查找过程
- DNS名字空间与层次结构
- DNS查询与应答
- 名字解析
- 域名服务器
- DNS隐私

DNS由来



■ ARPANET

- Hosts.txt

几百台主机，集中管理主机名称

■ DNS

- 1983，基于域的层次命名方案

见6.1.4套接字编程实例：Internet文件服务器，图6.6程序代码（P392）

```
struct hostent *h;                /* info about server */
struct sockaddr_in channel;        /* holds IP address */

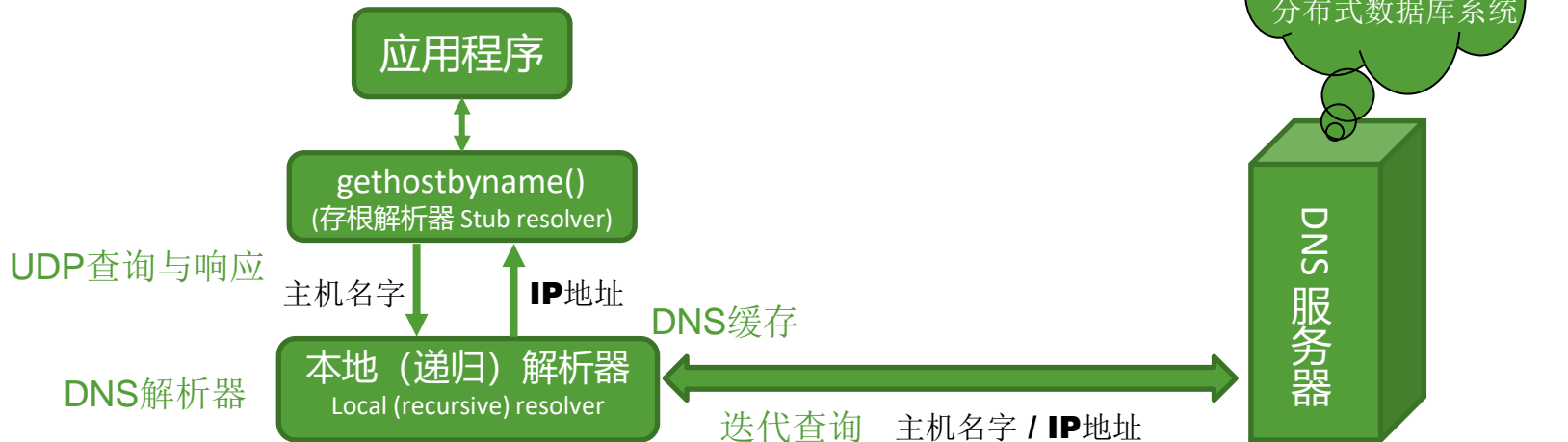
if (argc != 3) fatal("Usage: client server-name file-name");
h = gethostbyname(argv[1]);      /* look up host's IP address */
```

gethostbyname()

DNS查找过程

■ DNS查找

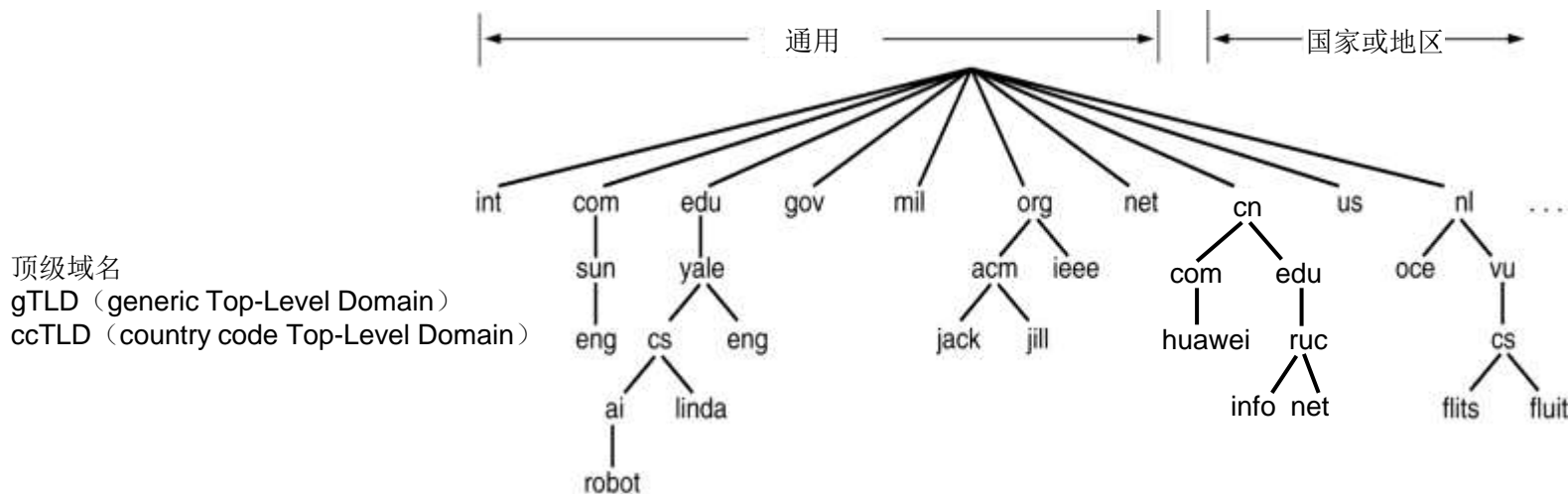
- 递归查询与迭代查询
- DNS缓存: Time_to_live 缓存记录时间
- QNAME最小化, 缓存区中毒 (DNS植入), TCP查询DNS



DNS名字空间与层次结构



■ 因特网名字与数字地址分配机构ICANN



- 通用域名由20世纪80年代推出，国家或地区域名由ISO3166文档定义
- 2010推出非拉丁字母的国家化域名，2011开放顶级域名（成本约\$20万）

DNS名字空间示例

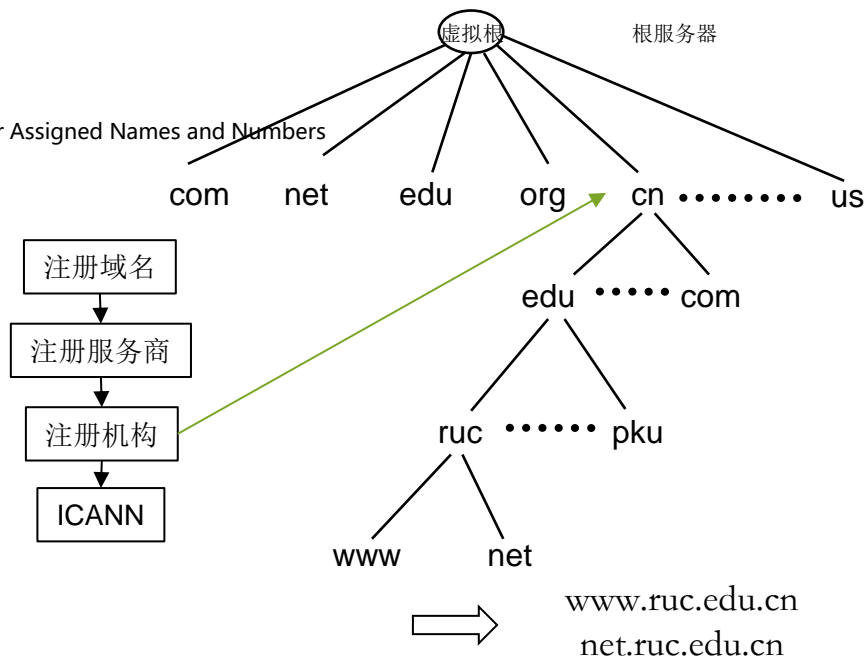


■ 域名的命名与特点

互联网名称与数字地址分配机构：Internet Corporation for Assigned Names and Numbers

中国互联网络信息中心：<https://www.cnnic.cn>

1985	域名	意义	域名	意义	2001
	com	公司企业	aero	航空运输	
	net	网络服务机构	biz	公司	
	edu	教育机构	coop	合作	
	org	非赢利组织	pro	专业	
2002	gov	美国政府部门	museum	博物馆	2010
	mil	美国军事部门	info	信息服务	
	int	国际组织	nom	个人	
	cn	中国	jp	日本	
	us	美国	uk	英国	



绝对域名：www.huawei.com.

相对域名：www.huawei.com
www.huawei.com.cn

域名不区分大小写，
总长≤255，组成部分≤63

DNS查询与应答



■ DNS查询

- A记录查询：域名映射为IP地址
- DNSBL：DNS黑名单查询

■ DNS查询扩展

- 扩展DNS客户子网（Extended DNS Client Subnet）
- 通过获取查询客户IP地址，可以将域名映射到附近的一个备份的重复服务器上

■ DNSSEC

- 为防止DNS安全威胁，要求DNS名称服务器应答携带数字签名

资源记录



■ 资源记录格式

五元组: Domain_name Time_to_live Class Type Value

- 域名
 - ◆ 记录所在域
- 生存期（保留时间）
- 类别
 - ◆ IN: internet信息
- 类型
- 值

类 型	意 义	值
SOA	授权起始	本区域参数
A	主机IPv4地址	32bit
AAAA	主机IPv6地址	128bit
MX	邮件交换	优先级, 愿意接受邮件
NS	域名服务器	本域服务器名
CNAME	规范名	域名(用于标识别名)
PTR	指针	IP地址别名(逆向查询)
SPF	发送者政策框架	邮件发送政策文本编码
SRV	服务	提供服务的主机
TXT	文本	任意描述文本

资源记录示例

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA    star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.      86400  IN  TXT    "Divisie Wiskunde en Informatica."
cs.vu.nl.      86400  IN  TXT    "Vrije Universiteit Amsterdam."
cs.vu.nl.      86400  IN  MX     1 zephyr.cs.vu.nl.
cs.vu.nl.      86400  IN  MX     2 top.cs.vu.nl.

flits.cs.vu.nl. 86400  IN  HINFO  Sun Unix
flits.cs.vu.nl. 86400  IN  A      130.37.16.112
flits.cs.vu.nl. 86400  IN  A      192.31.231.165
flits.cs.vu.nl. 86400  IN  MX     1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX     2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX     3 top.cs.vu.nl.
www.cs.vu.nl.   86400  IN  CNAME  star.cs.vu.nl
ftp.cs.vu.nl.   86400  IN  CNAME  zephyr.cs.vu.nl

rowboat         IN  A      130.37.56.201
                IN  MX     1 rowboat
                IN  MX     2 zephyr
                IN  HINFO  Sun Unix

little-sister   IN  A      130.37.62.23
                IN  HINFO  Mac MacOS

laserjet        IN  A      192.31.231.216
                IN  HINFO  "HP Laserjet III Si" Proprietary
```

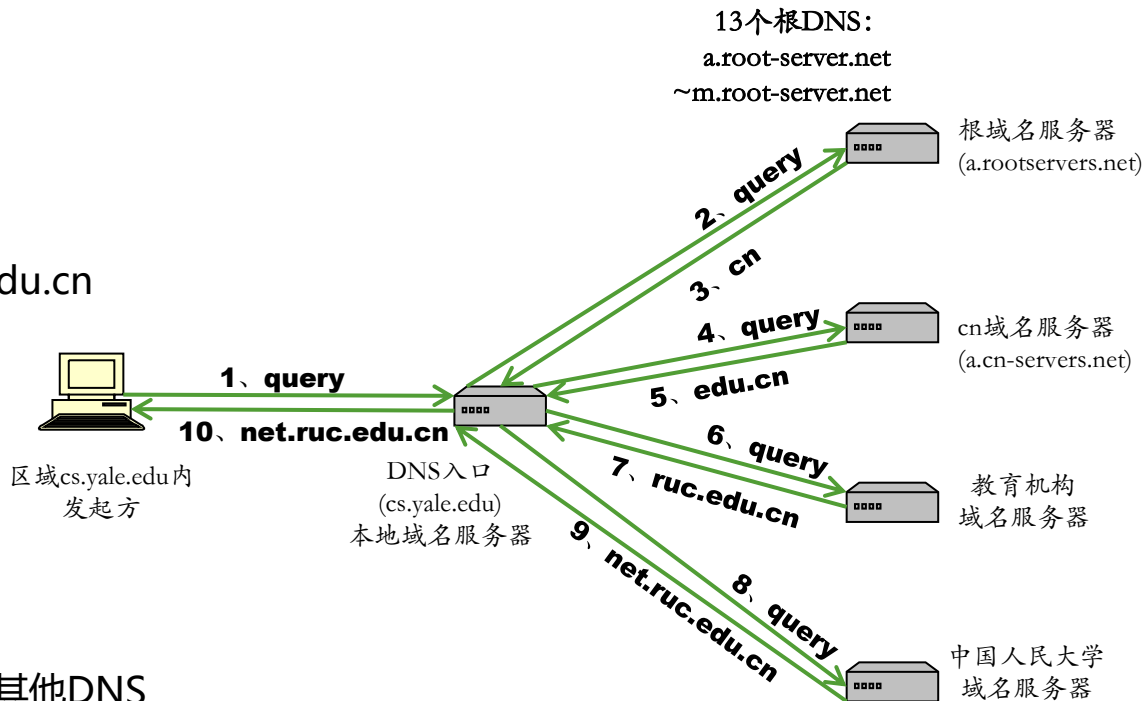
域名解析

■ 域名解析机制

- 递归查询: 1,10
- 迭代查询: 2-9

■ 缓存

- 访问: net.ruc.edu.cn



■ 传输层协议

- UDP
- 重复查询及尝试其他DNS

域名服务器



■ 域名区域 (zones) 与名字服务器

- 名字空间划分为不重叠的区域 (分布式服务)
- 每个区域关联多个域名服务器 (主、辅) ; 域名服务器可以在其管辖的区域外

- 区域 ruc.edu.cn

- ◆ 网络实验室能否成为独立区域?

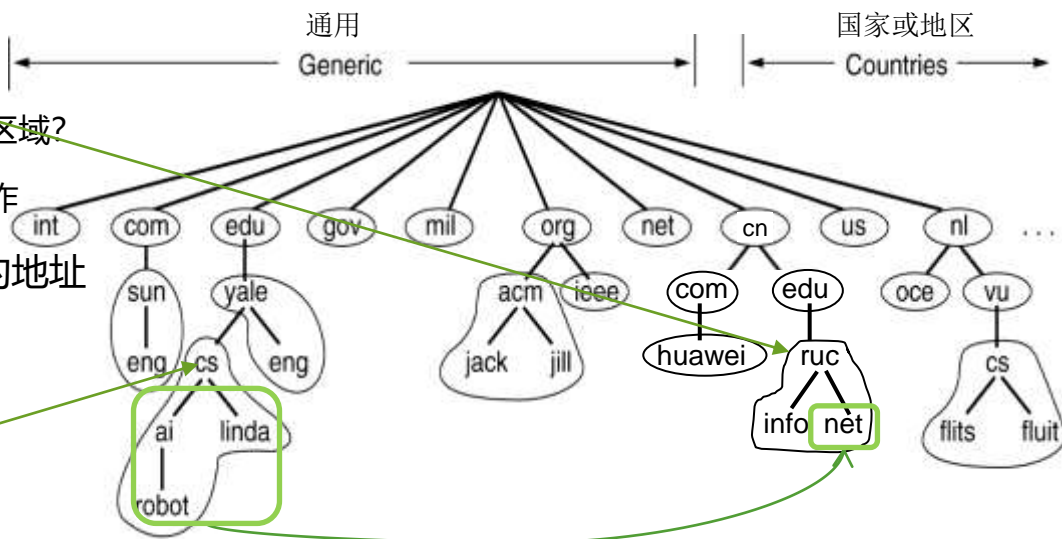
- ◆ 设立独立名字服务器的工作

- 域名解析: 查询域名对应的地址

- ◆ 访问 ai.cs.yale.edu

- ◆ 查询交给本地域名服务器

- ◆ 返回权威记录



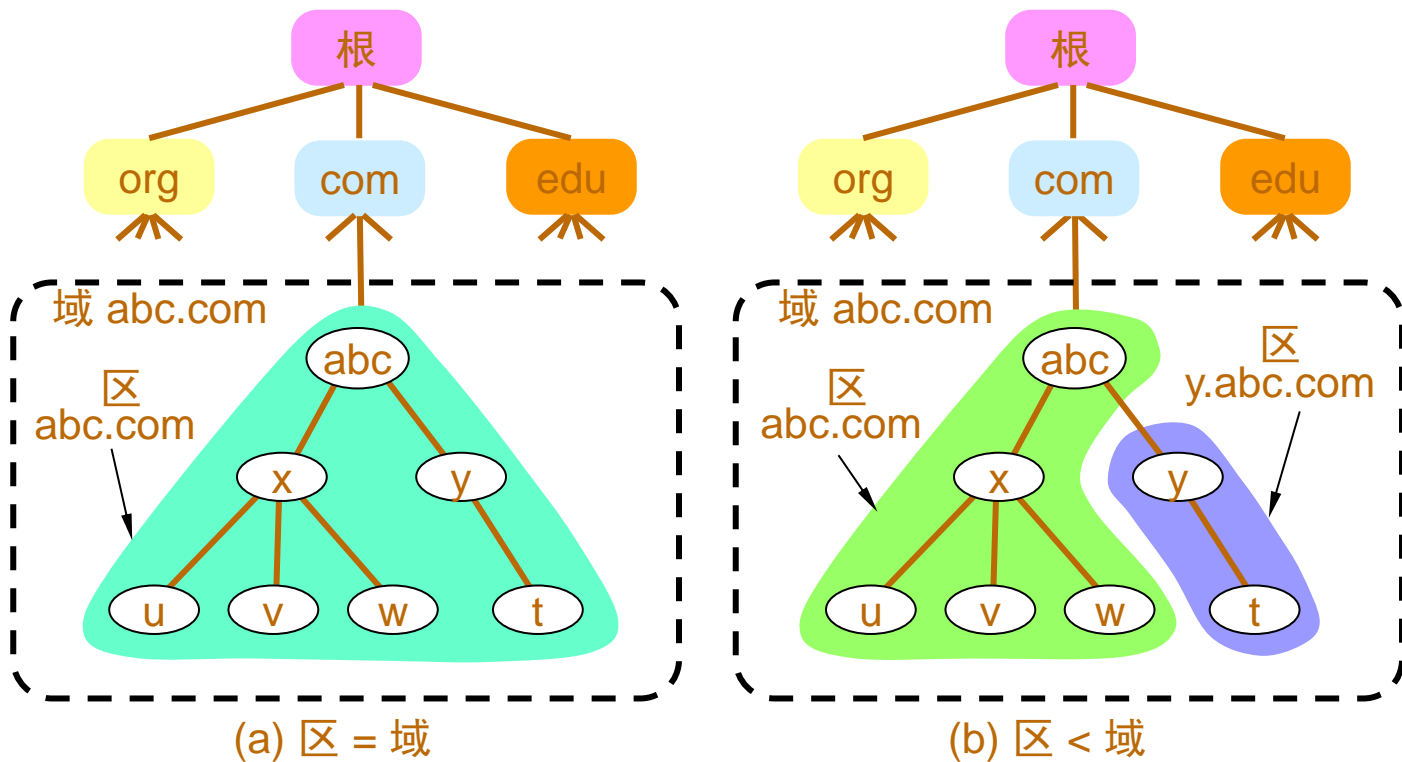
域名服务器说明



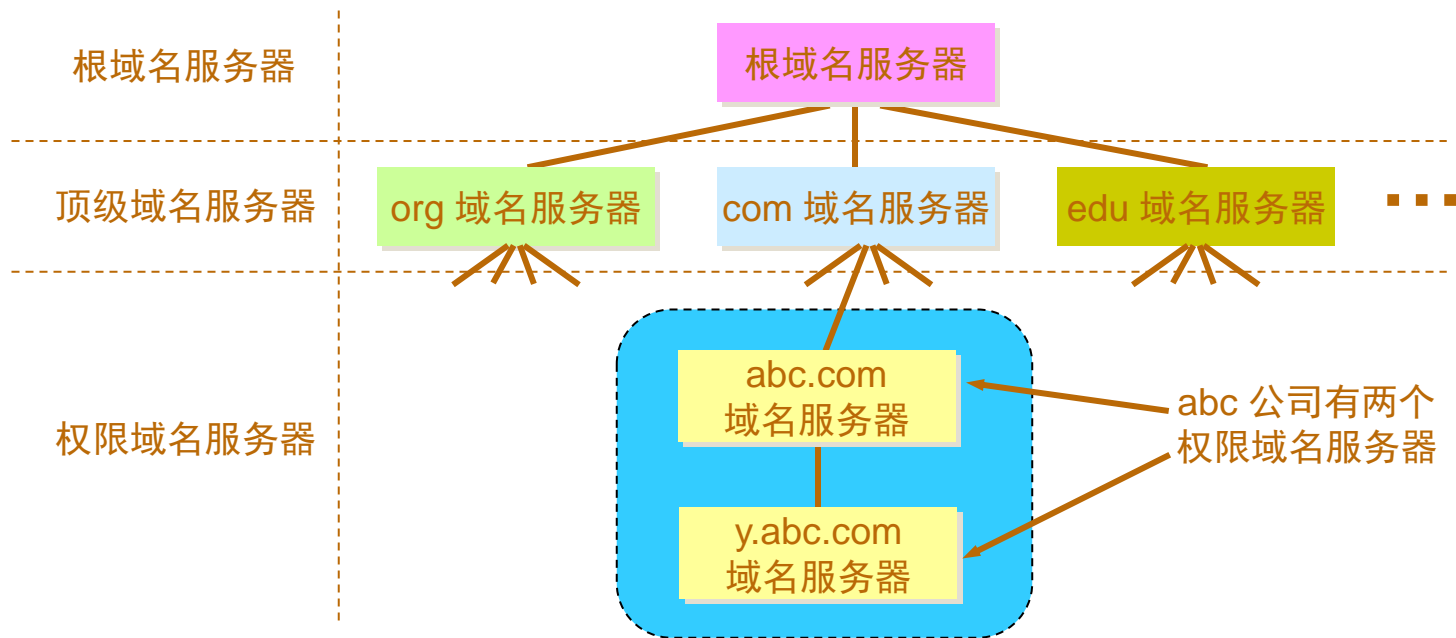
- 一个服务器所负责管辖的（或有权限的）范围叫做**区(zone)**。
- 各单位根据具体情况来划分自己管辖范围的区。但在一个区中的所有节点必须是能够连通的。
- 每一个区设置相应的**权限域名服务器**，用来保存该区中的所有主机的域名到IP地址的映射。
- DNS 服务器的管辖范围不是以“域”为单位，而是以“区”为单位。



区的不同划分方法举例



树状结构的 DNS 域名服务器





域名服务器类型

- 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

根域名服务器——最高层次的域名服务器



- 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）



根域名服务器共有 13 套装置 (不是 13 个机器)

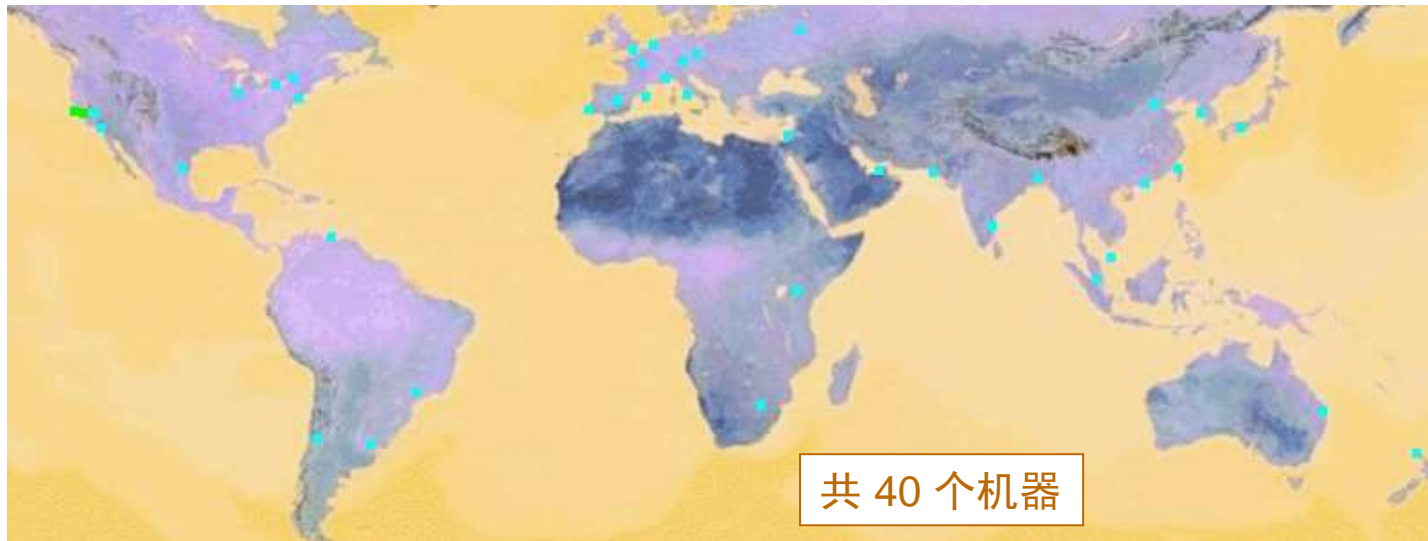
■ 这些根域名服务器相应的域名分别是

A.root-servers.net	198.41.0.4	美国
B.root-servers.net	192.228.79.201	美国 (另支持IPv6)
C.root-servers.net	192.33.4.12	法国
D.root-servers.net	128.8.10.90	美国
E.root-servers.net	192.203.230.10	美国
F.root-servers.net	192.5.5.241	美国 (另支持IPv6)
G.root-servers.net	192.112.36.4	美国
H.root-servers.net	128.63.2.53	美国 (另支持IPv6)
I.root-servers.net	192.36.148.17	瑞典
J.root-servers.net	192.58.128.30	美国
K.root-servers.net	193.0.14.129	英国 (另支持IPv6)
L.root-servers.net	198.32.64.12	美国
M.root-servers.net	202.12.27.33	日本 (另支持IPv6)

■ 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

举例：根域名服务器 f 的地点分布图



- 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。



顶级域名服务器（即 TLD 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

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

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

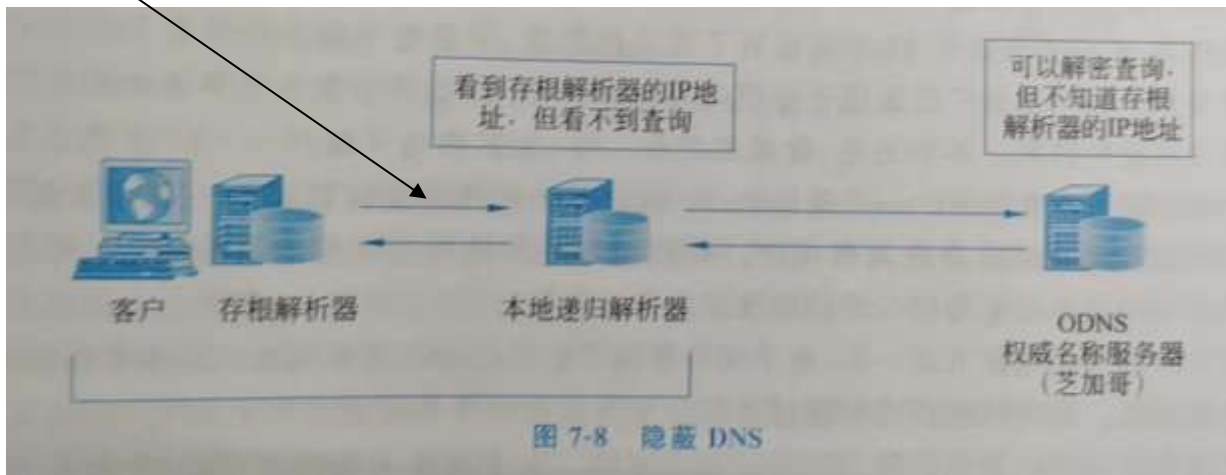
提高域名服务器的可靠性



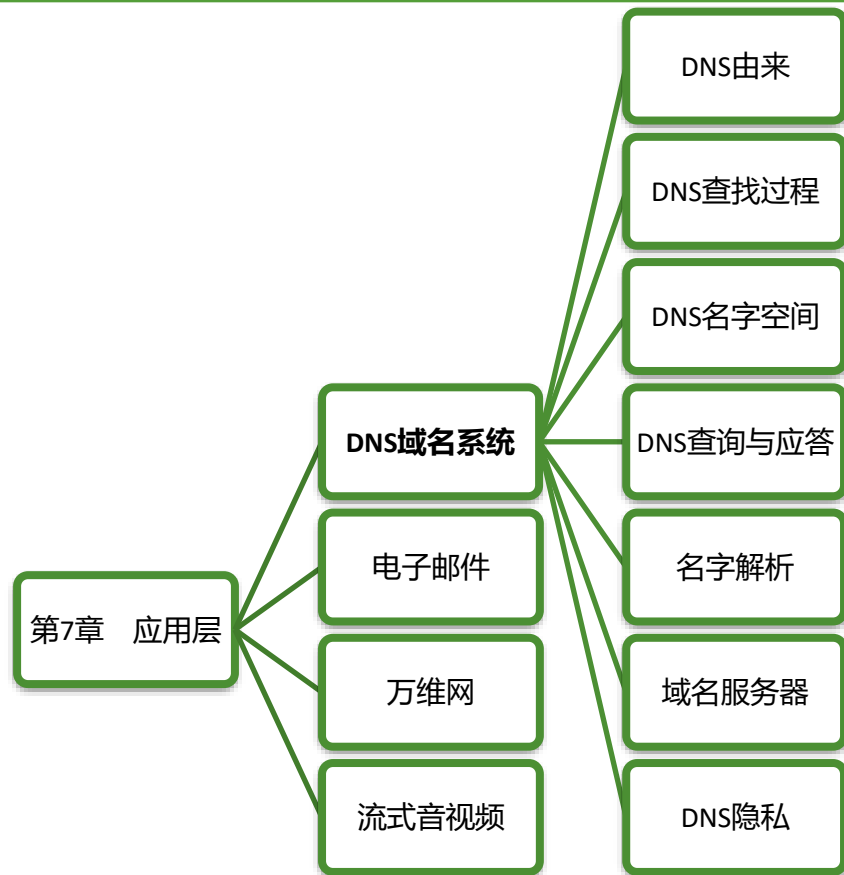
- DNS 域名服务器都把数据复制到几个域名服务器来保存，其中的一个是主域名服务器，其他的是辅助域名服务器
- 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断
- 主域名服务器定期把数据复制到辅助域名服务器中，而更改数据只能在主域名服务器中进行。这样就保证了数据的一致性

DNS隐私

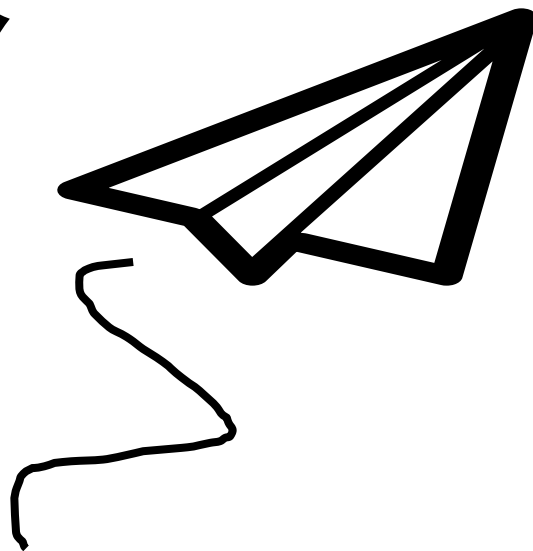
- DNS查询与应答皆为明码
 - 网吧系统管理员，物联网中的敏感设备和信息
- 加密DNS查询与应答



本章导航与要点



本节课程结束



7.2 电子邮件



- 体系结构和服务
- 用户代理
- 邮件格式
- 邮件传送
- 最后传递

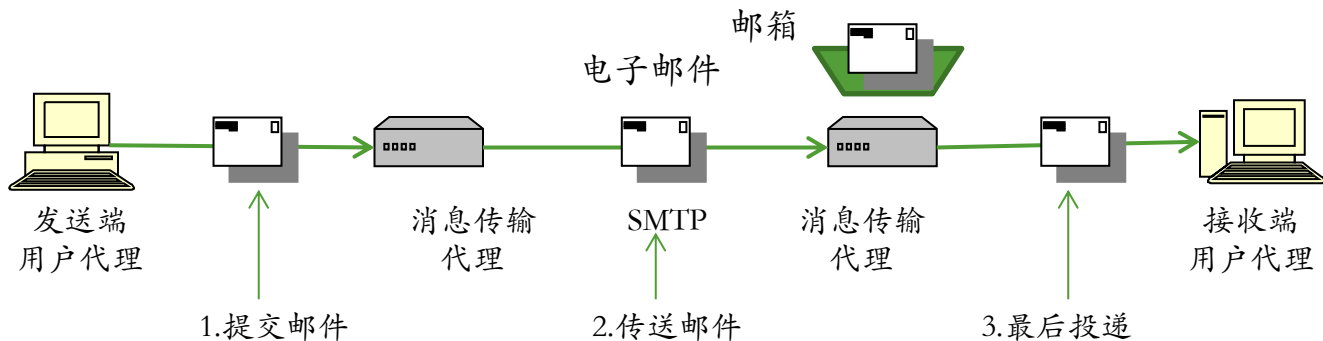
电子邮件的一些标准



- 发送邮件的协议：SMTP
- 读取邮件的协议：POP3 和 IMAP
- 邮件格式：RFC822-RFC5322-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

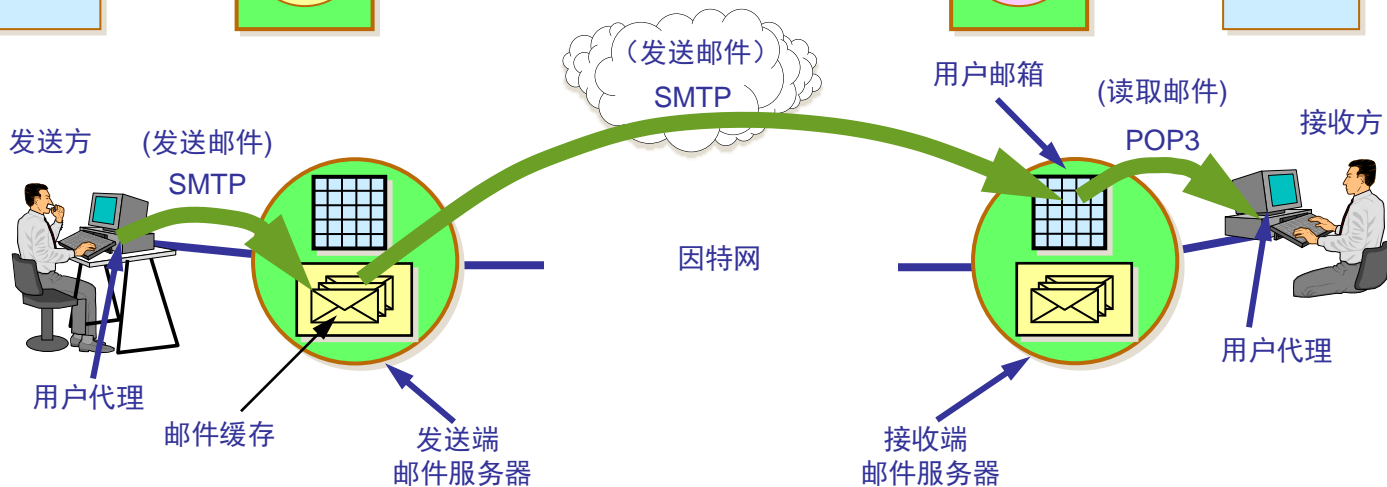
体系结构和服务

- 子系统：用户代理、消息传输代理
- 功能与协议：编写、显示、处理；简单邮件传输协议SMTP
- 其他：邮箱、收件人列表；信封、头部、主体





电子邮件的最主要的组成构件



用户代理 UA



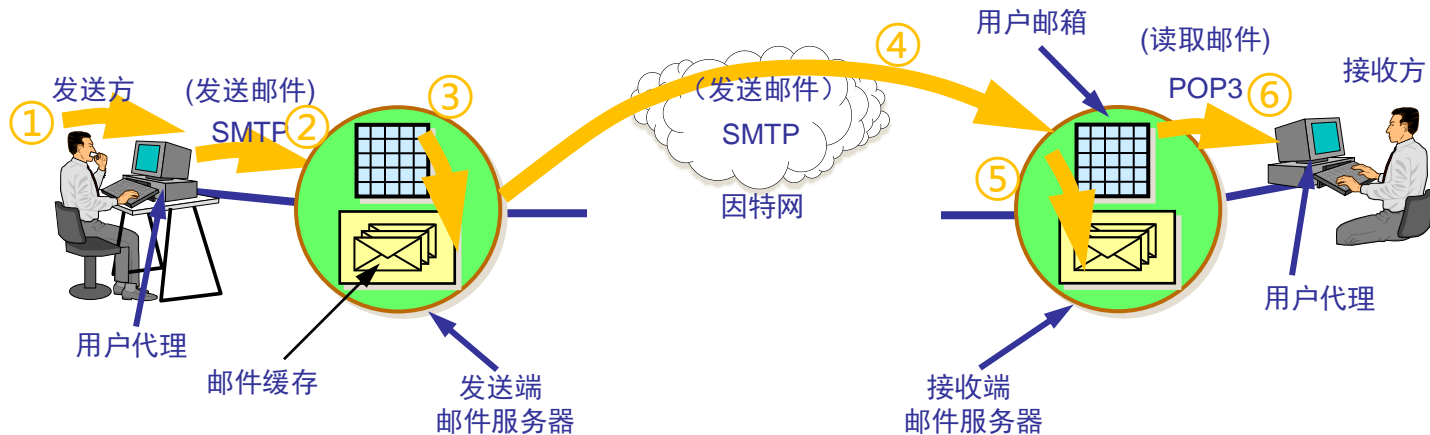
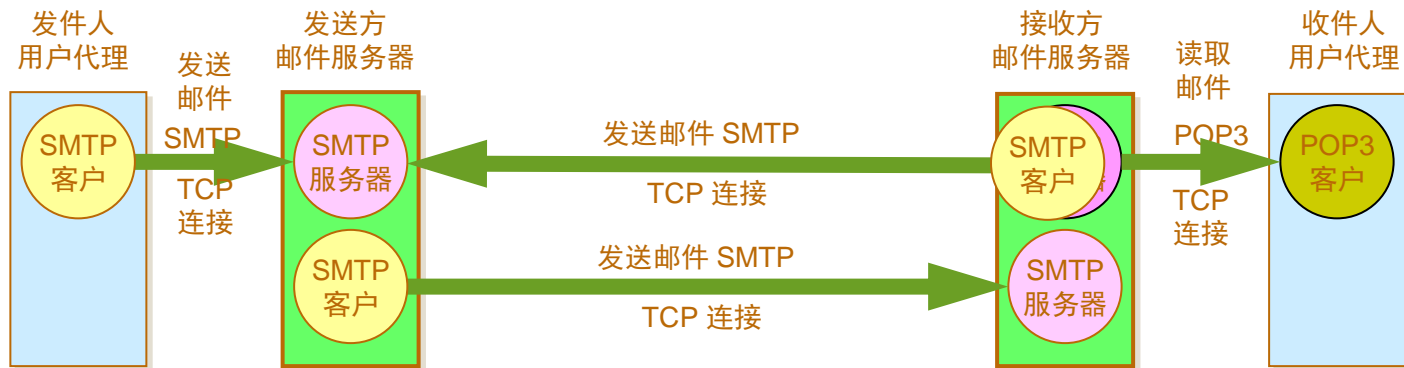
- 用户代理 UA (User Agent)就是用户与电子邮件系统的接口，是电子邮件客户端软件
- 用户代理的功能：撰写、显示、处理和通信
- 邮件服务器的功能是发送和接收邮件，同时还要向发信人报告邮件传送的情况（已交付、被拒绝、丢失等）
- 邮件服务器按照客户服务器方式工作。邮件服务器需要使用发送和读取两个不同的协议

应当注意



- 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

发送和接收电子邮件



发送和接收电子邮件的几个重要步骤



- ❶ 发件人调用 PC 机中的用户代理撰写和编辑要发送的邮件
- ❷ 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
- ❸ SMTP 服务器把邮件临时存放在邮件缓存队列中，等待发送
- ❹ 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
- ❺ 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
- ❻ 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

邮件格式

■ RFC 5322 头域 —— Internet邮件格式

- 早期为RFC 822

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

■ MIME —— 多用途Internet邮件格式

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	E-mail address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display



电子邮件的信息格式

- 电子邮件由信封(envelope)和内容(content)两部分组成
- 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



邮件内容的首部

- “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

多用途因特网邮件扩充 MIME



■ MIME (Multipurpose Internet Mail Extensions) 概述

● SMTP 有以下缺点：

- ◆ 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 不能传送可执行文件或其他的二进制对象
- ◆ 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

MIME 和 SMTP 的关系

- 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



MIME 主要包括三个部分



- 5 个新的邮件首部字段，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

邮件首部	含义
MIME-Version	标志 MIME 的版本
Content-Description	这是可读字符串，说明此邮件是什么
Content-Id	邮件的唯一标识符
Content-Transfer-Encoding	传送时邮件主体如何编码打包
Content-Type	定义了邮件内容的数据类型和格式

有效的类型有：text, image, audio, video, applications, multipart和message

- 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

内容传送编码



- 最简单的编码就是 7 位 ASCII 码，而每行不能超过 1000 个字符
 - MIME 对这种由 ASCII 码构成的邮件主体不进行任何转换
- 另一种编码称为 8 位字符，遵循最大行长度
 - 这种编码方法适用于：所传送的数据中只有少量的非 ASCII 码
 - 可打印(quoted-printable)编码：超过 127 的字符用 “=” 加 2 个十六进制字符代替
- 对于任意的二进制文件，不遵循每行 1000 个字符的限制，可用 base64 编码
 - 每 24 位一组，每组 4 个 6 位单元，每个单元当作一个合法的 ASCII 码字符来发送
 - 每单元编码：0~25 用 A~Z 表示，26~51 用 a~z 表示，52~61 用 0~9 表示，62、63 用 + 和 / 表示；
“=” 和 “=” 分别表示最后一个组只有 1 或 2 个单元，回车换行将被忽略
- 最后，可用 “Content-Transfer-Encoding” 字段说明用户自定义的编码

内容类型



- MIME标准规定 Content-Type 的说明必须含有两个标识符，即内容类型(type)和子类型(subtype)，中间用 “/” 分开
 - 如 “Content-Type: video/mpeg”
- MIME 标准定义了 7 个基本内容类型和 15 种子类型

类型	子类型实例	描述
text	Plain,html,xml,css	不同格式文本内容
image	gif,jpeg,tiff	照片
audio	basic,mpeg,mp4	声音
video	mpeg,mp4,quicktime	影片
application	octet-stream,pdf,javascript,zip	应用程序产生的数据
message	http,rfc822	封装的邮件
multipart	mixed,alternative,parallel,digest	多个类型的组合
<u>model</u>	<u>vrml</u>	<u>3D模型 (目前很少应用)</u>

邮件传送

■ SMTP

- 简单邮件传输协议

使用简单的ASCII，易于开发、测试和调试；

如可以通过键盘发送命令，消息记录便于阅读等。

- 端口25

elinore@abc.com 发送邮件给 *arolyn@xyz.com*

```
S: 220 xyz.com SMTP service ready
C: HELO abcd.com
S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
S: 250 recipient ok
C: DATA
S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdghijklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdghijklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold>Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdghijklzxcvbnm
C: Content-Type: message/external-body;
C:   access-type="anon-ftp";
C:   site="bicycle.abcd.com";
C:   directory="pub";
C:   name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdghijklzxcvbnm
C:
S: 250 message accepted
C: QUIT
S: 221 xyz.com closing connection
```

简单邮件传送协议 SMTP



- SMTP 所规定的就是在两个相互通信的 SMTP 进程之间应如何交换信息
- 由于 SMTP 使用客户服务器方式，因此负责发送邮件的 SMTP 进程就是 SMTP 客户，而负责接收邮件的 SMTP 进程就是 SMTP 服务器
- SMTP 规定了 14 条命令和 21 种应答信息
 - 每条命令用 4 个字母组成，而每一种应答信息一般只有一行信息，由一个 3 位数字的代码开始，后面附上（也可不附上）很简单的文字说明



SMTP 通信的三个阶段

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

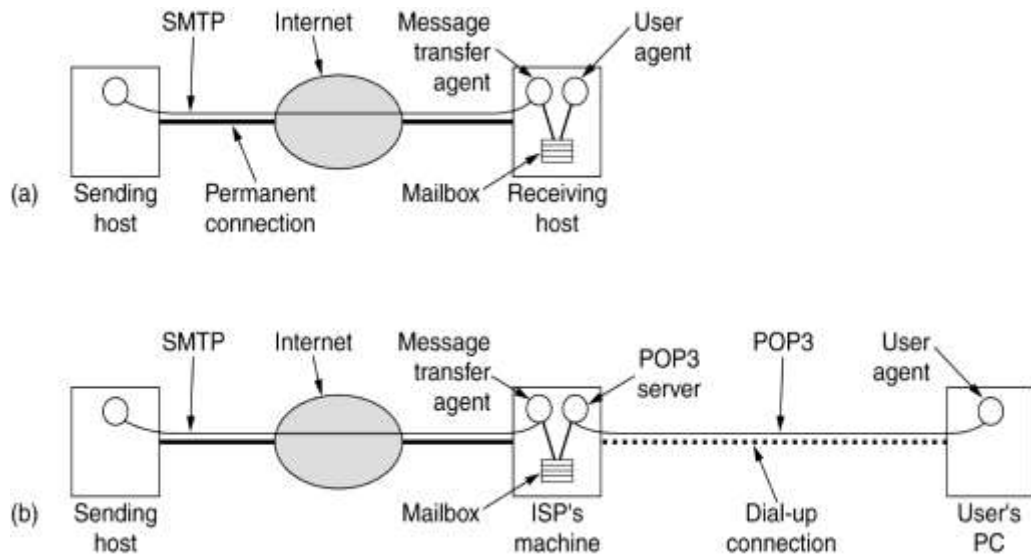
2. 邮件传送

3. 连接释放


邮件发送完毕后，SMTP 应释放 TCP 连接

最后的投递

- POP3——邮局协议
- IMAP——交互式电子邮件访问协议
- Webmail



POP3协议



- 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

IMAP 协议(Internet Message Access Protocol)



- 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

IMAP 的特点



- 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 机上，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

必须注意



- 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

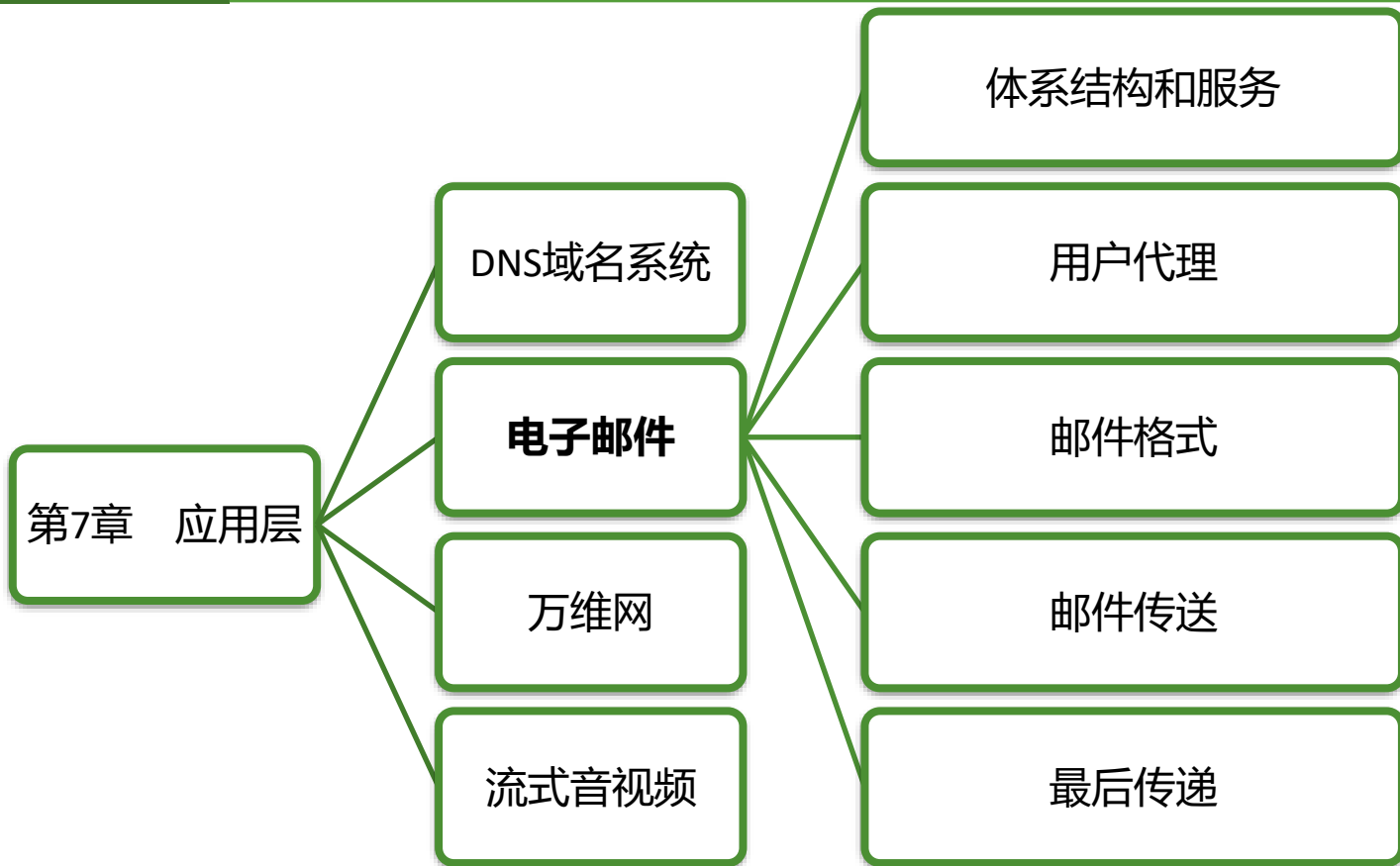


基于万维网的电子邮件

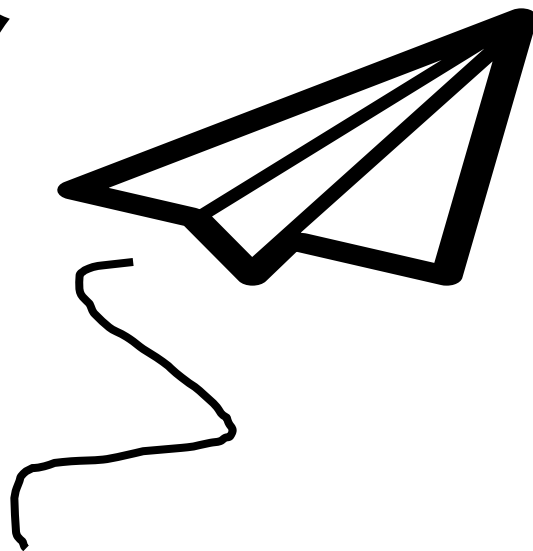
- 电子邮件从 A 发送到网易邮件服务器是使用 HTTP 协议
- 两个邮件服务器之间的传送使用 SMTP
- 邮件从新浪邮件服务器传送到 B 是使用 HTTP 协议



本章导航与要点



本节课程结束



7.3 万维网



- 体系结构概述
- 静态Web页面
- 动态Web页面和Web应用
- HTTP和HTTPS
- Web隐私

体系结构概述

■ HTTP



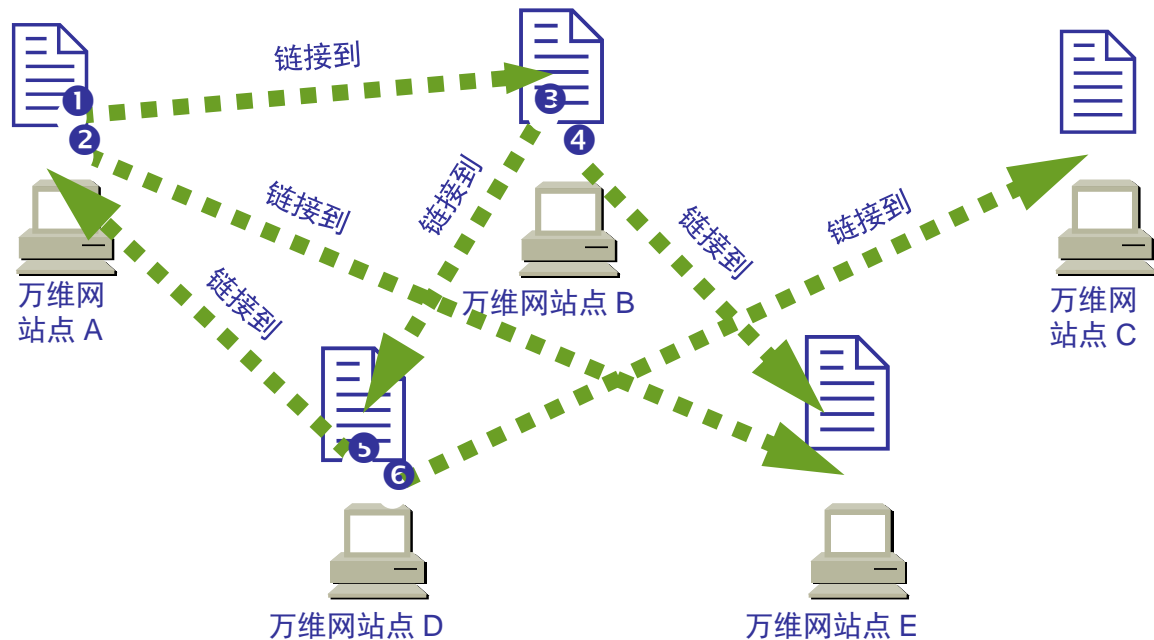
图 7-19 获取和渲染一个 Web 页面要向很多服务器发送 HTTP/HTTPS 请求



万维网概述

- 万维网 WWW (World Wide Web)并非某种特殊的计算机网络
- 起源于1989年的CERN（欧洲原子能研究中心）
用于粒子物理实验的报告、计划、绘图、照片和其他文档的协同展示
- 万维网是一个大规模的、联机式的信息储藏所
- 万维网用链接的方法能非常方便地从因特网上的一个站点访问另一个站点，从而主动地按需获取丰富的信息
- 这种访问方式称为 “链接”

万维网提供分布式服务





超媒体与超文本

- 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



万维网的工作方式

- 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)

万维网的问题



(1) 怎样标志分布在整个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) 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

(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

(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）

客户端



■ 统一资源定位符URL

- <http://www.ruc.edu.cn:80/index.html>

■ 超文本传输协议HTTP与常用的URL

名 字	用 于	例 子
http	超文本	http://www.ruc.edu.cn/index.html
https	安全的超文本	https://user.cmbchina.com
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	本地文件	file:///usr/suzanne/prog.c
mailto	发送电子邮件	mailto:JohnUser@acm.org
rtsp	流式媒体	rtsp://youtube.com/montypython.mpg
sip	多媒体呼叫	sip:eve@adversary.com
about	浏览器信息	about:plugins

统一资源定位符 URL



URL的格式

- 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

URL 的一般形式

- 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

<主机> 是存放资源的主机
在因特网中的域名

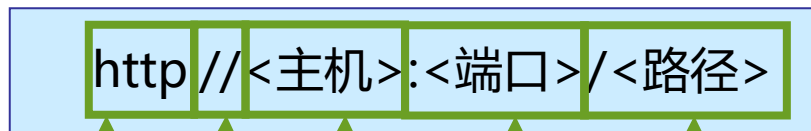
ftp —— 文件传送协议 FTP

http —— 超文本传送协议 HTTP

News —— USENET 新闻

使用 HTTP 的 URL

■ 使用 HTTP 的 URL 的一般形式



这表示使用 HTTP 协议

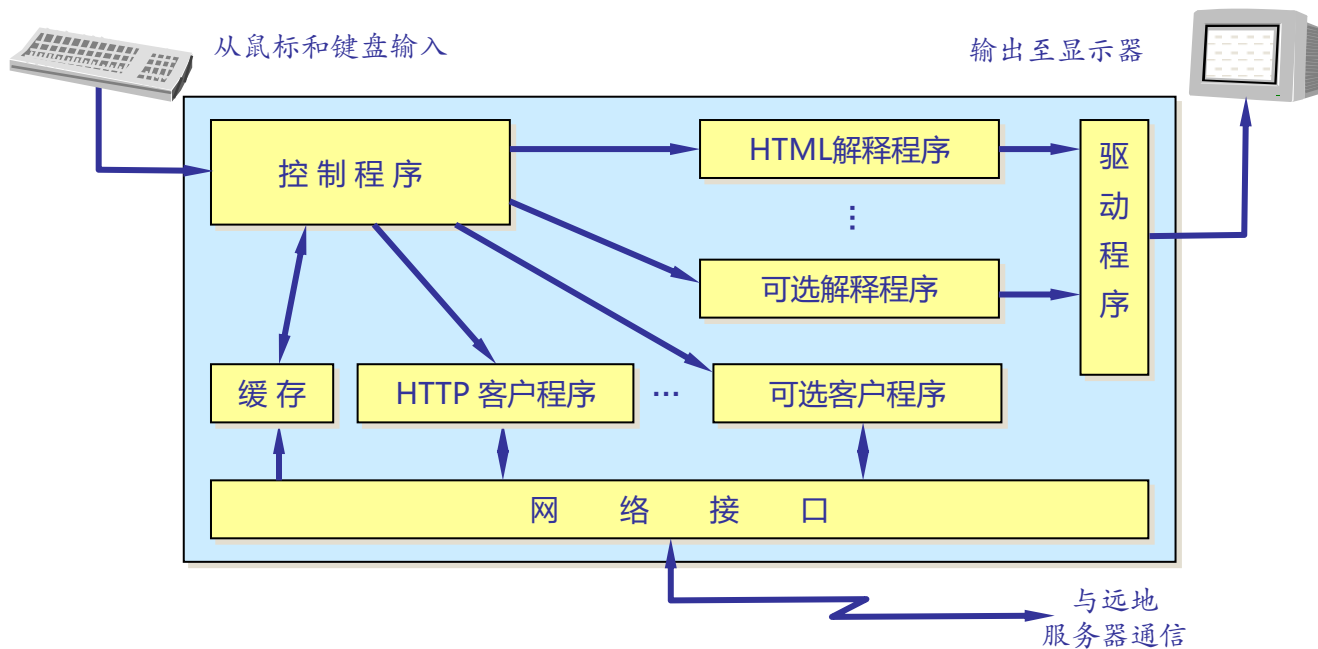
冒号和两个斜线是规定的格式

这里写主机的域名

HTTP 的默认端口号是 80，通常可省略

若再省略文件的<路径>项，则 URL 就指向 Web 服务器上的主页

浏览器的结构





浏览器的主要组成部分

- 浏览器有一组客户、一组解释程序，以及管理这些客户和解释程序的控制程序
- 控制程序是其中的核心部件，它解释鼠标的点击和键盘的输入，并调用有关的组件来执行用户指定的操作
- 例如，当用户用鼠标点击一个超链的起点时，控制程序就调用一个客户从所需文档所在的远地服务器上取回该文档，并调用解释程序向用户显示该文档

解释程序



- HTML 解释程序是必不可少的，而其他的解释程序则是可选的
- 解释程序把 HTML 规格转换为适合用户显示硬件的命令来处理版面的细节
- 许多浏览器还包含 FTP 客户程序，用来获取文件传送服务
- 一些浏览器也包含电子邮件客户程序，使浏览器能够发送和接收电子邮件

浏览器中的缓存



- 浏览器将它取回的每一个页面副本都放入本地磁盘的缓存中
- 当用户用鼠标点击某个选项时，浏览器首先检查磁盘的缓存。若缓存中保存了该项，浏览器就直接从缓存中得到该项副本而不必从网络获取，这样就明显地改善浏览器的运行特性
- 但缓存要占用磁盘大量的空间，而浏览器性能的改善只有在用户再次查看缓存中的页面时才有帮助
- 许多浏览器允许用户调整缓存策略

超链接选中后的步骤

■ <https://fcc.gov>

- ① 浏览器确定URL
- ② 请求DNS查询fcc.gov的IP地址
- ③ DNS返回23.1.55.196
- ④ 与该IP地址的433端口上建立TCP连接
- ⑤ 发送HTTPS请求，访问根页面（主页）
- ⑥ 服务器发回配置的页面（index.html）
- ⑦ 获取页面中的其他对象：本服务器的嵌入图形、fonts.gstatic.com的嵌入对象、以及一段取自google-analytics.com的脚本
- ⑧ 浏览器显示页面 “/index.html”
- ⑨ 释放TC连接

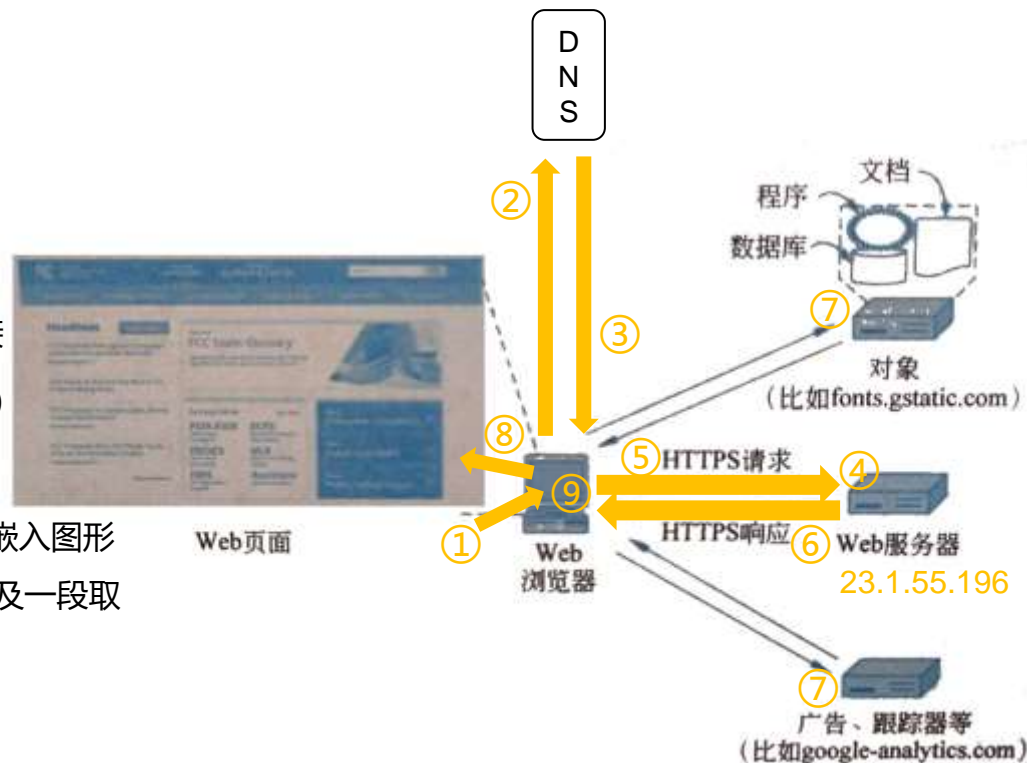


图 7-19 获取和渲染一个 Web 页面要向很多服务器发送 HTTP/HTTPS 请求

服务器端

■ 服务器的工作步骤

- ① 接受来自客户（浏览器）的TCP连接
- ② 获取请求页面的路径（包括请求的文件名）
- ③ 读取文件
- ④ 将文件发送给客户
- ⑤ 释放TCP连接

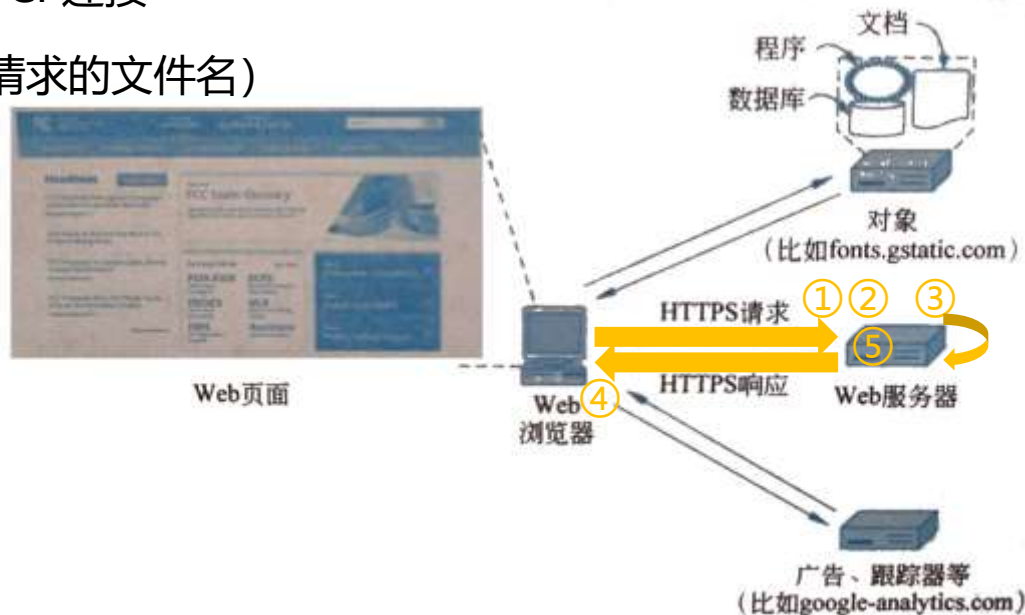


图 7-19 获取和渲染一个 Web 页面要向很多服务器发送 HTTP/HTTPS 请求

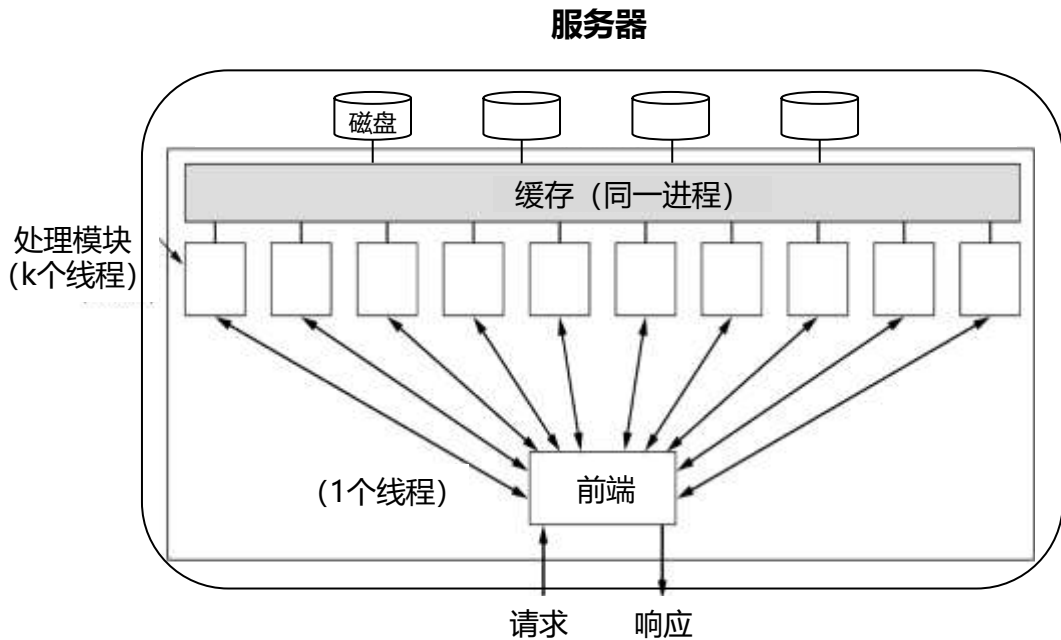
多线程模式



- 监听TCP 80端口

- Web服务器

- 内存缓存技术
- 多线程模式



静态Web对象



■ HTML——超文本标记语言

```
<html>
<head><title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope<i> </i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
<li> <a href="http://widget.com/products/big"> Big widgets </a>
<li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
<li> By telephone: 1-800-WIDGETS
<li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a) 一个web页面的HTML



(b) 格式化后的web页面

万维网文档



- **静态文档**是指该文档创作完毕后就存放在万维网服务器中，在被用户浏览的过程中，内容不会改变
- **动态文档**是指文档的内容是在浏览器访问万维网服务器时才由应用程序动态创建
- 动态文档和静态文档之间的主要差别体现在**服务器**一端。这主要是文档内容的生成方法不同。而从浏览器的角度看，这两种文档并没有区别

动态Web页面与Web应用



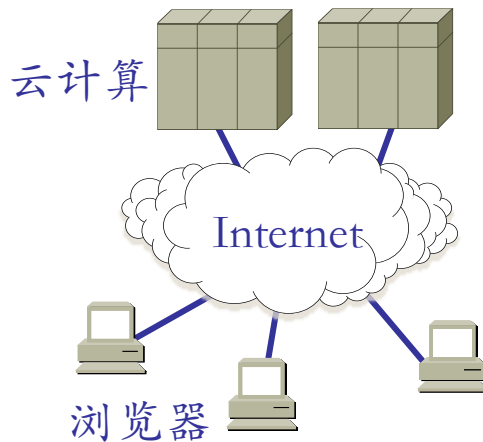
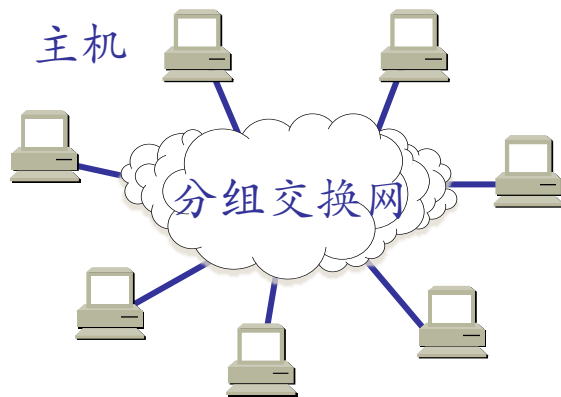
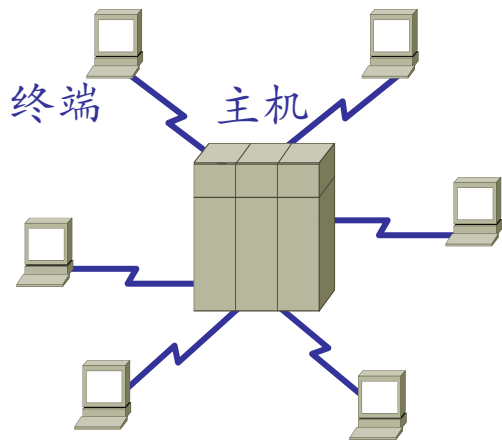
以主机为中心



以网络为中心



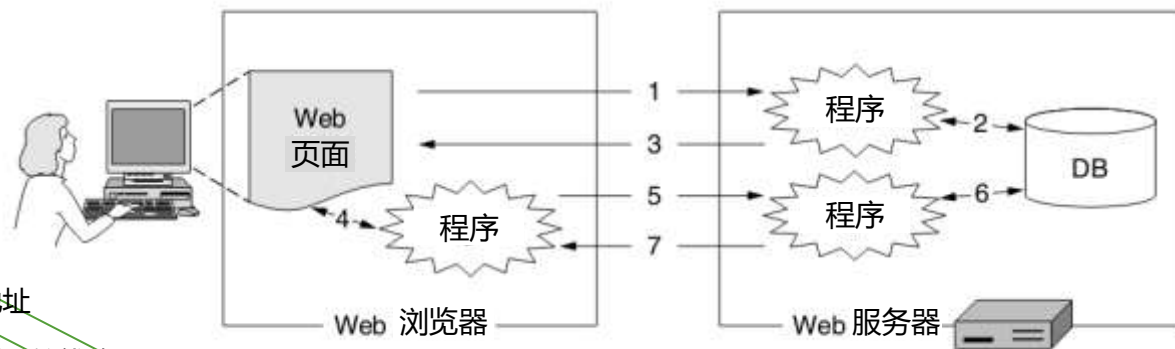
以云主机为中心



动态Web页面的应用

■ 动态页面应用

1. 输入地址请求地图
2. 查询地理信息
3. 返回地图页面
4. 放大缩小操作、输入导航地址
5. 请求相应比例尺地图、规划导航线路
6. 查询更多信息、计算线路时间
7. 返回相应显示信息



动态页面（地图应用）

■ 服务器端动态Web页面生成

■ 客户端动态Web页面生成



服务器端动态Web页面生成

■ 公共网关接口CGI

- 包含表单的Web页面
- 允许Web服务器与后端程序或脚本通信
- 后端程序(由Python等开发)常驻于cgi-bin目录下:
order.cgi (静态页面表单例子)

■ 超文本预处理器

- PHP脚本

■ 其他

- JSP
- ASP

```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a) 包含一个表单的web页面

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

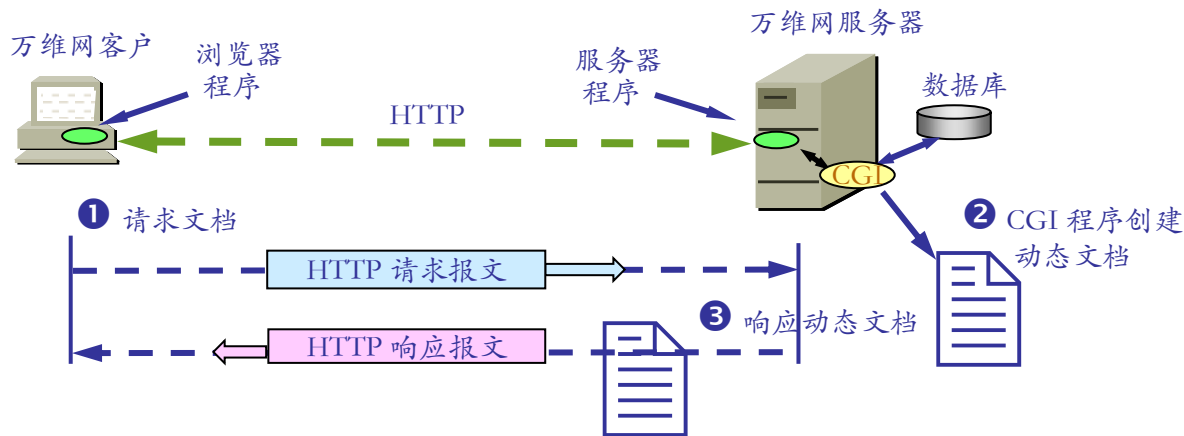
(b) 处理该表单输出的PHP脚本

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

(c) 输入为“Barbara”和“24”时PHP脚本的输出

万维网服务器功能的扩充

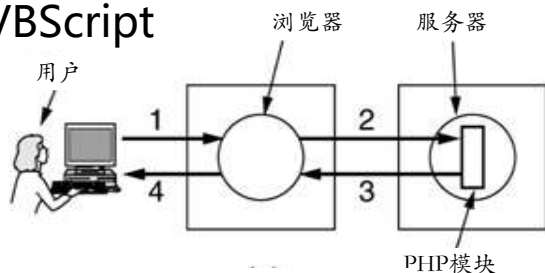
- (1) 增加另一个应用程序，用来处理浏览器发来的数据，并创建动态文档
- (2) 增加一个机制，用来使万维网服务器把浏览器发来的数据传送给这个应用程序，然后万维网服务器能够解释这个应用程序的输出，并向浏览器返回 HTML 文档



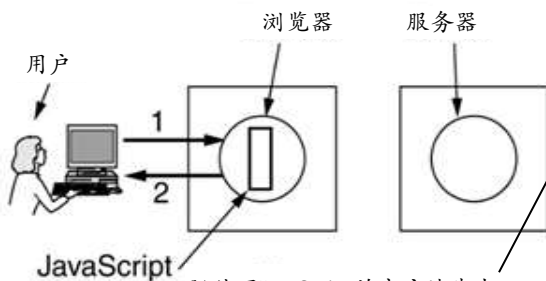
客户端动态网页生成

■ 动态HTML

- JavaScript
- VBScript



(a)使用PHP的服务器端脚本



(b)使用JavaScript的客户端脚本

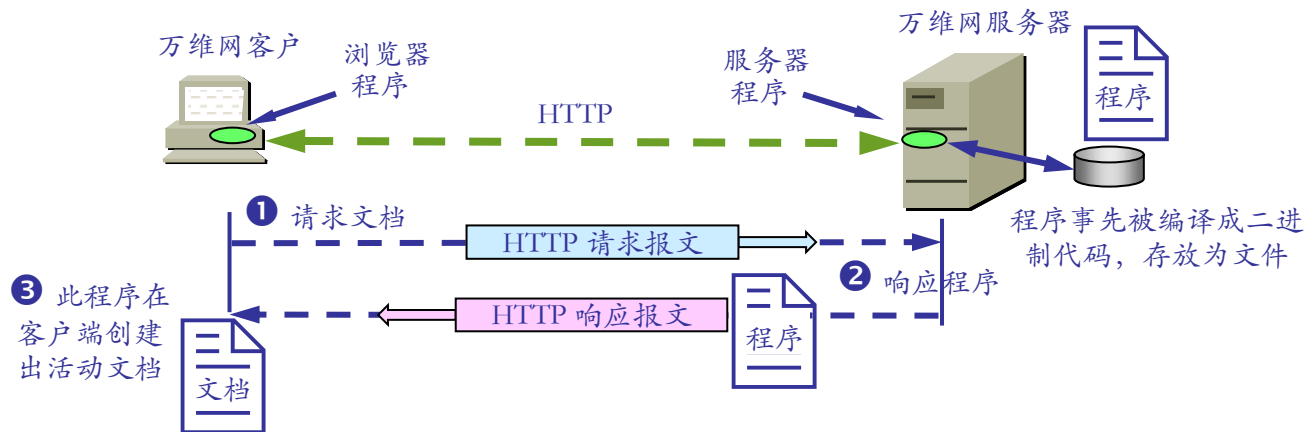
```
<head>
<script language="javascript" type="text/javascript">
function response(test form) {
    var person = test form.name.value;
    var years = eval(test form.age.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hello " + person + "<br>");
    document.writeln("Prediction: next year you will be " + years + ".");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```

活动万维网文档



- **活动文档**(active document)技术把所有的工作都转移给浏览器端
- 每当浏览器请求一个活动文档时，服务器就返回一段程序副本在浏览器端运行
- 活动文档程序可与用户直接交互，并可连续地改变屏幕的显示
- 由于活动文档技术不需要服务器的连续更新传送，对网络带宽的要求也不会太高



HTTP和HTTPS



■ 概述

- HTTP像一个传输协议（简单的请求-响应协议）
 - ◆ RFC 2616
- HTTPS: HTTP + TLS (Transport Layer Security, 传输层安全传输协议)

■ 与传输层关系

- HTTP通常运行在TCP之上
- HTTP 3也可以运行在UDP之上



HTTP请求方法

方 法	描 述
GET	请求读取Web网页
HEAD	请求读取Web网页头
PUT	请求存储Web网页
POST	附加一个命名资源(如Web网页)
DELETE	删除万维网页
TRACE	送回收到的请求
CONNECT	保留将来使用
OPTIONS	查询特定选项

HTTP 的报文结构

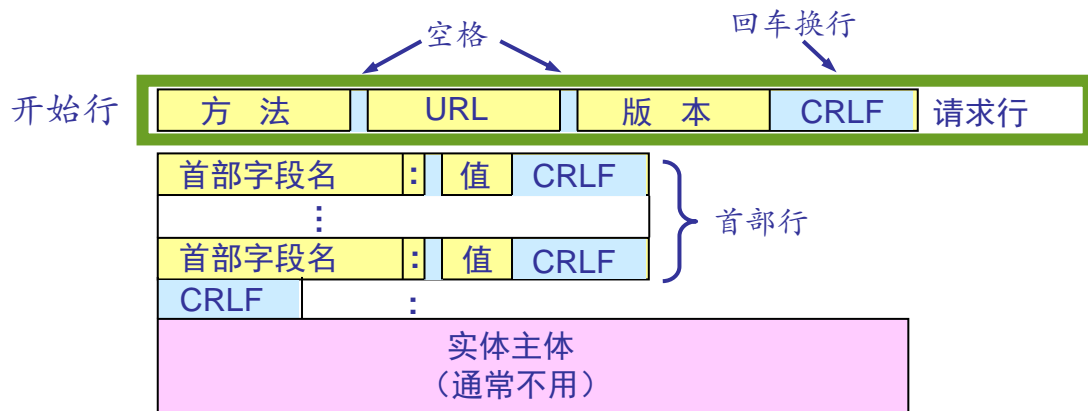


HTTP 有两类报文：

- 请求报文——从客户向服务器发送请求报文
- 响应报文——从服务器到客户的回答

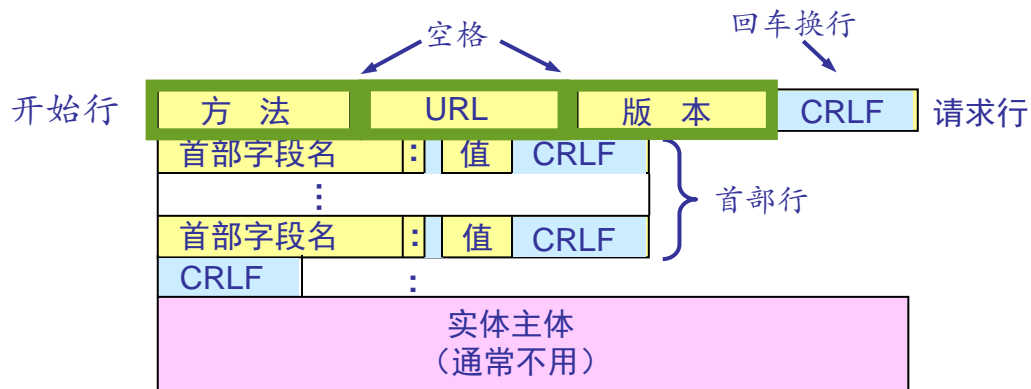
由于 HTTP 是面向正文的(text-oriented)，因此在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的

HTTP 的报文结构 (请求报文)



报文由三个部分组成，即开始行、首部行和实体主体
在请求报文中，开始行就是请求行

HTTP 的报文结构 (请求报文) (2)

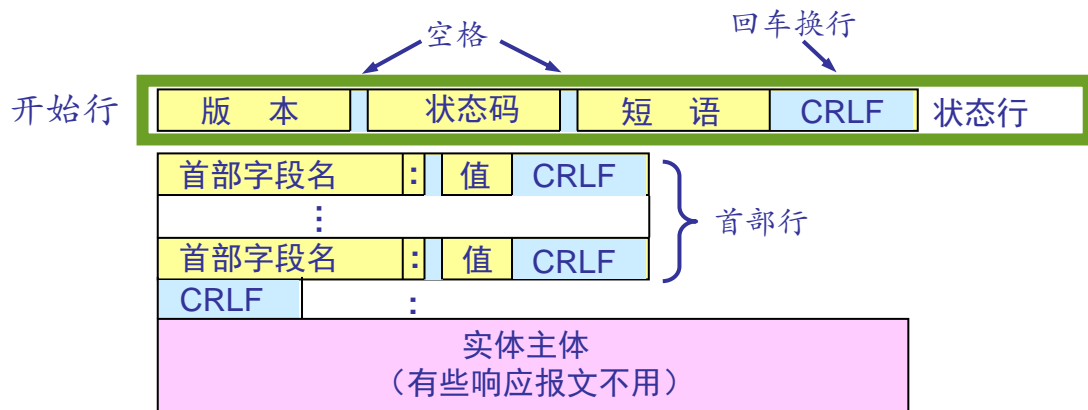


“方法”是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的

“URL”是所请求的资源的 URL

“版本”是 HTTP 的版本

HTTP 的报文结构 (响应报文)



响应报文的开始行是状态行

状态行包括三项内容，即 HTTP 的版本，状态码，以及解释状态码的简单短语



状态码都是三位数字

- 1xx 表示通知信息的，如请求收到了或正在进行处理
- 2xx 表示成功，如接受或知道了
- 3xx 表示重定向，表示要完成请求还必须采取进一步的行动
- 4xx 表示客户的差错，如请求中有错误的语法或不能完成
- 5xx 表示服务器的差错，如服务器失效无法完成请求

HTTP的消息头与缓存

■ 消息头

- 请求行（如GET）后还有额外的行，称为消息头（类似与过程调用的参数，参见书P517）

■ 缓存

● 方法一

根据页面“过期字段”判断

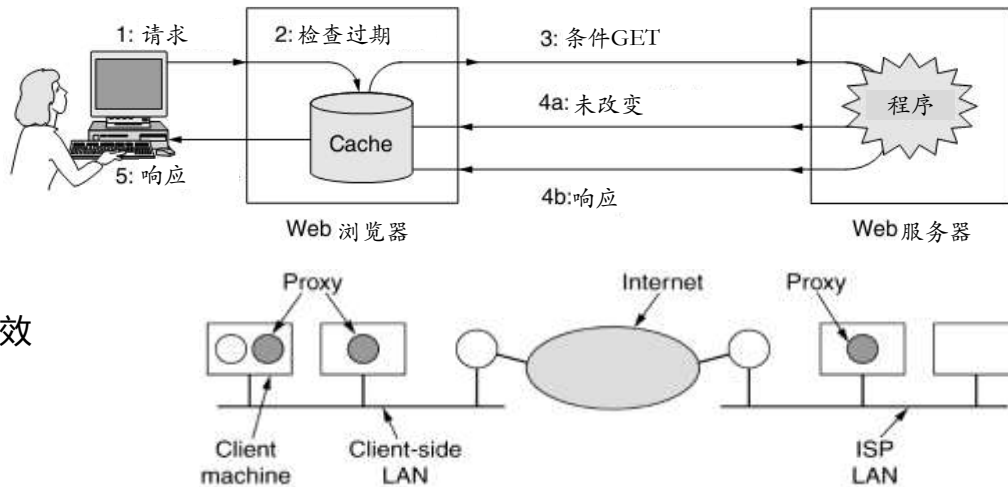
1, 2, 5

● 方法二

3、条件GET，询问副本是否有效

4a、是

4b、返回更新页面



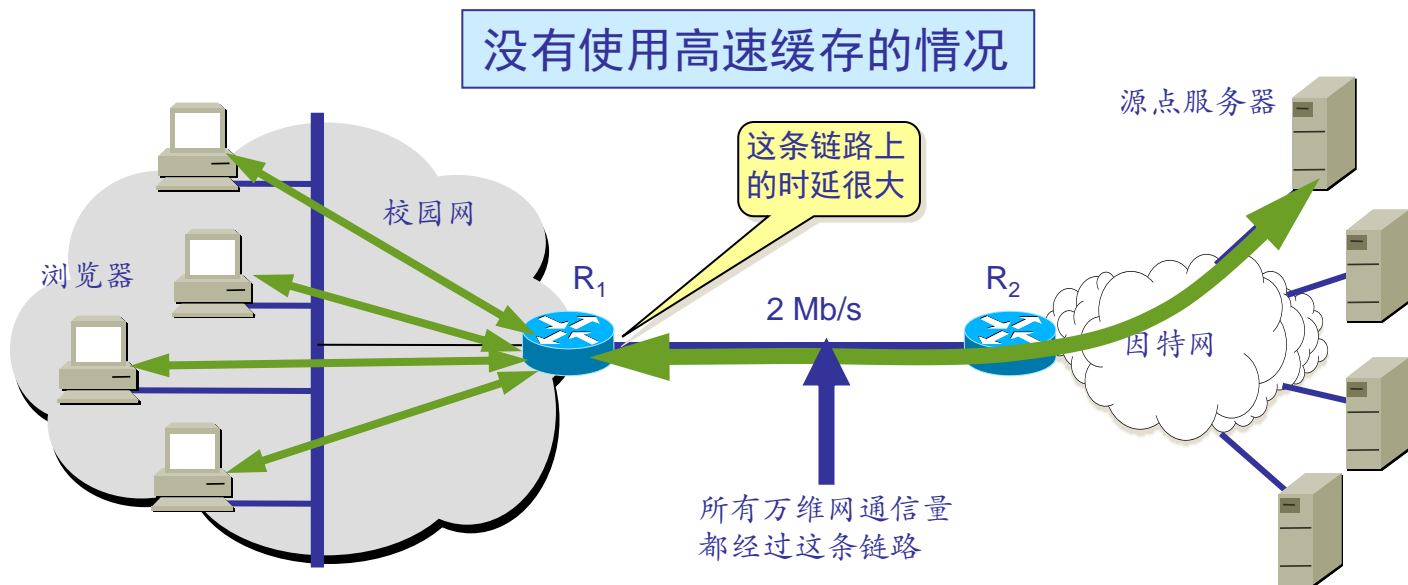


代理服务器

- 代理服务器(proxy server)又称为万维网高速缓存(Web cache), 它代表浏览器发出 HTTP 请求
- 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中
- 当与暂时存放的请求相同的新请求到达时, 万维网高速缓存就把暂存的响应发送出去, 而不需要按 URL 的地址再去因特网访问该资源

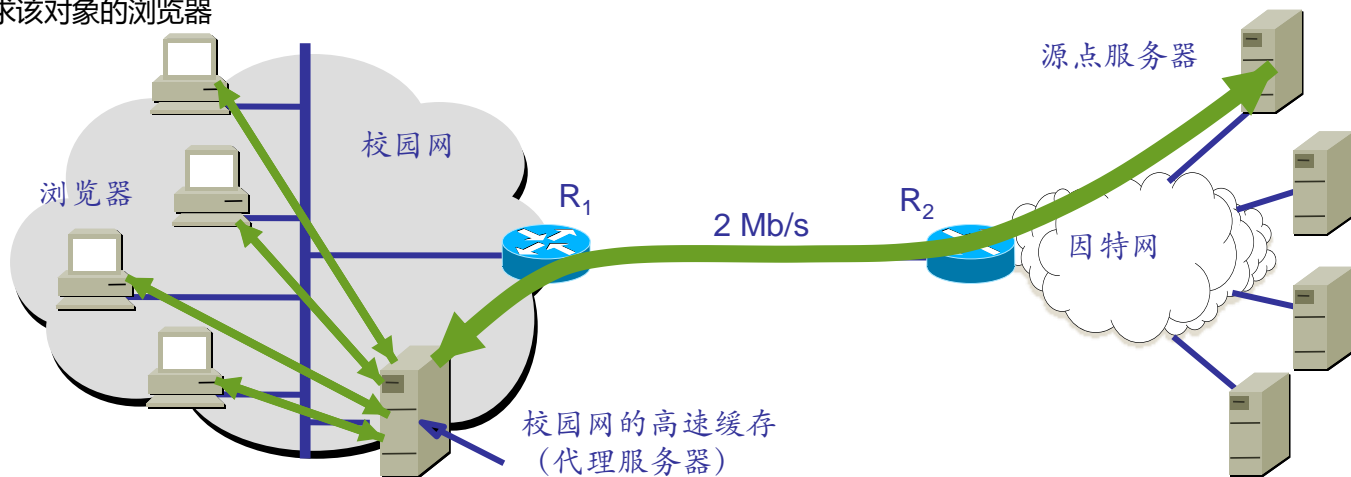


使用高速缓存可减少访问因特网服务器的时延



使用高速缓存的情况

- (1) 浏览器访问因特网的服务器时，要先与校园网的高速缓存建立 TCP 连接，并向高速缓存发出 HTTP 请求报文
- (2) 若高速缓存已经存放了所请求的对象，则将此对象放入 HTTP 响应报文中返回给浏览器
- (3) 否则，高速缓存就代表发出请求的用户浏览器，与因特网上的源点服务器建立 TCP 连接，并发送 HTTP 请求报文
- (4) 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存
- (5) 高速缓存收到此对象后，先复制在其本地存储器中（为今后使用），然后再将该对象放在 HTTP 响应报文中，通过已建立的 TCP 连接，返回给请求该对象的浏览器

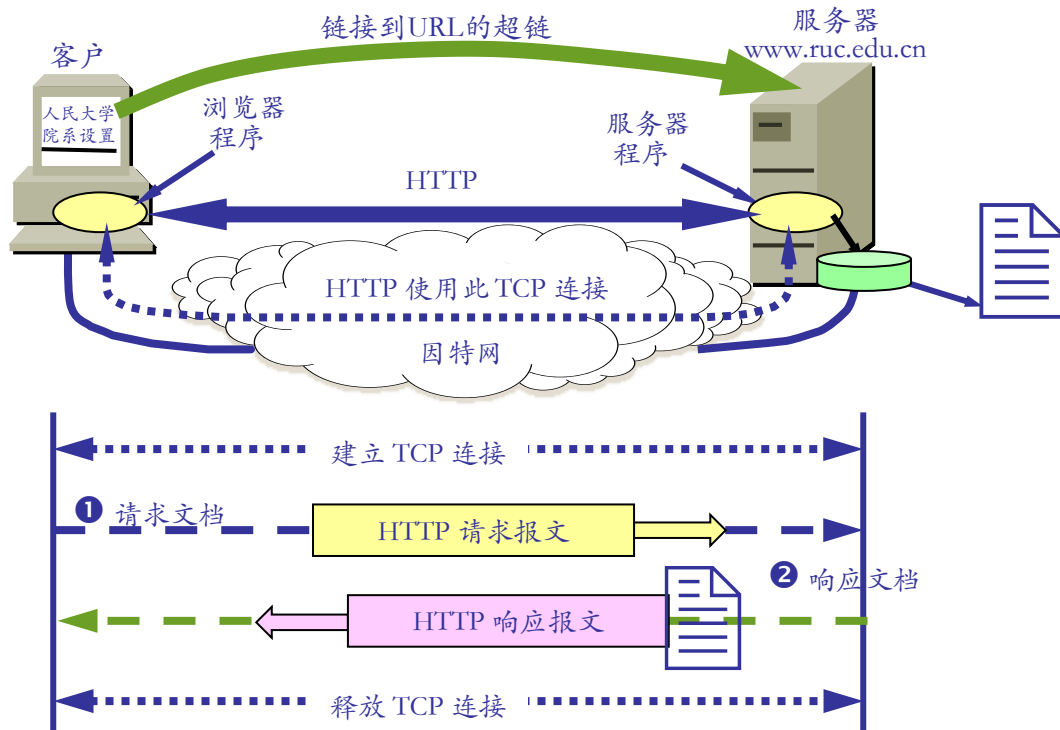


HTTP 1和HTTP 1.1



- 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
- HTTP的主要特点
 - 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

万维网的工作过程





用户点击鼠标后所发生的事件

- (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.ruc.edu.cn` 的 IP 地址
- (3) 域名系统 DNS 解析出中国人民大学服务器的 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 /scienceandtechnology/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 “中国人民大学院系理工学部” 文件 `index.htm` 中的所有文本

HTTP1.1



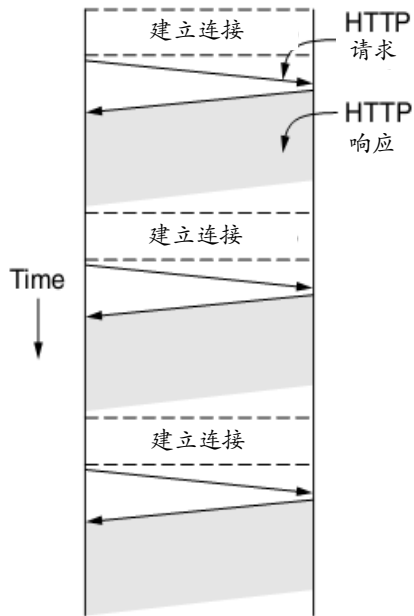
■ 连接

HTTP1.1: 持续连接

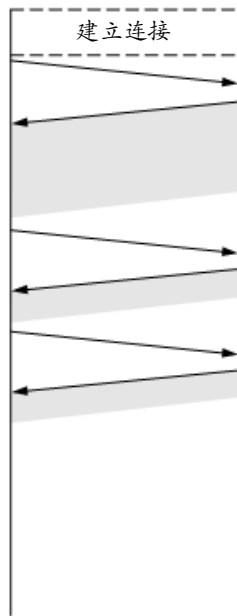
a. 多个连接

b. 一个持续连接

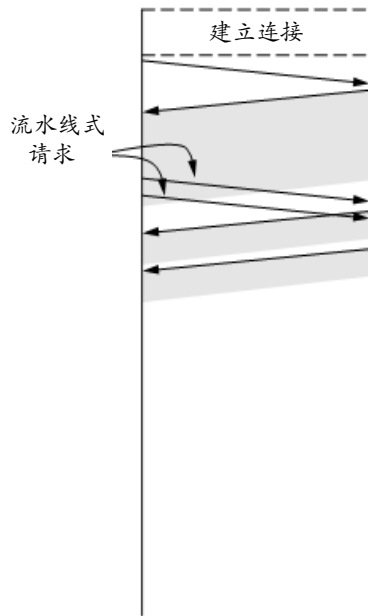
c. 流水线式请求



(a)

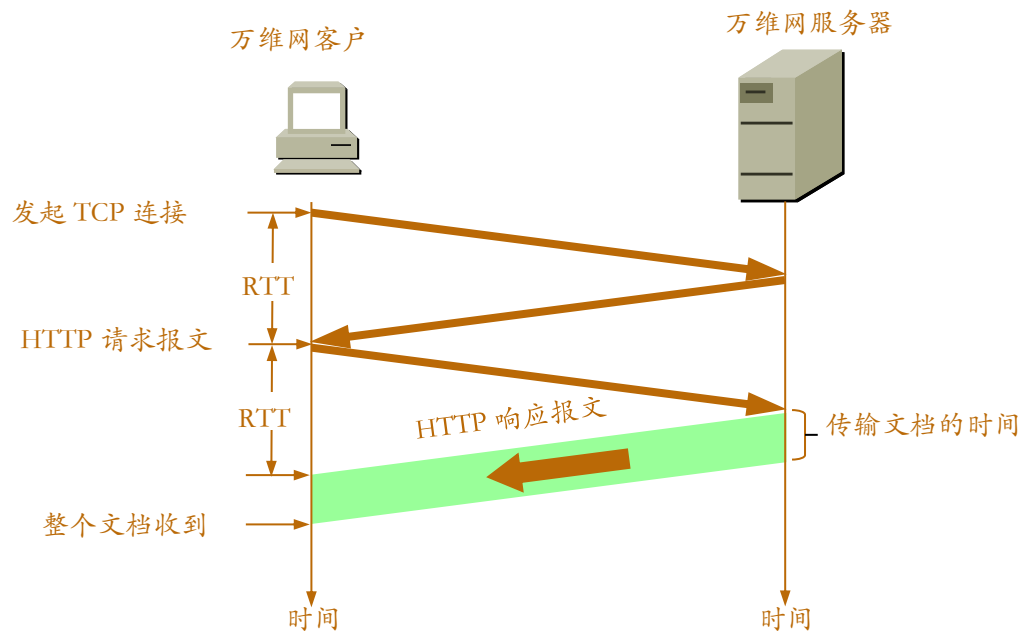


(b)



(c)

请求一个万维网文档所需的时间



持续连接

■ 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
- 某些浏览器（例如，IE 6.0）的默认设置就是使用 HTTP/1.1

■ 工作方式

- 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

HTTP 2



■ 发展趋势

- 1989Web开始, HTTP 1
- 2007 HTTP 1.1
- 2012 HTTP 2 起始于Google公司的SPDY协议, 2015.5 RFC 7540
- Windows 11上的Edge放弃自己的技术采用chrome内核

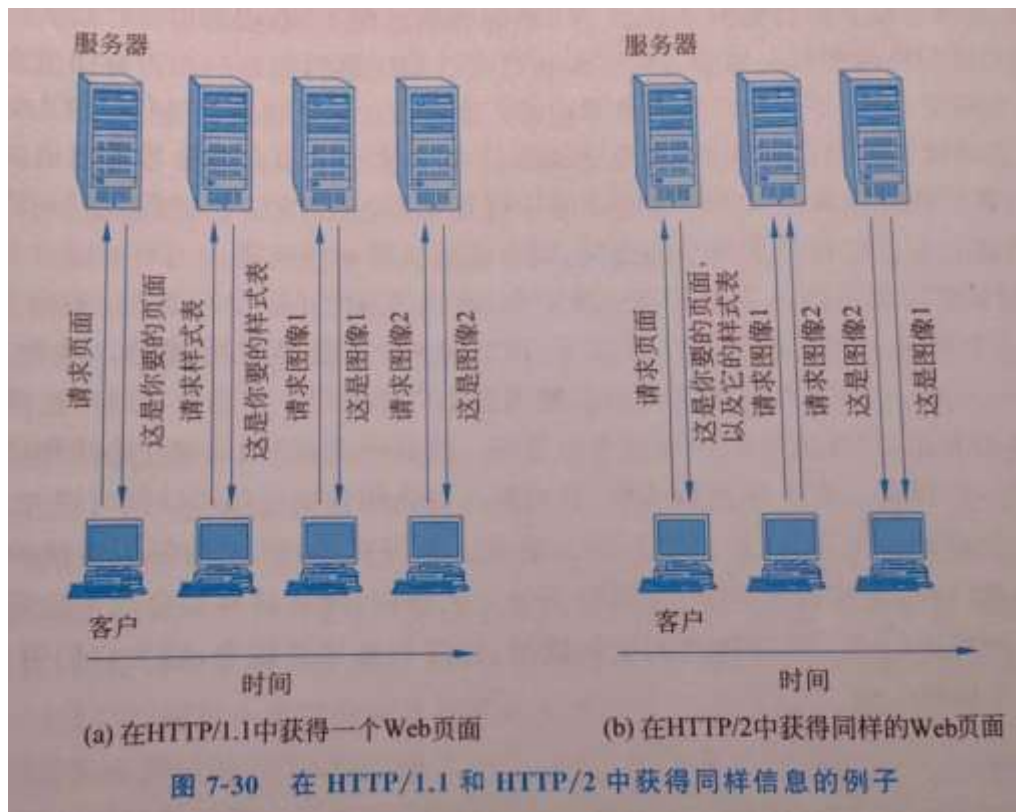
■ 目标

- 允许客户和服务端选择使用哪个HTTP版本
- 维持与 HTTP 1.1的兼容性
- 通过多路复用、流水线、压缩等方法提高性能
- 支持浏览器、服务器、代理、分发网络等已有技术

HTTP 2服务器推送

■ 兼容

- 头、URL和通用语义基本不变
- 改变的是其中的被编码方式和客户与服务器交互方式
- 例：Web页面中包含“样式表”和两个图像



HTTP 3



- HTTP 3与QUIC (Quick UDP Internet Connections)
 - 支持HTTP 3消息的传输协议不再依赖于TCP
 - QUIC为UDP增强版本, 依赖于运行在UDP之上的用户空间拥塞控制机制
 - QUIC传输协议支持流的多路复用及每个流的流量控制
 - HTTP 3允许客户的多个不同的URL复用同一连接

Web隐私



- Cookie
- 第三方跟踪器
- 设备与浏览器指纹



在服务器上存放用户的信息

- 网站使用 Cookie 来跟踪用户
- Cookie 表示在 HTTP 服务器和客户之间传递的状态信息
- 使用 Cookie 的网站服务器为用户产生一个唯一的识别码。利用此识别码，网站就能够跟踪该用户在该网站的活动
- 同源策略：不同域设置的Cookie相互隔离

Cookie



■ 内容 (5个字段)

域	路径	内容	过期	安全
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

■ 使用

- 标识顾客
- 商品列表 (购物车)
- 参数设置 (Web门户网站)
- 服务器访问记数

第三方跟踪器



■ 第三方跟踪

- 加载第三方公告或脚本，第三方设置唯一Cookie
- 以后访问不同站点，可能加载同一个第三方的Web对象
- 各个不同的Web站点使用同样的广告网络提供的服务，第三方可以跨站点跟踪，生成超级Cookie

■ 第三方跟踪软件

- 跨设备跟踪
- Cookie同步
- 为用户建立全面的互联网行为画像



设备和浏览器指纹

■ 基于设备

- 根据设备反馈给服务器的环境信息、上下文信息和设备信息
- 画布指纹：利用定制的HTML页面，通过显示的画面特征作为设备指纹

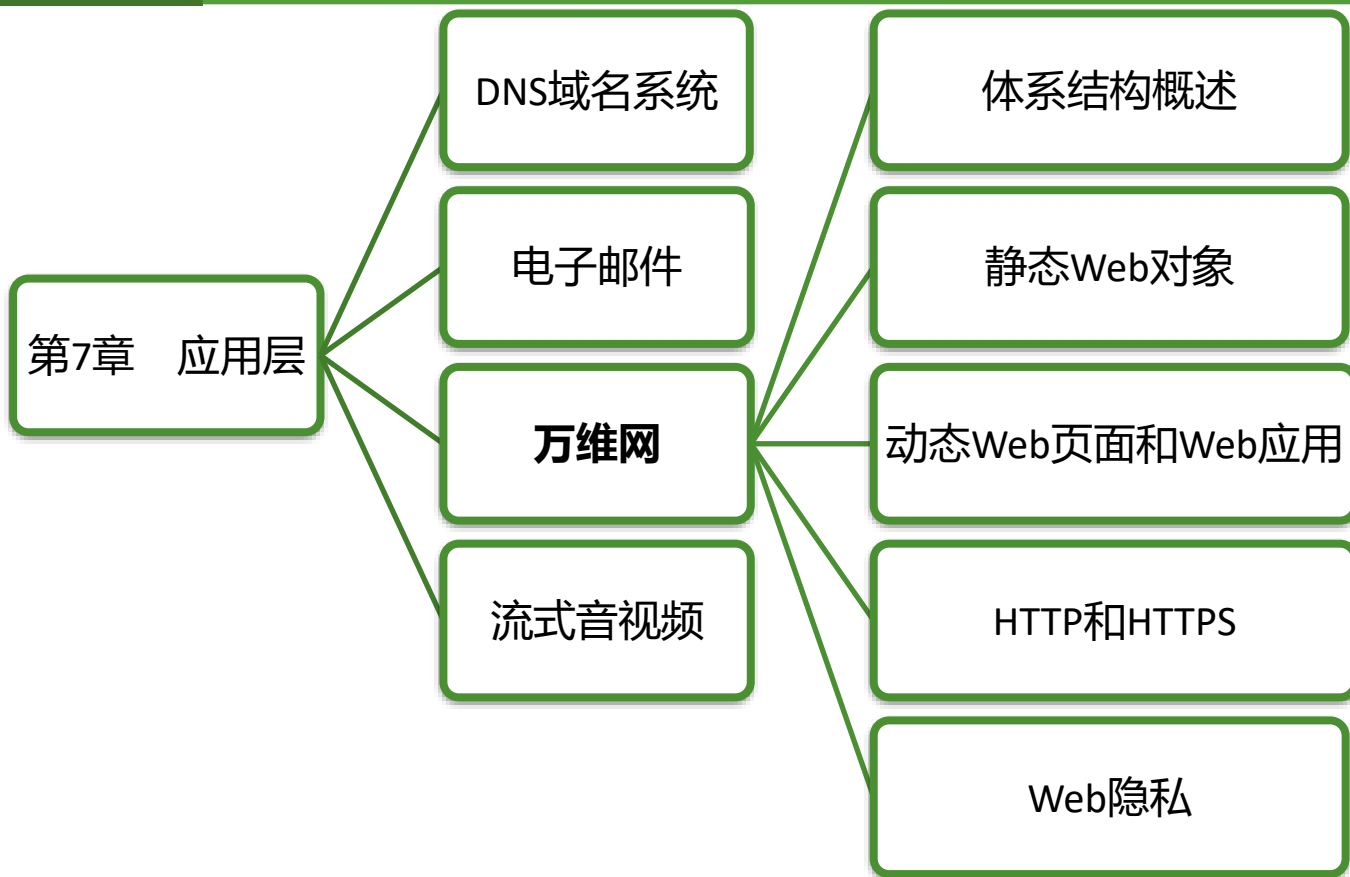
■ 基于浏览器

- 使用浏览器API获得可用于跟踪设备的信息，如电池信息等

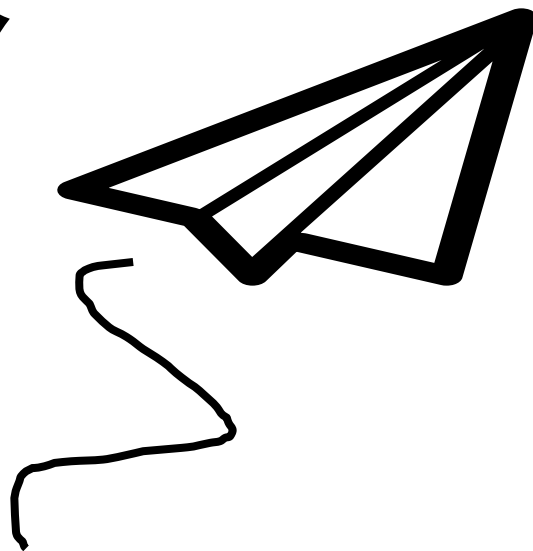
■ Cookie同步

- 不同的第三方跟踪器彼此共享信息

本章导航与要点



本节课程结束



7.4 流式音视频



- 数字音频
- 数字视频
- 存储媒体的流式传输
- 实时流式传输

数字音频



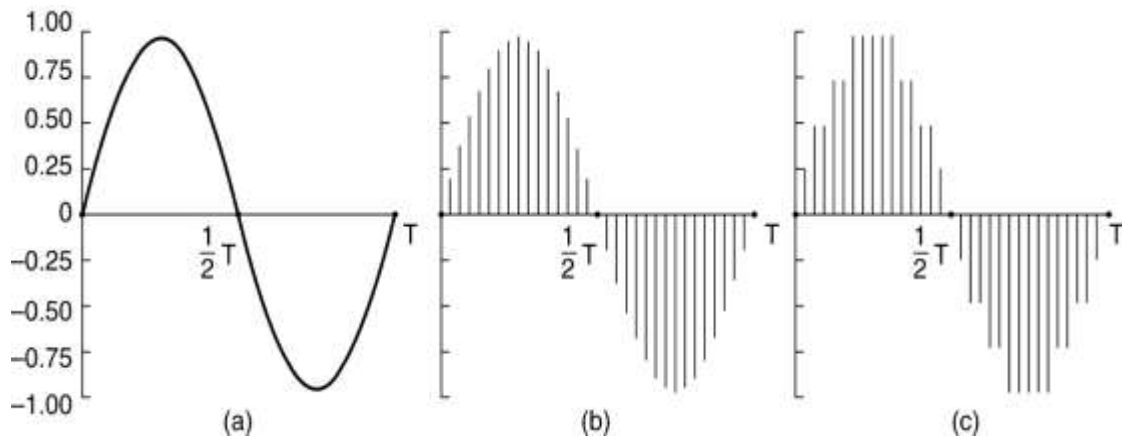
■ 电话系统

- 56kbps、64kbps

■ CD

- 705.6kbps、1.411Mbps

■ 模数转换ADC





音频带宽

■ 电话和CD

- 脉冲编码调制PCM ($<4\text{kHz}$)
 - ◆ $8000 \times 7 = 56\text{kb/s}$ 北美和日本
 - ◆ $8000 \times 8 = 64\text{kb/s}$ 欧洲
- CD ($<22050\text{Hz}$)
 - ◆ $44100 \times 16 = 705.6\text{kb/s}$ ($\times 2 = 1.411\text{Mb/s}$)

■ 乐器数字接口MIDI

- MIDI码: 0 大钢琴, 40 小提琴
- 带宽为1/1000

音频压缩

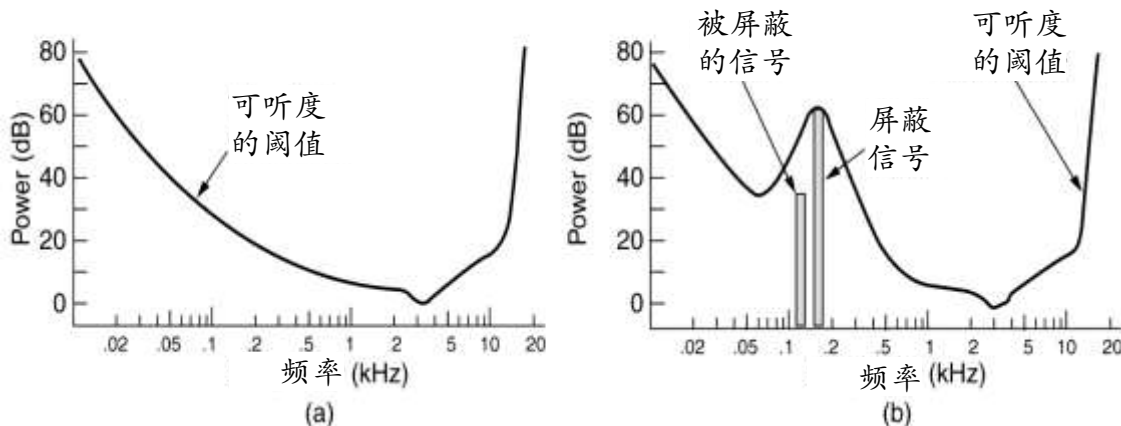
■ 编码与解码

- MEPEG音频层3(MP3)与高级音频编码(AAC)

- ◆ 视频: MPEG1、MPEG2、MPEG4

- ◆ 音频: MP3、AAC(MP4)

- 音频压缩: 波形编码、感知编码



数字视频



■ 模拟系统

制式	行数	比例	帧频
NTSC	525(483)	4:3	30(29.97)
PAL	625(576)	4:3	25

■ 数字系统

格式	Internet		DTV			
	低	全屏	LDTV	SDTV	HDTV	UHDTV
分辨率	320*240	640*480	340*255	720*570	1280*720(720p) 1920*1080(1080p)	8K: 7680*4320 (4K: 3840*2160)

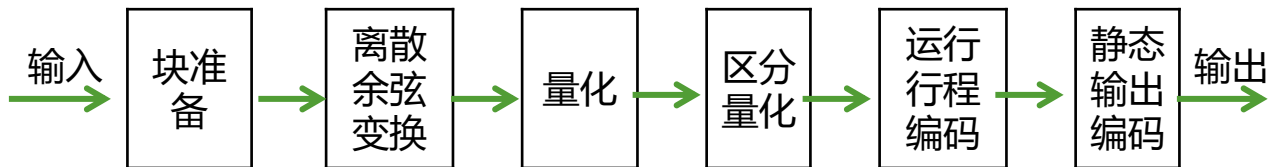
$$1280*720*24*25 = 553\text{Mbps} (552960000)$$



视频压缩——JPEG

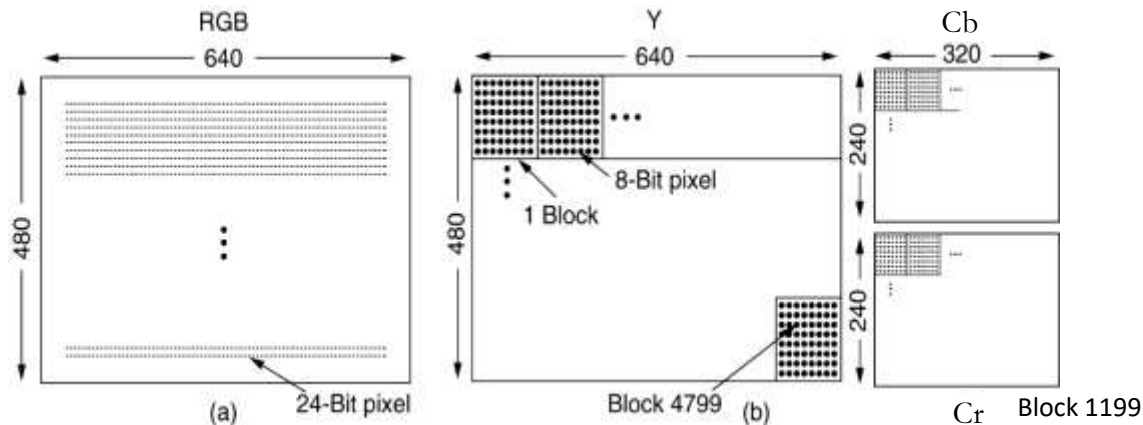
■ JPEG

- 亮度 $Y = 16 + 0.26R + 0.50G + 0.09B$
- 色度 $Cb = 128 + 0.15R - 0.29G - 0.44B$
- 色度 $Cr = 128 + 0.44R - 0.37G + 0.07B$



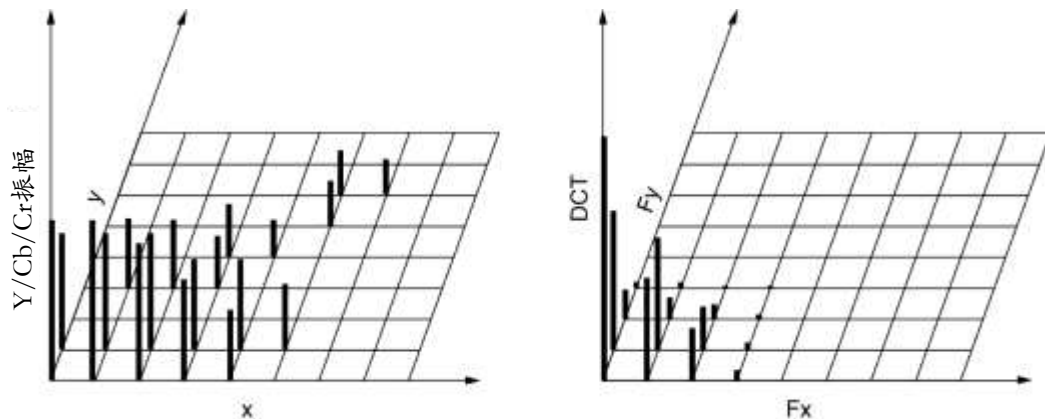
JPEG

1. 块准备



2. 离散余弦变换

7200块单独应用DCT



JPEG(2)

3. 量化

DCT 系数

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

量化表

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

C量化后的系数

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

4. 将每块(0,0)元素值，用与前块对应的差值代替

5. 行程编码(Z字形扫描)

6. 对数值进行霍夫曼编码

常见的数值分配较短的编码

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

MPEG标准

■ MPEG-1(1993)

- 要求1.2Mbps带宽
- $352*240*24*25 \approx 48.34\text{Mbps}$
- $48/1.2=40\text{倍}$

■ MPEG-2(1996)

- 要求4~8Mbps

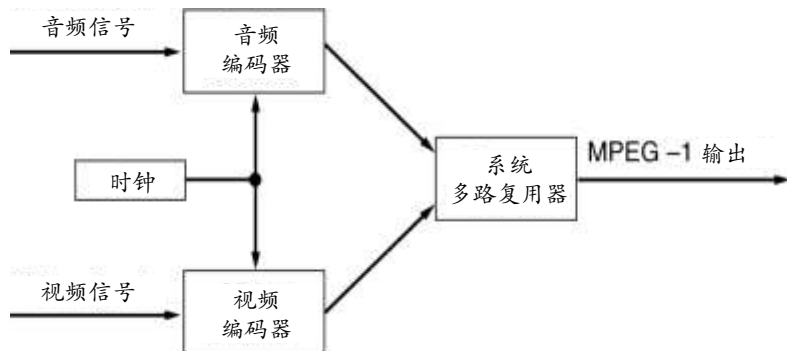
■ MPEG-4

- (1999)合成图像混合
- (2003)H.264,AVC

高级视频编码用于蓝光光盘的HDTV

■ MPEG同时压缩音频和视频

- 接收端同步两个流：用时钟产生时间戳



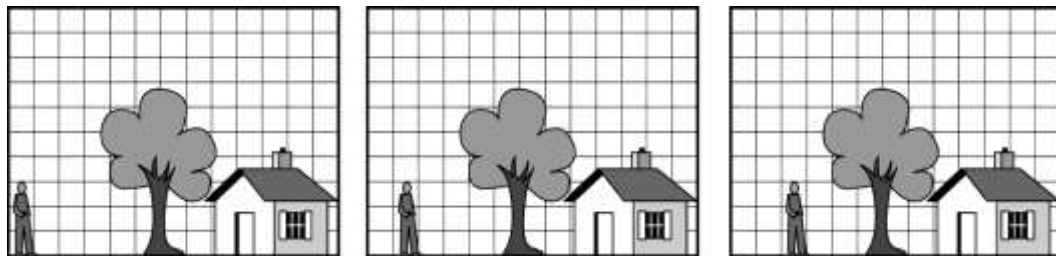


MPEG标准(2)

■ MPEG压缩空间和时间冗余

MPEG输出的3类帧

- I(Intracoded): 帧内编码帧包含压缩的静止图片
- P(Predictive): 预测帧是与前一帧的逐块差值
- B(Bidirectional): 双向帧是与前一帧和后一帧的逐块差值

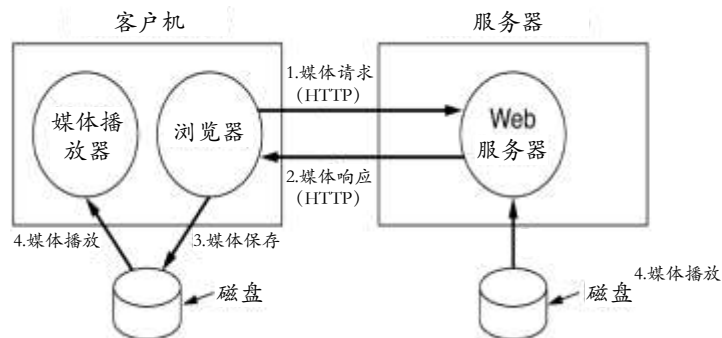


存储媒体的流式传输



■ 音、视频点播

■ 下载播放



■ Web播放

■ 媒体播放器

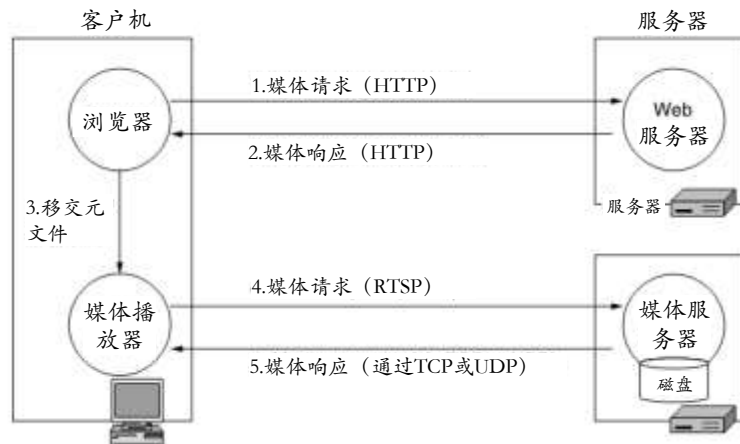
■ 管理用户界面

■ 处理传输错误

■ 解压缩内容

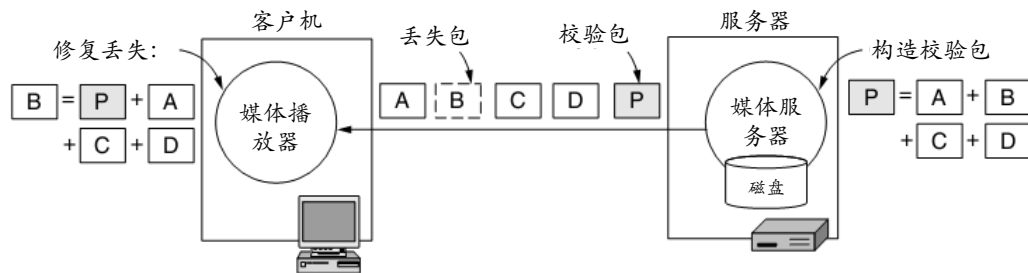
■ 消除抖动

■ 解密文件 (商业视频)

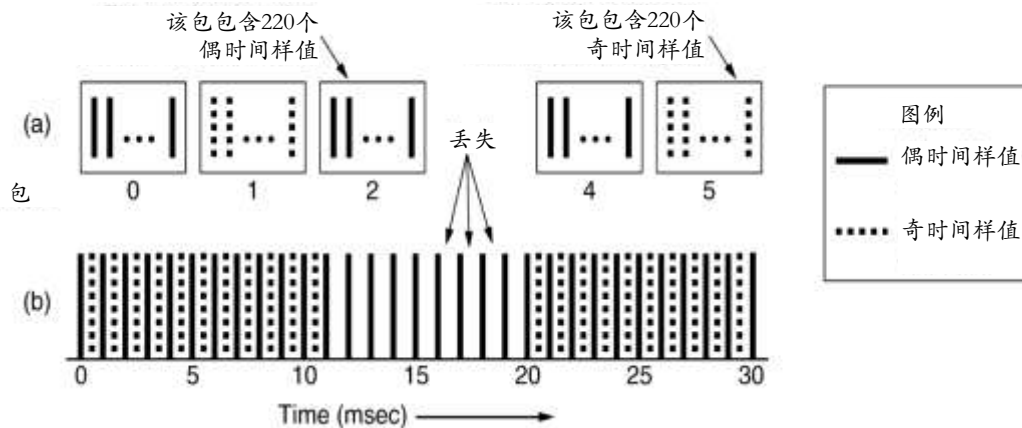


流式媒体纠错

■ 前向纠错



■ 交替编码



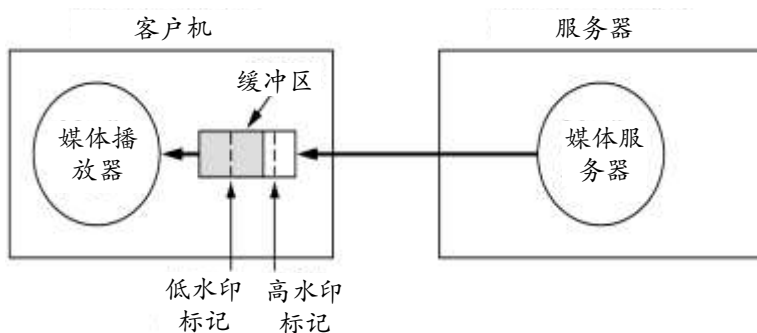
流式存储媒体



■ 解压缩

- 丢包与否都能解压缩
- MPEG中的I-帧、 P-帧、 B-帧

■ 消除抖动



DASH

■ DASH (Daynamic Adaptive Streaming over HTTP)

基于HTTP的动态自适应流传输



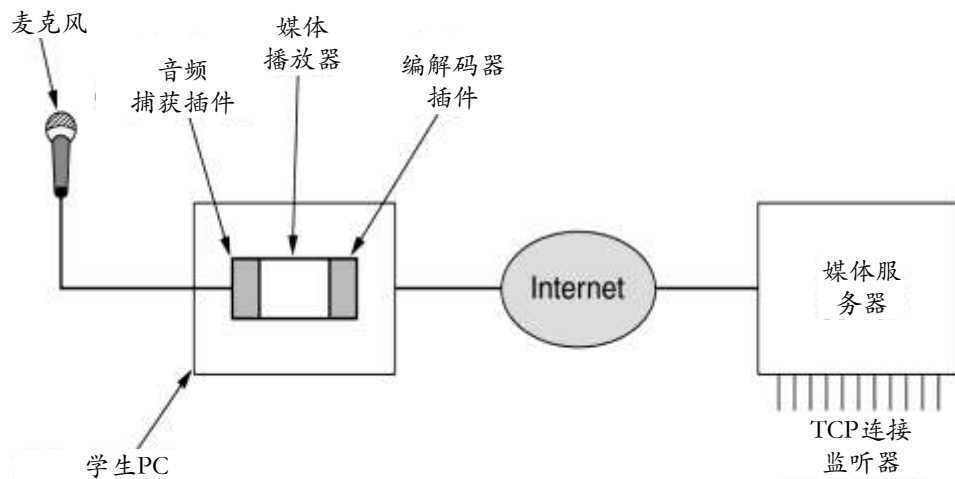
实时流式传输



- IP语音
- H.323
- SIP

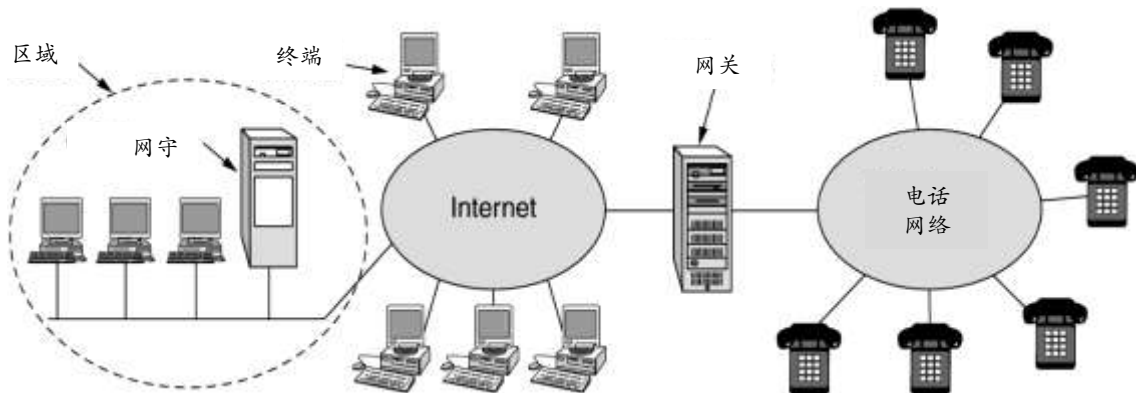
流式直播媒体

■ IPTV与网络广播



IP语音与H.323

- IP语音：延迟与服务质量机制（DS区分服务）
- H.323的Internet电话结构模型



H.323



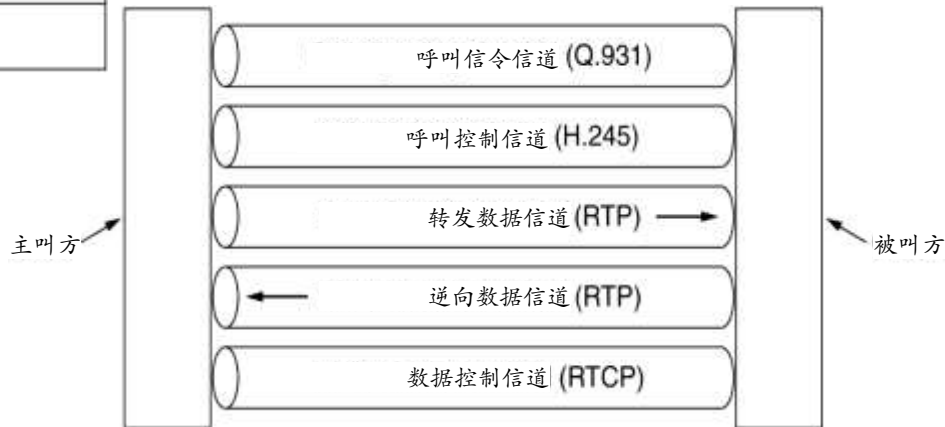
■ 协议栈

音频 (视频)	控制			
G.7xx (H.26x)	RTCP	H.225 (RAS)	Q.931 (信令)	H.245 (呼叫控制)
RTP				
UDP			TCP	
IP				
数据链路层协议				
物理层协议				

H.323协议栈

- 语音编码解码：G.711
- 协商压缩算法和数据速率：H.245
- RTCP用来控制RTP通道
- Q.931用于建立释放连接，提供拨号音、响铃等
- 与网守进行通话：H.225

■ 呼叫逻辑信道



SIP——会话发起协议

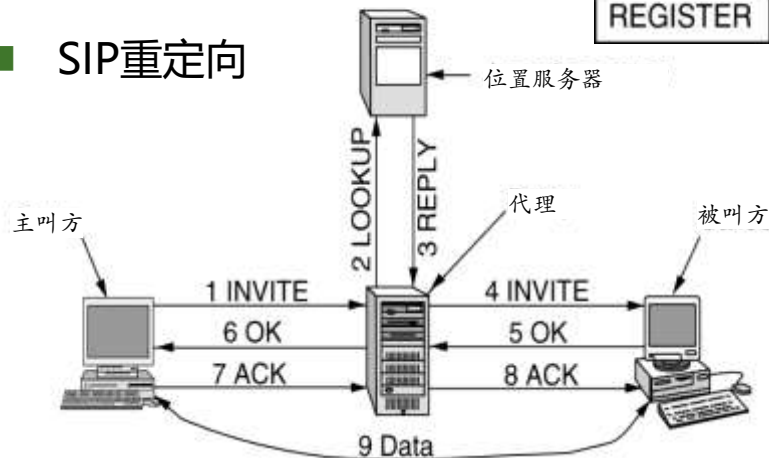
■ 简单的IP语音

SIP: Session Initiation Protocol

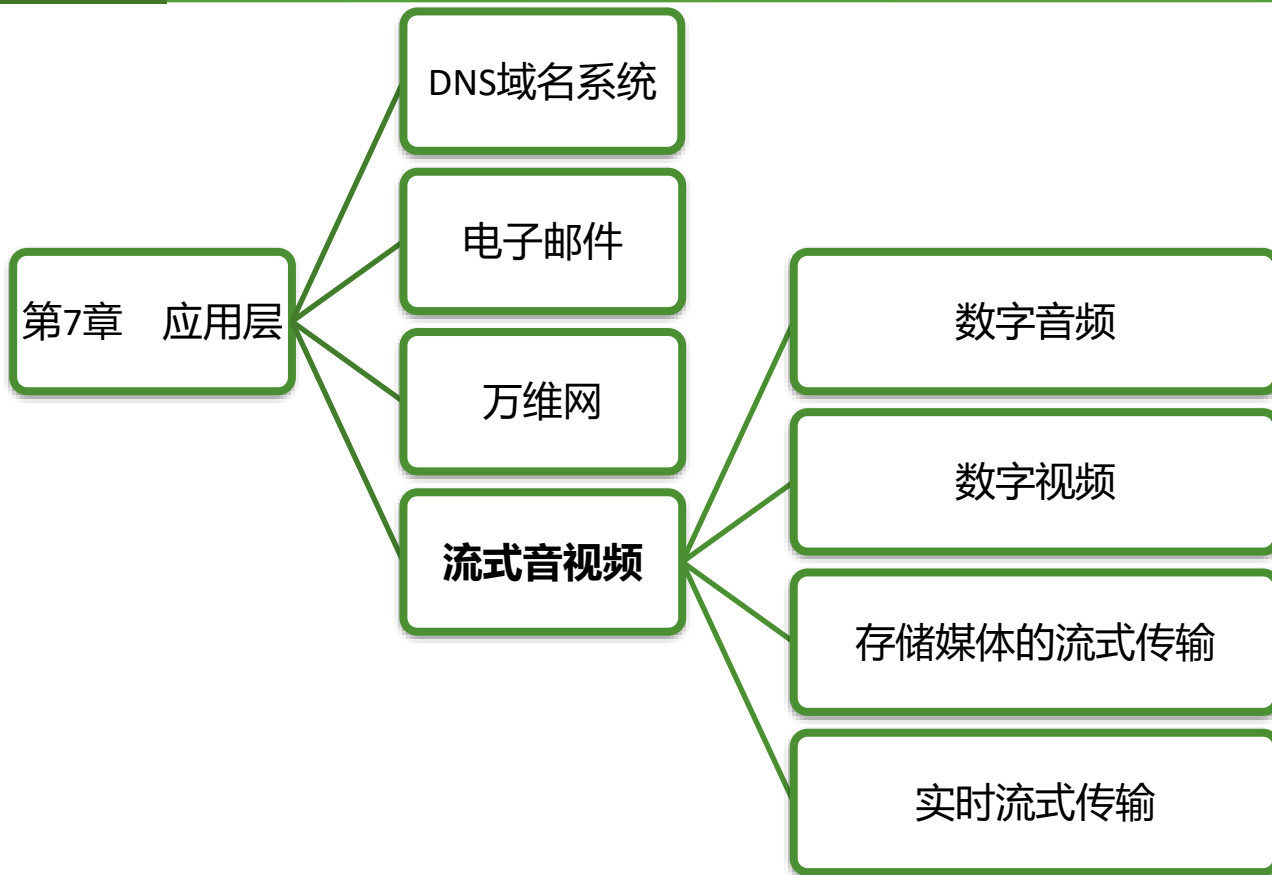
■ SIP方法

方法	描述
INVITE	请求发起一次会话
ACK	确认会话已经启动
BYE	请求终止会话
OPTIONS	查询主机的能力
CANCEL	取消正在进行中的请求
REGISTER	通知服务器重新定向用户的当前位置

■ SIP重定向



本章导航与要点



本章课程结束

