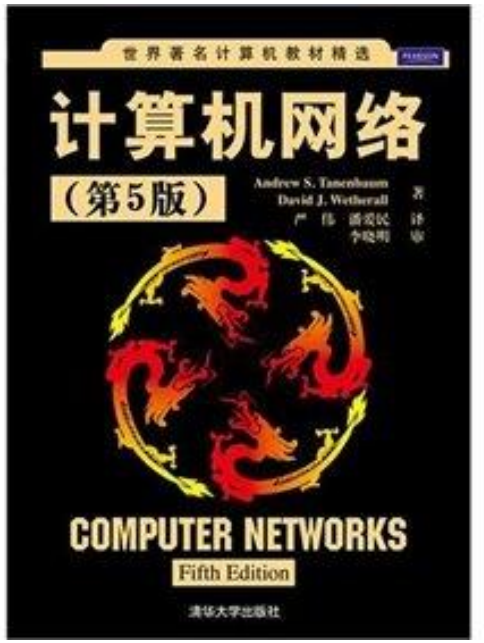
计算机网络

Andrew S. Tanenbaum（5 Edition）



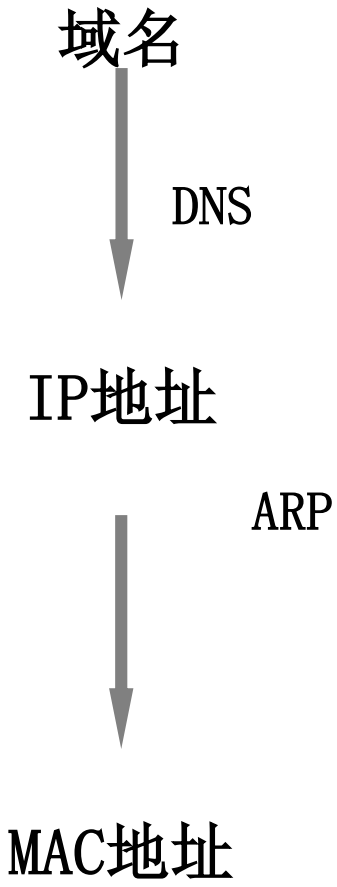
安徽大学 互联网学院
School of Internet Anhui University

计算机网络

第7章 应用层

7.1 DNS域名系统

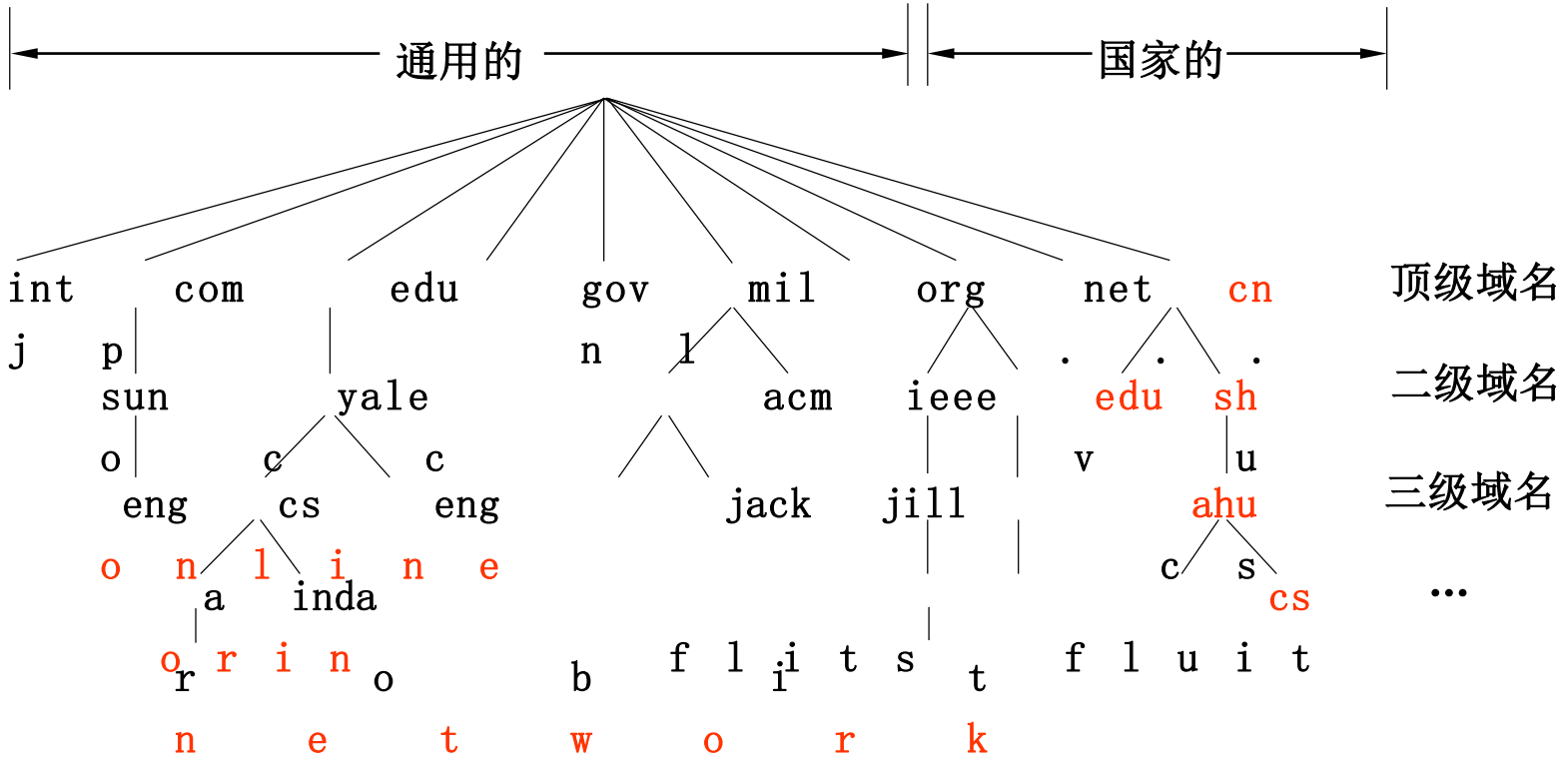
- DNS系统将域名转换为网络地址（IP地址）
- 早期的域名-IP之间的映射使用一个配置文件，所有主机从服务器上下载该文件到本地，然后更新和查询。
- 随着网络规模的增长，一种分布式的、层次结构的域名系统(DNS)被广泛应用。



7.1 DNS域名系统

DNS的名字空间

- DNS的分层结构



7.1 DNS域名系统

- 命名：域名是从叶到根的路径，用点分开

四级域名 … 顶级域名

- **例：** http://www.ahu.edu.cn/

- 其中的域名分为四级,并且对应一台地址为101.76.160.174的服务器。

- 域名需要申请和注册，不区分大小写

- 我国允许在cn顶级域名下直接注册二级域名，例如***.cn

- 创建新域名必须得到上级域名管理机构的批准。以组织为边界而非物理边界。

7.1 DNS域名系统

域名资源记录

- 每一个域都有一组资源记录，构成DNS数据库，用于响应DNS查询，对任意一台主机来说，最常见的资源记录是它的IP地址

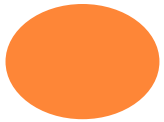
Domain name	Time to live	class	type	value

- Domain name：该资源记录对应的域
- Time to live：该资源记录的生存期（秒）
- Class：常量“IN”，即Internet类
- Type：记录类型
- Value：取决于记录类型

7.1 DNS域名系统

记录类型 TYPE

类型	意义	值
SOA	提供有关域名服务器区域、管理者的Email地址等信息	有关该区域的一组参数
A	主机的IP地址	32位整型数
MX	邮件服务器	邮件服务器名
NS	域名服务器	该域的域名服务器名
CNAME	别名	域名
PTR	指针	某一IP地址的别名，用于反向解析
HINFO	主机描述	ASCII方式表示的CPU或OS
TXT	文本	任意字符串



7.1 DNS域名系统

资源记录实例

荷兰VRIJE 大学计算机系DNS数据库可能的权威记录信息

域名	生存期	类	类型	说明
cs.vu.nl	86400	IN	SOA	star boss (9527,7200)
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	MX	1 flits.cs.vu.nl
ftp.cs.vu.nl	86400	IN	CNAME	2 zephyr.cs.vu.nl
Rowboat		IN	A	130.37.62.23

7.1 DNS域名系统

域名服务器

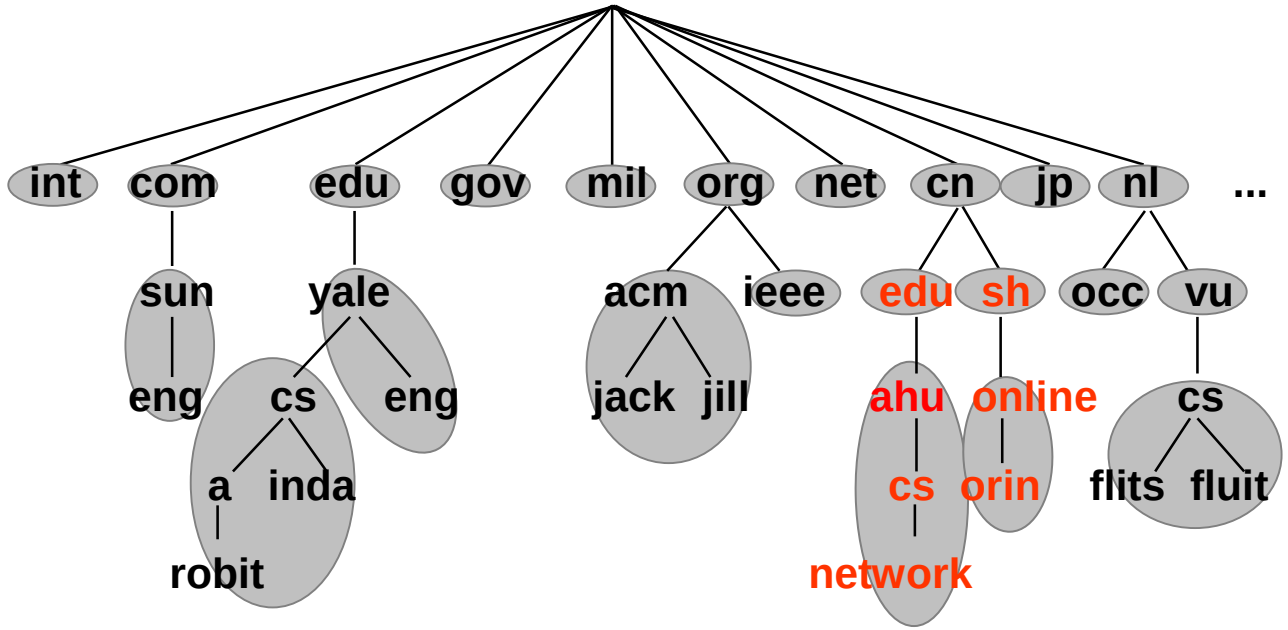
- DNS名字空间被划分为不重叠的区域，对应部署一定数量的DNS服务器
- 一台域名服务器必须负责自己所管辖区域的所有主机的域名解析，也必须知道上一层域名服务器的IP地址
- 域名服务器有以下四种类型

根域名服务器

顶级域名服务器

权限域名服务器

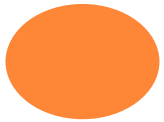
本地域名服务器



7.1 DNS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。本地域名服务器如果对因特网上任何一个域名无法解析，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,总共部署了数百台主机，分布在世界各地，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
- 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



7.1 DNS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）

- **权限域名服务器**：负责一个区域的域名服务器。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。

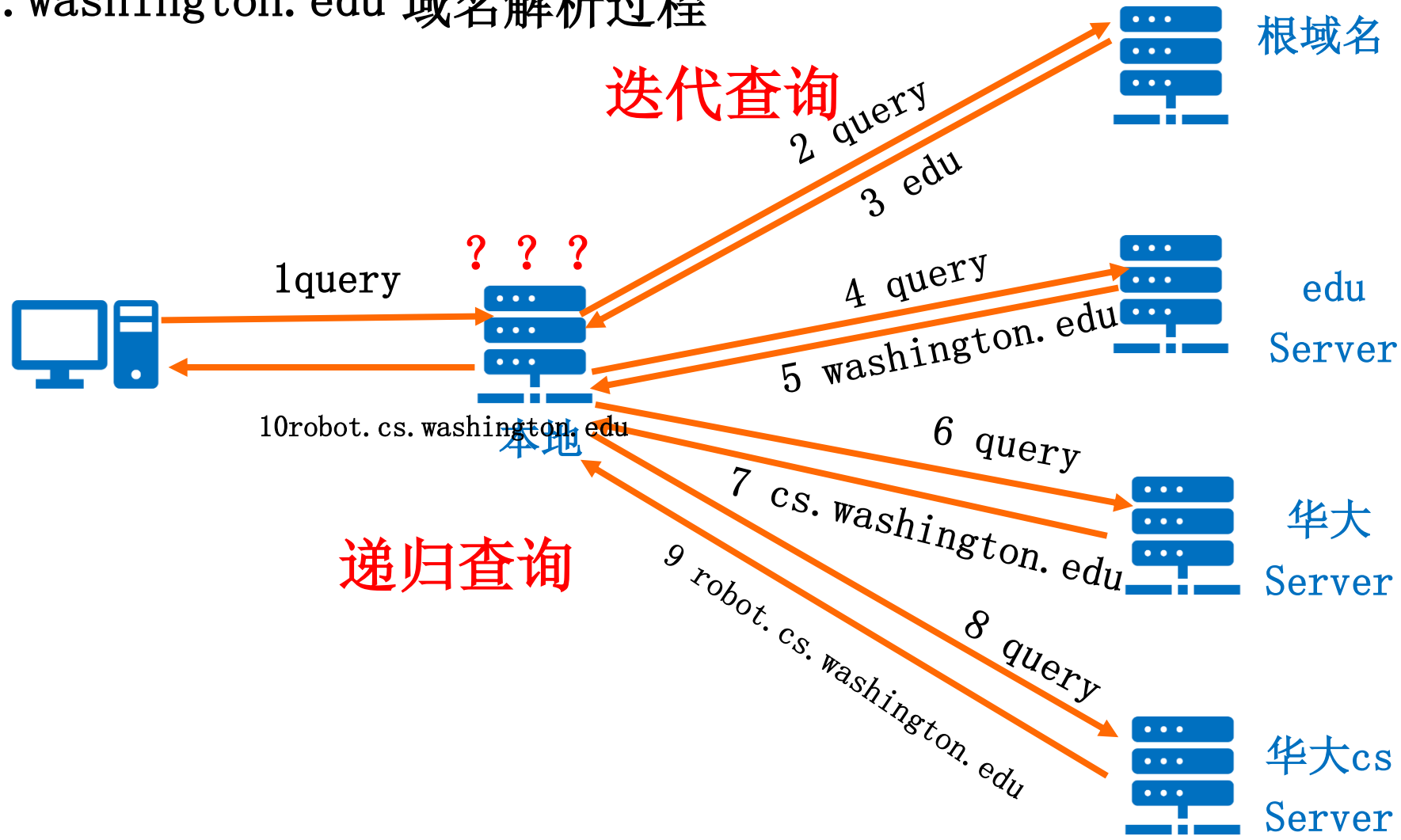
- **本地域名服务器**：离用户最近的域名服务器。当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器。每一个 ISP都可以拥有一个本地域名服务器，有时也称为默认域名服务器。

7.1 DNS域名系统

- 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。

7.1 DNS域名系统

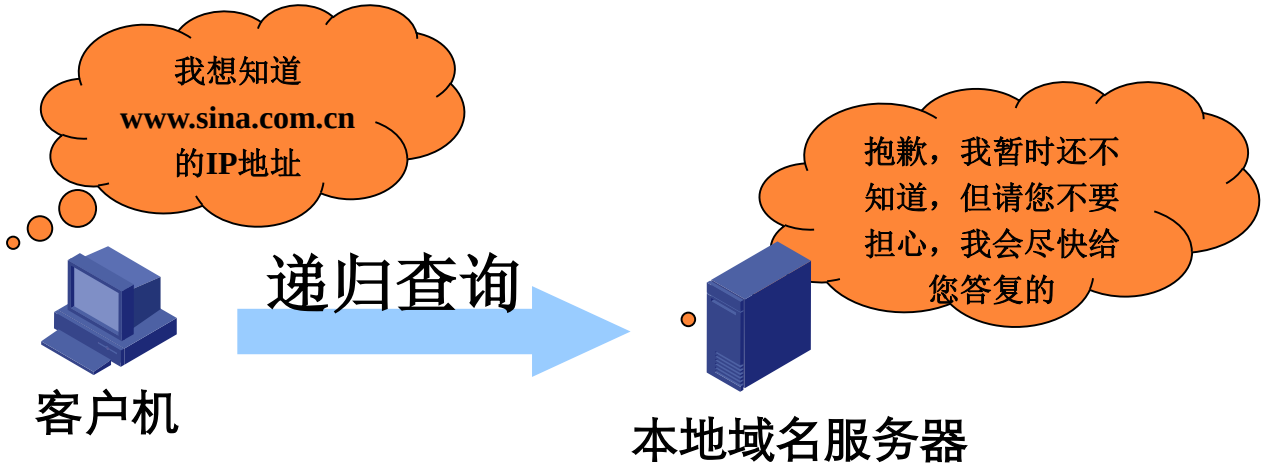
robot.cs.washington.edu 域名解析过程



7.1 DNS域名系统

递归查询

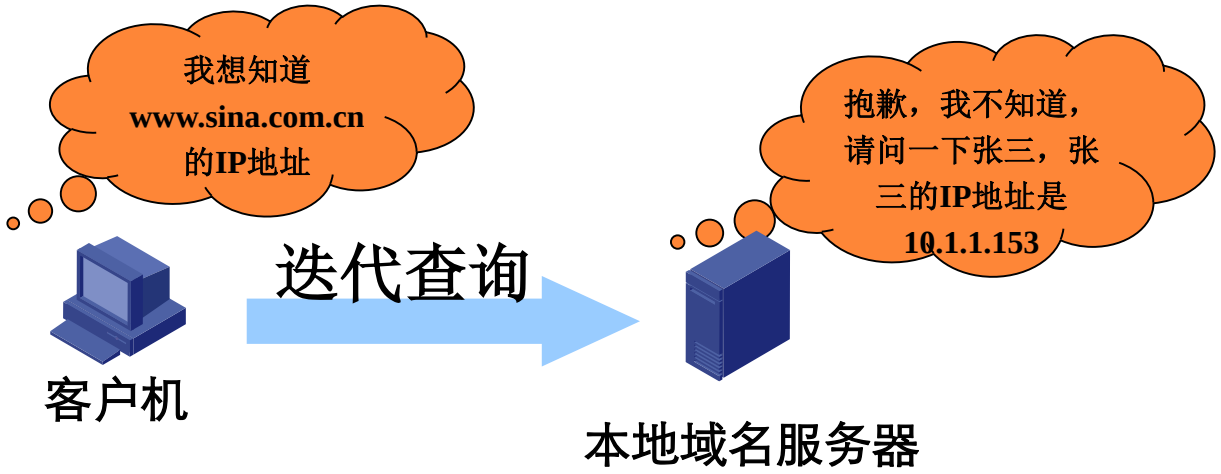
- 如果DNS服务器支持递归查询，那么当它接收到递归查询请求后，它将负责把最终的查询结果返回请求发送方。即使执行递归查询的DNS服务器无法从本地数据库返回查询结果，它也必须查询其他的DNS服务器，直到得到确认的查询结果
- 一般客户机与本地DNS域名服务器之间的查询交互采用的就是递归查询方式



7.1 DNS域名系统

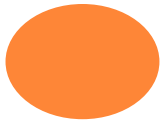
迭代查询

- DNS服务器接收到迭代查询请求后，如果无法从本地数据库返回查询结果，它会返回一个可能知道查询结果的DNS服务器地址给请求者，由请求者自行查询该DNS服务器，以此类推，请求者最终将得到查询结果
- 一般本地域名服务器发送至根域名服务器的查询采用的就是迭代查询



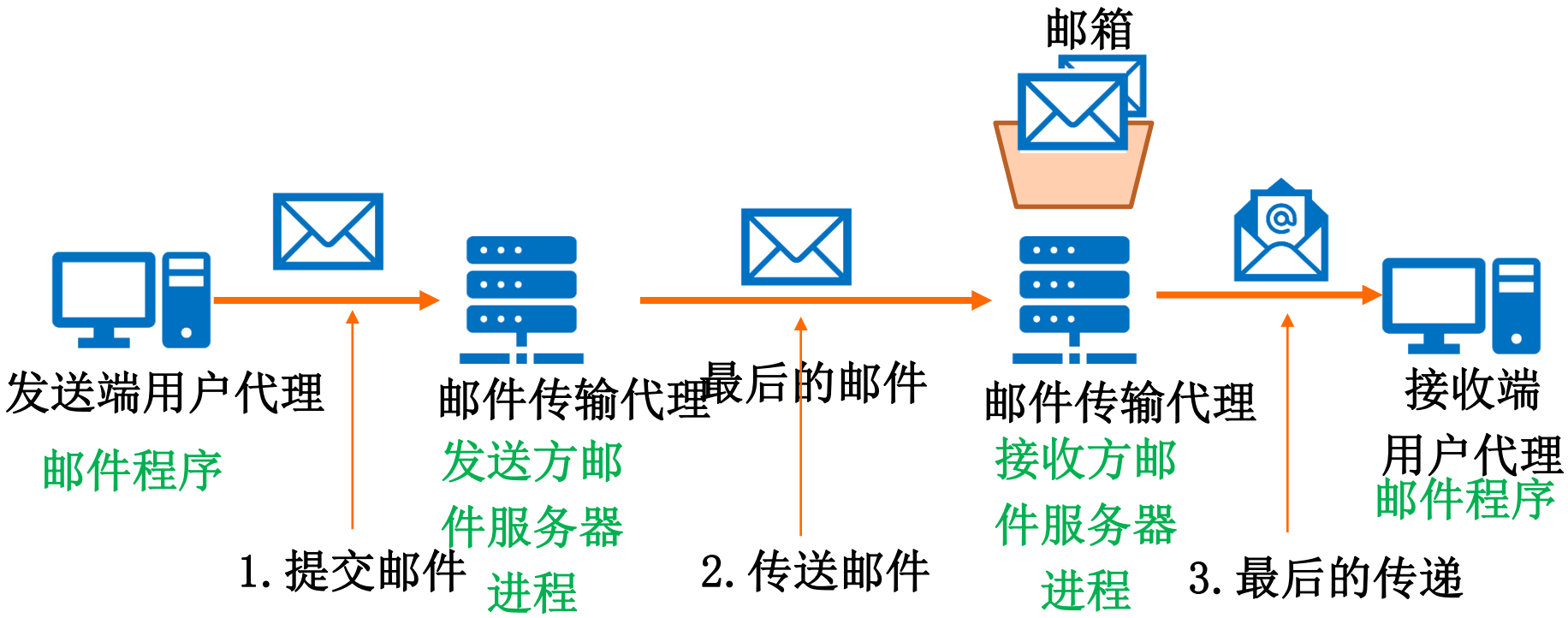
7.1 DNS域名系统

- 每个根服务器被复制许多份，存于整个世界，实际使用时，找物理上最近的服务器
- 每个服务器都有一个域名缓存，当查找到一个新的记录时，DNS将它的副本存于缓存中，此后如有域名解析请求，服务器优先查找缓存，但此信息不是权威（最原始）的，所以有一定生存期。
- DNS采用的是UDP协议，报文格式只有响应和查询。



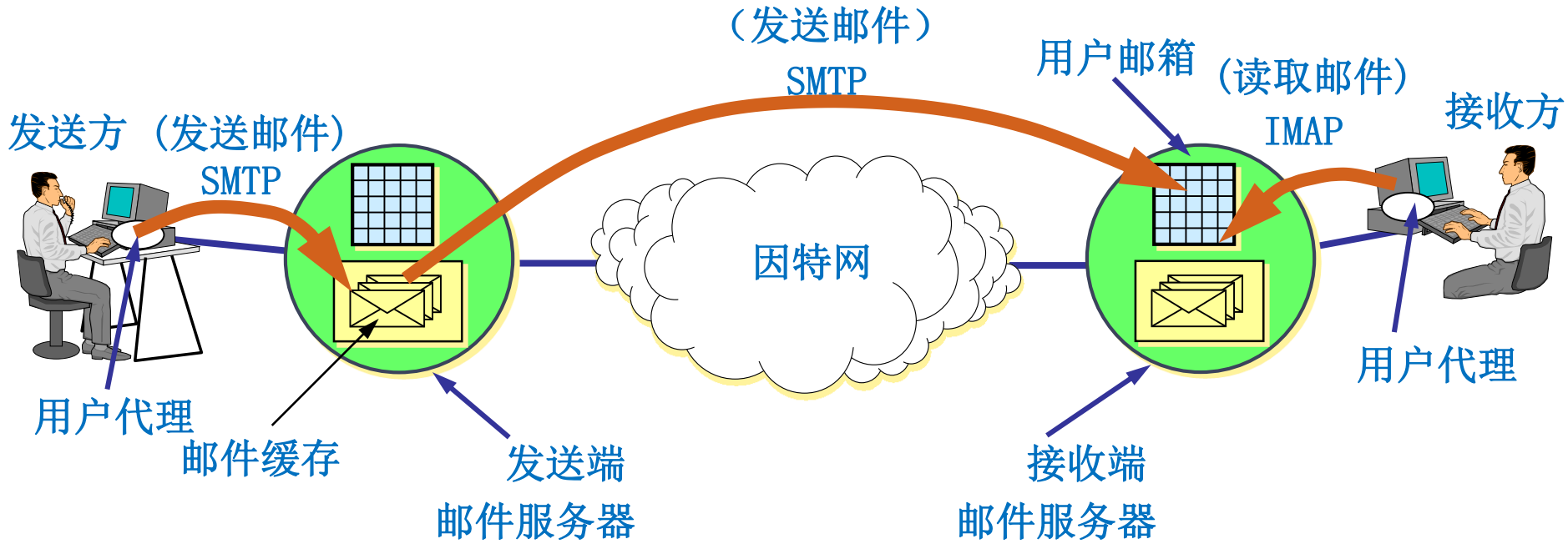
7.2 电子邮件

- 电子邮件体系结构包括用户代理和邮件传输代理两部分



7.2 电子邮件


发件方
用户代理
发送
邮件
SMTP
TCP
连接
发送方
邮件服务器
SMTP
客户
SMTP
服务器
发送邮件 SMTP
TCP 连接
接收方
邮件服务器
IMAP
服务器
SMTP
服务器
读取
邮件
IMAP
TCP
连接
接收方
用户代理
IMAP
客户


（发送邮件）
SMTP
用户邮箱
（读取邮件）
IMAP
接收方
发送方（发送邮件）
SMTP
因特网
用户代理
邮件缓存
发送端
邮件服务器
接收端
邮件服务器
用户代理

7.2 电子邮件

- 用户代理
  - 用户代理是一个本地程序，提供行命令方式、菜单方式或图形方式的界面，用于收发和管理电子邮件
  - 具有撰写、显示和处理功能，在常用的浏览工具中都带有电子邮件收发器
  - 常见的包括Gmail，QQ邮箱，网易邮件大师，Outlook，IE、foxmail等
  - 邮件的格式是 用户@dns地址（邮件服务服务器）

7.2 电子邮件

- 邮件格式
  - 邮件格式包括两个协议
  - RFC5322（之前的RFC822）
  - MIME

7.2 电子邮件

邮件格式

- RFC5322:由基本的信封，头字段，空行和邮件体组成
- 与邮件传输相关的头字段如下

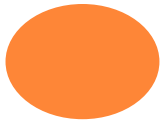
头字段	意义
To:	主要收件人的地址
Cc:	次要收件人的地址
Bcc:	密件抄送的地址
From:	发送者名字
Sender:	发送者的地址
Received:	沿途每一个转发者增加的信息行
Return-Path:	标识返回发送者的路径

7.2 电子邮件

- 邮件格式

用户代理或接收者使用的头部字段

头字段	含义
Date:	发送消息的日期和时间
Reply-To:	回复邮件应采用的电子邮件地址
Message-ID:	以后引用此消息的唯一标识号
In-Reply-To:	回信消息的标识号
References:	其它相关消息的标识号（一个或多个）
Keywords:	用户选择的关键字
Subject:	用于一行显示的本消息的简短摘要



7.2 电子邮件

➢ 邮件格式

- 早期E-mail的缺陷:只能传送以ASCII代码保存的文本信息（适合于英语国家）
- 当 Internet 在全球广泛应用时：
  - RFC822仅限于7位ASCII码，许多非英语国家的文字无法传送
  - 不能传送可执行文件等二进制文件，包括音频、视频文件
  - 解决方案是提出了多用途Internet邮件扩展（MIME，Multipurpose Internet Mail Extensions）
- MIME基于RFC822，增加了结构和新的非ASCII的编码规则

7.2 电子邮件

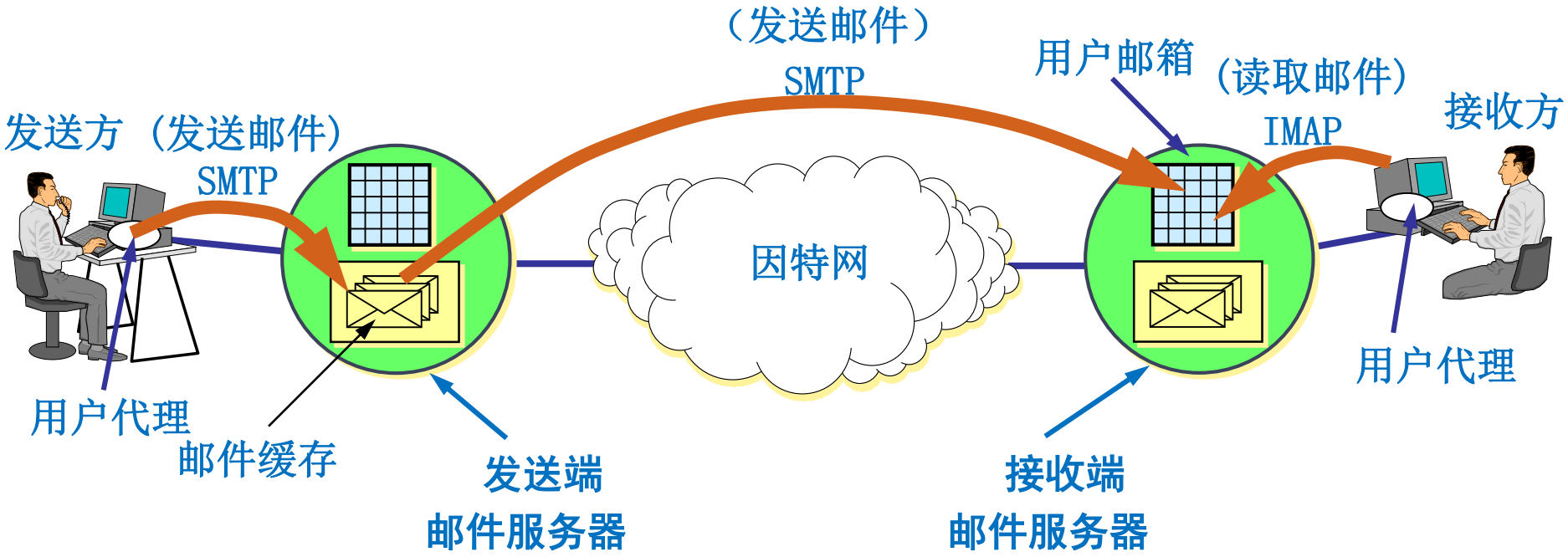
- 邮件格式

字段名	含义
MIME-Version：	MIME版本为1.0，表示服从RFC文档
Content-Description：	人能阅读的串，说明消息中的内容
Content-ID：	唯一的标识符
Content-Transfer-Encoding：	说明传送时如何包装消息主体
Content-Type：	说明内容的类型和格式

7.2 电子邮件

邮件传送

- 邮件传送采用的是简单邮件传输协议（Simple Mail Transfer Protocol，SMTP），负责用户向服务器提交邮件的通信传输，以及邮件传输代理之间的邮件传送通信。



7.2 电子邮件

邮件传送

- 当邮件传输程序需向远程服务器发送邮件时，将建立一个TCP连接（端口号为25）并通过该连接传输电子邮件信息
- SMTP是两个报文传输代理之间的通信协议，它有14条命令和21条应答信息，每条命令用四个字母组成，一般情况下一条命令只有一行，由三位数字的代码开始，后面附上简单的文字说明 。
- SMTP的通信包括建立连接，邮件传送和释放连接三个部分。

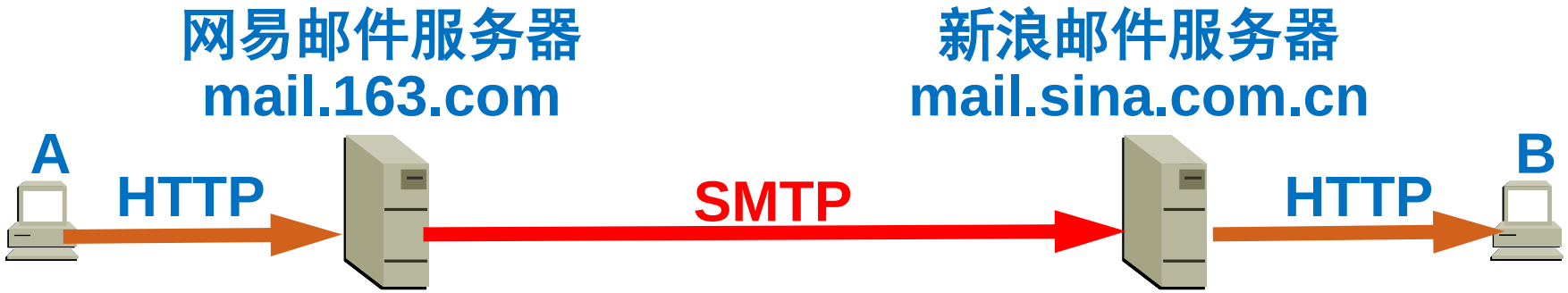
7.2 电子邮件

最后递交

- 邮件的提交和传送都采用SMTP协议，接收方必须一直在线，确定能够建立连接并接收信息。
- 但如果接收方不在线，就无法建立连接，不能发送邮件。
- 所以邮件的最终交付未采用SMTP协议，而是使用internet邮件访问协议（IMAP，Internet Message Access Protocol），是对邮局协议3（POP3）协议的改进版本。端口号143.
- 协议思想是在ISP的机器上设置一个消息传输代理（邮局）来替代用户接收邮件，并把它存入他们的邮箱中，当用户在线并有请求时再发给用户。邮件服务器即 “ 邮局 ”。

7.2 电子邮件

- 通过Web站点提供E-mail服务
- 服务器上有正常的信息传输代理来监听端口25，以及时应答SMTP用户的连接请求
- 在连接建立后，通过用户名和口令的输入，验证身份
- 由服务器读取用户信箱，并组装成HTML的网页发给用户



7.3 万维网

- WWW（World Wide Web）万维网是分布式的超媒体信息系统，是超文本的扩展。
- 超文本：由多个文本信息源连接而成，通过链接，用户可以找到其他文档
- 超媒体：与超文本的区别是文档内容不同，超媒体信息可包括声音、图象、活动图象等

  1991年欧洲原子能研究中心，网景公司，浏览器

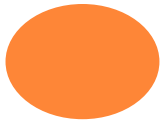
7.3 万维网

万维网的体系结构

基于客户/服务器结构，客户端称为浏览器，用TCP作为传输层协议.

服务器用固定端口80/8080侦听连接请求

使用超文本传输协议（HTTP）



7.3 万维网

- 客户端软件即浏览器，主要用于：

  连接Web server

  解释执行由HTML语言编写的文档

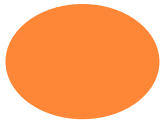
  将执行结果显示在屏幕上

- 使用统一资源定位符URL（Uniform Resource Locator）标识和访问页面

- URL一般格式：

  protocol://computer_name:port/document_name

  （其中port 通常不用 ）



7.3 万维网

当点击了http:// www.cs.washington.edu/index.html

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.cs.washington.edu的IP地址

3.DNS解析出服务器的IP地址为128.208.3.88

4.浏览器与服务器建立TCP连接（128.208.3.88 使用端口80）

5.浏览器发出取HTTP报文：请求/index.html

6.www.cs.washington.edu服务器响应HTTP，将文件index.html发给浏览器

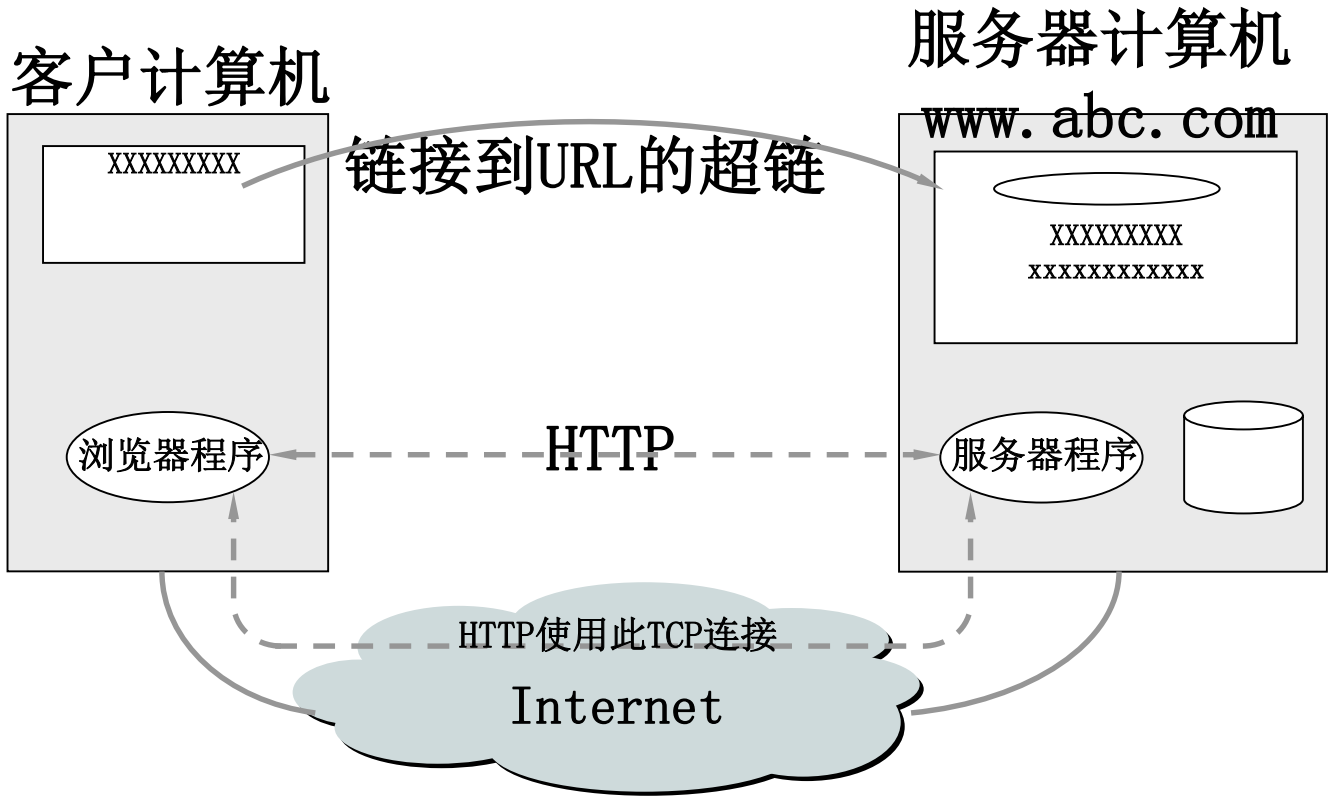
7.如果该页面包含其他URL，浏览器经同样处理获取其他URL包含内容 （视频、图像等）

8.浏览器显示文件index.html中的所有内容

9.如果没有其他请求，TCP连接释放

7.3 万维网

超文本传输协议HTTP工作过程



7.3 万维网

当点击了http:// www.cs.washington.edu/index.html

服务器端

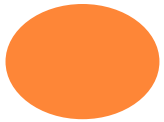
1、接受来自客户端浏览器的请求

2、获取页面的路径，即被请求的文件的名字

3、从本地磁盘上获取文件（或者从缓存中获取）

4、将文件内容发送给客户

5、释放TCP连接



7.3 万维网

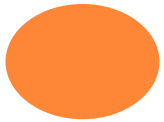
Cookie

一般情况下，客户从服务器断开后，服务器会忘记该用户。但是对于一些新的需求，例如注册用户、个性化设置等，需要让服务器记住用户之前的状态。

使用IP是难以实现的，因此引入了Cookie文件

服务器为用户产生一个唯一的识别码（Cookie文件）。利用此识别码，网站就能够跟踪该用户在该网站的活动。

当客户在访问服务器时，服务器会记录客户的状态并创建对应的Cookie发送给客户，由浏览器保存在本地。当浏览器访问该Web站点时，会检查本地是否包含Cookie，并发送给服务器。



7.3 万维网

Cookie

- 一当客户请求一个web页面时，服务器除了提供所请求的页面以外，还以Cookie的形式提供了一些附加的信息。
- Cookie是一个相当小的命名的串（4K），服务器将它与浏览器关联，这种关联与用户关联不一样，但它非常接近，而且比IP地址更有用。
- 浏览器把服务器所提供的Cookie通常存储在客户机键盘Cookie目录下一段时间，这样在整个浏览器调用期间一直坚持Cookie ，除非用户禁用Cookie
- Cookie顾客只是字符串而不是可执行程序，可能包含病毒，黑客有可能会利用浏览器漏洞来激活病毒，

7.3 万维网

Cookie

域	路径	内容	过期	安全
toms-casino.com	/	CustomerID=297793521	15-10-1017:00	是
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-1114:22	不
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-2023:59	不
sneaky.com	/	UserID=4627239101	31-12-19 23:59	不

图 7-22　Cookie 的某些例子

一个cookie可能包含至多5个字段：

- 1、域：指出Cookie来自何方。浏览器应该检查服务器没有谎报域名，每个域为每个客户端应该存储不超过20个Cookie。
- 2、路径：服务器目录结构中的一个路径，它标识服务器文件树的哪些部分可能使用该Cookie。路径通常是/，意味着整棵树。

7.3 万维网

Cookie

域	路径	内容	过期	安全
toms-casino.com	/	CustomerID=297793521	15-10-1017:00	是
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-1114:22	不
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-2023:59	不
sneaky.com	/	UserID=4627239101	31-12-19 23:59	不

图 7-22　Cookie 的某些例子

一个cookie可能包含至多5个字段：

- 3、内容字段：存放cookie的内容，采用“名字=值”的形式。名字和值可以是服务器期望的任何内容。
- 4、过期时间自断制定来该cookie何时过期。若不存在，则在退出时丢弃cookie—非持续性cookie。非持续性cookie和持续性cookie通过有没有提供过期时间和日期来区分。
- 5、安全字段：浏览器只向使用安全传输连接的服务器访问cookie所谓的安全传输就是SSL/TLS。

7.3 万维网

Cookie

域	路径	内容	过期	安全
toms-casino.com	/	CustomerID=297793521	15-10-1017:00	是
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-1114:22	不
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-2023:59	不
sneaky.com	/	UserID=4627239101	31-12-19 23:59	不

图 7-22　Cookie 的某些例子

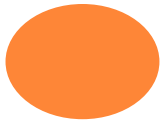
一个cookie可能包含至多5个字段：

- 5、安全字段：浏览器只向使用安全传输连接的服务器访问顾客所谓的安全传输就是SSL/TLS。这个功能可用于电子商务银行和其他安全的应用将在第6章第8章开始，

7.3 万维网

Cookie的使用

- 在浏览器下某个Web站点发出一个页面请求之前，浏览器检查它的Cookie目录，确定这个请求前往的目标域是否在当前客户端放置了Cookie。
- 如果存在相应的Cookie，则该域放置的所有Cookie都被包含到请求消息中。
- 服务器得到这些Cookie以后就可以按他所期望的方式来解释它们。



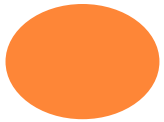
7.3 万维网

Cookie—可能的用法

域	路径	内容	过期	安全
toms-casino.com	/	CustomerID=297793521	15-10-1017:00	是
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-1114:22	不
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-2023:59	不
sneaky.com	/	UserID=4627239101	31-12-19 23:59	不

图 7-22　Cookie 的某些例子

- Toms-casino.com设置的cookie，记录的内容是用户id，标识用户。当顾客下次登陆并进行相应的操作后，浏览器将Cookie发送过去，因此服务器就能确定这个顾客。并可以查找顾客的相关消费记录，并利用这些记录信息来创建一个合适的Web页面显示给该顾客。



7.3 万维网

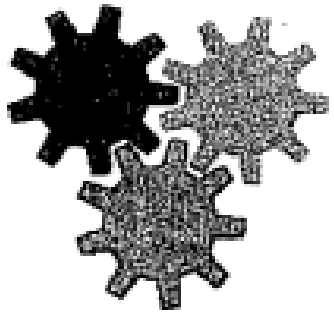
- 静态web页面：存放在服务器上，每次被客户端获取和显示表现的都是一样的方式。注意视频动画页面也可以是静态页面
- HTML（HyperText Markup Language）允许用户创建包含文本、图像及指向其它Web页面的指针
- 标记语言的关键优点：将内容与其应该如何表示相分离。使得编写浏览器就非常简单，浏览器只需理解标记命名，并将这些命令应用于内容，即可在每个HTML的文件上中嵌入所有的标记命令，并且标准化，这些标记命令使得任何一个Web浏览器都可以读取任何页面，并对页面重新格式化，这点非常关键，因为一个页面有可能是一台具有1600*1200大小24位颜色窗口的高端计算机上设计出来的，那它必须能在一个移动电话只有640×320大小8位颜色的窗口中显示，

7.3 万维网

```
<html>
<head> <title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page </h1>
<img src="http://www.widget.com/images/logo.gif" ALT="AWI Logo"> <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's</b>
home page. We hope <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 email. </p>
<hr>
<h2> Product information </h2>
<ul>
    <li> <a href="http://widget.com/products/big"> Big widgets </a> </li>
    <li> <a href="http://widget.com/products/little"> Little widgets </a> </li>
</ul>
<h2> Contact information </h2>
<ul>
    <li> By telephone: 1-800-WIDGETS </li>
    <li> By email: info@amalgamated-widget.com </li>
</ul>
</body>
</html>
```

7.3 万维网

Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope you will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by email.

Product Information

- Big widgets
- Little widgets

Contact Information

- By telephone: 1-800-WIDGETS
- By email: info@amalgamated-widget.com

7.3 万维网

项目	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
超链接	×	×	×	×	×
图像	×	×	×	×	×
列表	×	×	×	×	×
活动地图和图像		×	×	×	×
表单		×	×	×	×
方程式			×	×	×
工具条			×	×	×
表格			×	×	×
访问功能				×	×
对象嵌入				×	×
风格				×	×
样式表				×	×
脚本				×	×
视频和音频				×	×
内联矢量图型					×
XML表示					×
后台线程					×
浏览器存储					×
画曲线					×

图 7-24　HTML 版本之间的某些差异

7.3 万维网

- 输入和表单

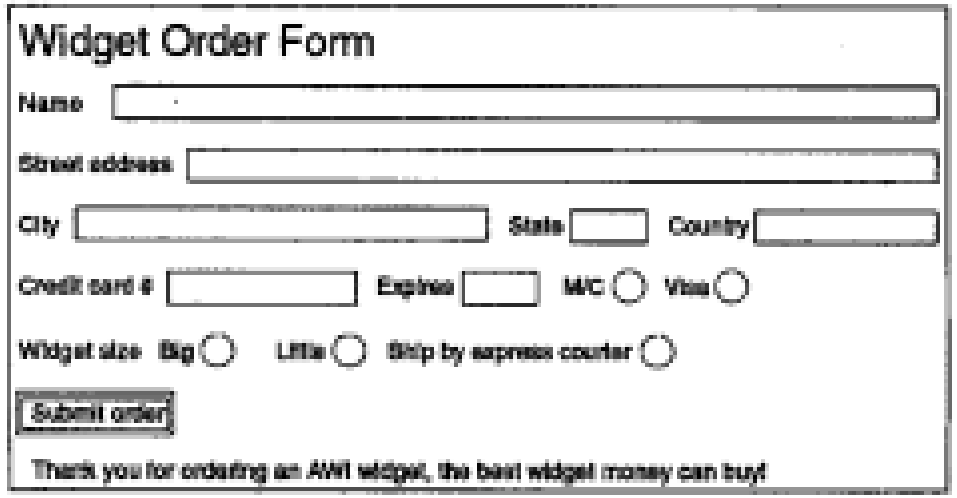
- 1、html1.0基本上是单向的：用户可以从信息提供者获取网页，但很难在另一个方向将信息发送过去。
- 2、双向通信的需求变得很明显，当用户的输入发送到服务器需要两种支持：

（1）他要求http可以在这个方向携带数据，使用post方法。

（2）能够提出用户界面元素用来收集和封装收入的数据，而且HTML2.0，具备了这种功能，并包括了表单。

7.3 万维网

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/order.cgi" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="Submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```




```
customer=John+Doe&address=100+Main+St.&city=White+Plains&
state=NY&country=USA&cardno=1234567890&expires=6/14&cc=mastercard&
product=cheap&express=on
```

图 7-26　浏览器对服务器的一种可能响应，包括了用户填入的信息

7.3 万维网

- CSS---层叠样式表

网页设计师希望能绝对控制自己设计的网页如何呈现

1、会将新的标签添加到html来控制页面的外观。此外还添加了屏幕上准确定位的控制方法，产生来不可移植的臃肿html

2、更好的替代方案是使用样式表来解决。

3、层叠样式表将样式表引入到html4.0的Web，Css定义了一种简单的语言，用来描述控制标签内容，标签内容外观的规则，

4、样式表表可以被放置在htm文件中，但更常见的做法是将它们放置在一个单独的文件中，然后引用它们。

```
body {background-color:linen; color:navy; font-family:Arial;}
h1 {font-size:200%;}
h2 {font-size:150%;}
```

```
<head>
<title> AMALGAMATED WIDGET, INC. </title>
<link rel="stylesheet" type="text/css" href="awistyle.css" />
</head>
```

图 7-28　包括了一个 CSS 样式表

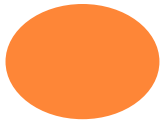
7.3 万维网

- CSS---层叠样式表

将CSS放置在一个单独文件中，在HTML使用时引用的两大优势：

1、可以使得一组样式被应用到一个网站上的许多网页，这种组织方式带来了页面外观上的一致性。

2、下载的HTML问句保持得很小。浏览器只需要下载CSS文件的一个副本供所有引用它的页面使用。



7.3 万维网

7.3.3 动态Web页面和Web应用

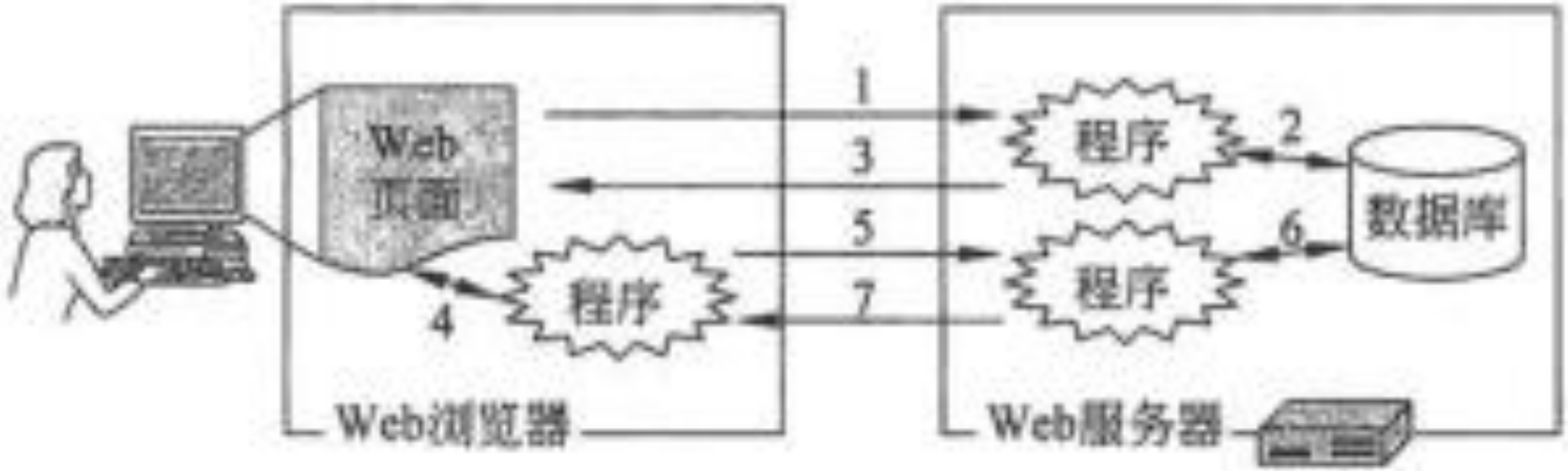
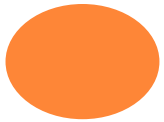


图 7-29　动态页面

7.3.3 动态Web页面和Web应用

- 服务器端的动态网页的生成：服务器根据用户提交的表单数据，从数据库中读取数据并组织成网页，并发送给用户。或者服务器执行HTML页面中的脚本，并生成结果返回给用户。
- 客户端的动态网页的生成，如在客户端利用一段SCRIPT程序生成一些预定格式的网页。



7.3.3 动态WEB页面和WEB应用

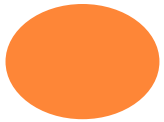
- 动态页面处理的方法：

1、公共网关接口，RFC 3875建议，但是自从有了WEB就一直可用的方法，CGI提供了一个接口，允许外部服务器与后端程序及脚本通信这些后边程序和脚本接受输入的信息并生成html，页面作为响应。

2、API方法：在html页面中嵌入少量的脚本，然后让服务器来执行这些脚本，以便生成最终发送给客户的页面。

编写这些脚本的一种流行语言是超文本预处理器PHP，PHP的使用类似于CGI

另外还有JSP、ASP.NET等都可以用来作为生成动态HTML页面的不同方法。



7.3.3 动态Web页面和Web应用

```
<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)

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

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 33
</body>
</html>
```

(c)

图 7-30

(a) 包含了一个表单的 Web 页面；(b) 处理该表单输出的 PHP 脚本；
(c) 当输入分别是“Barbara”和“32”时 PHP 脚本的输出

7.3.3 动态WEB页面和WEB应用

- 客户端的动态网页的生成，如在客户端利用一段SCRIPT程序生成一些预定格式的网页。最流行的客户端脚本语言是JavaScript

```
<html>
<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>
```

图 7-31　使用 JavaScript 处理表单

7.3.3 动态WEB页面和WEB应用

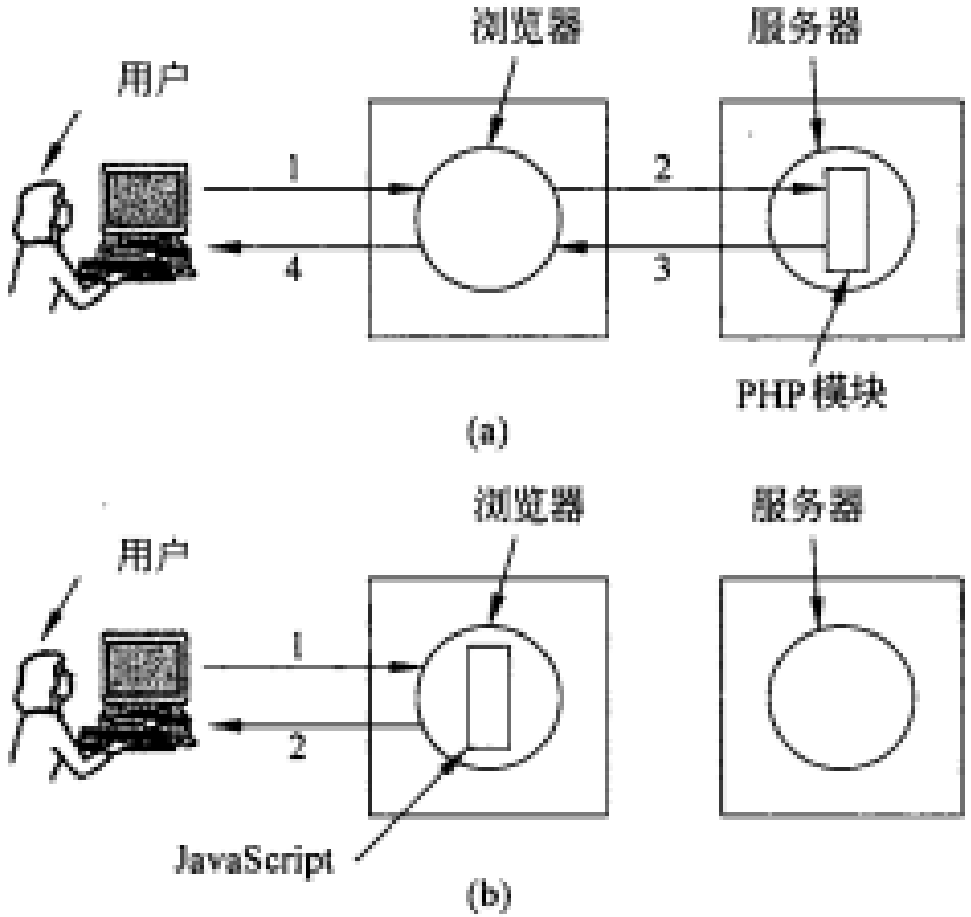


图 7-32

(a) 使用 PHP 的服务器端脚本；(b) 使用 JavaScript 的客户端脚本

7.3.3 动态WEB页面和WEB应用

AJAX——异步 JavaScript 和 XML

引人注目的 Web 应用程序需要用户界面具备可响应能力，而且能无缝访问存储在远程 Web 服务器上的数据。客户端上的脚本（例如，使用 JavaScript）和服务器上的脚本（比如 PHP）只是提供某种解决方案的基本技术，这些技术通常与其他几个关键技术结合起来一起使用，这些技术的组合就称为异步 JavaScript 和 XML（AJAX，Asynchronous JAvascript and Xml）。许多全功能的 Web 应用，比如谷歌的 Gmail、地图和文档都是以 AJAX 编写的。

AJAX 有点混乱，因为它不是一种语言。它是一组需要一起协同工作的技术，正是这些技术使得 Web 应用程序和传统的桌面应用一样能响应用户的每个动作。这些技术包括：

（1）用来表现页面信息的 HTML 和 CSS。

（2）浏览时改变部分页面的 DOM（Document Object Model）。

（3）使得程序与服务器交换应用数据的 XML（eXtensible Markup Language）。

（4）程序发送和检索 XML 数据的异步方式。

（5）将所有功能组合在一起的 JavaScript。

7.3.3 动态WEB页面和WEB应用

- DOM文档对象模型是通过程序访问html页面的一种表示，这种表示结构化成一颗反映HTML元素的树

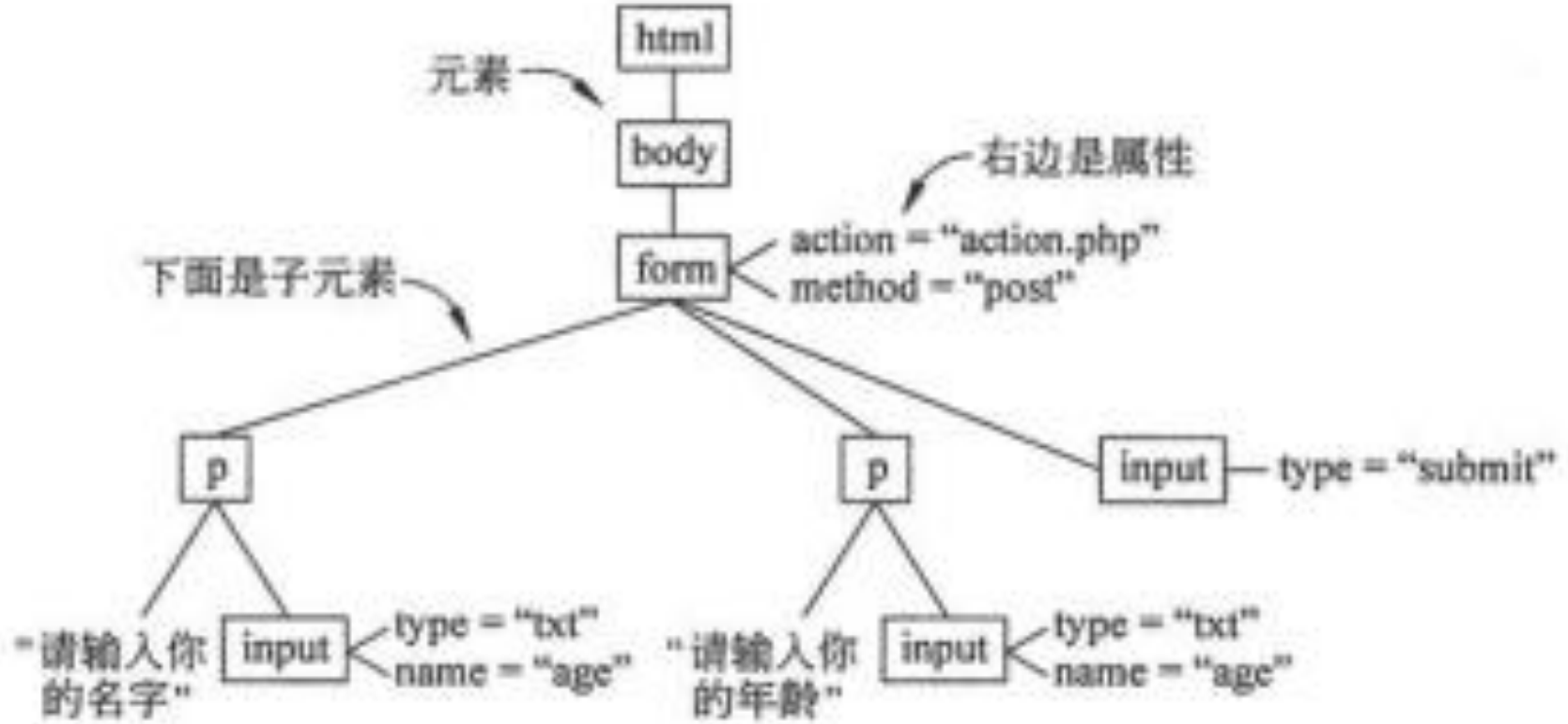


图 7-33　如图 7-30(a)所示 HTML 的 DOM 树

7.3.3 动态Web页面和Web应用

- XML（eXtensible Markup Language）允许定义结构化的数据

```
<?xml version="1.0" ?>
<book#list>
<book>
<title> Human Behavior and the Principle of Least Effort </title>
<author> George Zipf </author>
<year> 1949 </year>
</book>
<book>
<title> The Mathematical Theory of Communication </title>
<author> Claude E. Shannon </author>
<author> Warren Weaver </author>
<year> 1949 </year>
</book>
<book>
<title> Nineteen Eighty-Four </title>
<author> George Orwell </author>
<year> 1949 </year>
</book>
</book#list>
```

图 7-34　一个简单的 XML 文档

7.3.3 动态WEB页面和WEB应用

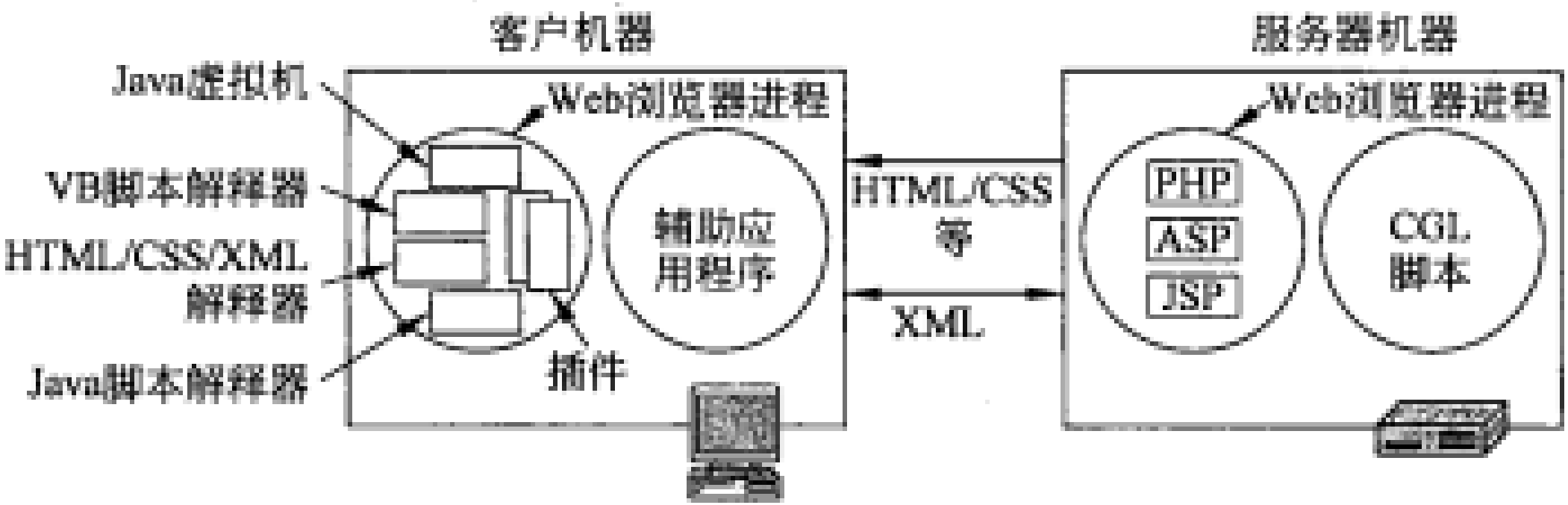
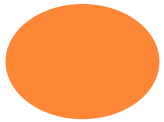


图 7-35　用来生成动态页面的不同技术

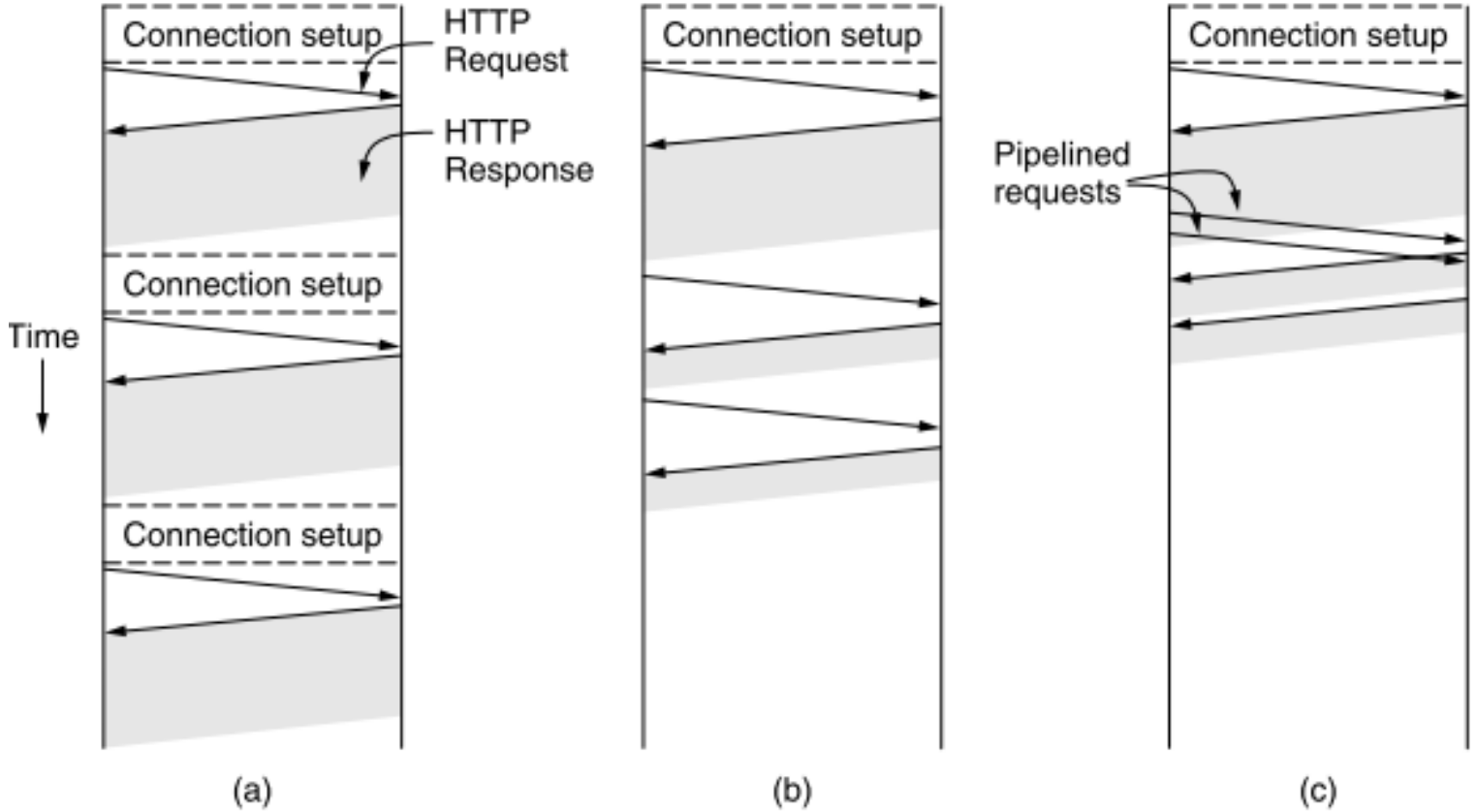
7.3.4超文本传输协议

- HTTP超文本传输协议
- HTTP是一个简单的请求-响应协议，基于TCP协议。
- 早期WEB页面只有文本，使用HTTP1.0协议，建立连接后只发送一个请求。获取内容后，释放连接。
- 现在的WEB页面上包含很多元素，并且可能处于服务器不同目录下，采用HTTP1.1协议，利用持续连接技术，建立一个TCP连接，可以发送多次请求和响应。还可以发送流水线请求。



7.3.4超文本传输协议

- HTTP超文本传输协议—连接



7.3.4超文本传输协议

- HTTP超文本传输协议—方法

方法	描述
GET	读取一个Web页面
HEAD	读取一个Web页面的头
POST	附加一个Web页面
PUT	存储一个Web页面
DELETE	删除一个Web页面
TRACE	回应入境请求
CONNECT	通过代理连接
OPTIONS	一个页面的查询选项

图 7-37　内置的 HTTP 请求方法

7.3.4超文本传输协议

- HTTP超文本传输协议—响应组的状态码

代码	含义	例子
1××	信息	100＝服务器同意处理客户请求
2××	成功	200＝请求成功；204＝没有内容
3××	重定向	301＝移动页面；304＝缓存的页面仍然有效
4××	客户错误	403＝禁止页面；404＝页面没找到
5××	服务器错误	500＝服务器内部错误；503＝稍后再试

图 7-38　响应组的状态码

7.3 万维网

- HTTP超文本传输协议—消息头

头	类型	内容
User-Agent	请求	有关浏览器及其平台的信息
Accept	请求	客户可处理的页面类型
Accept-Charset	请求	客户可接受的字符集
Accept-Encoding	请求	客户可处理的页面编码
Accept-Language	请求	客户可处理的自然语言
If-Modified-Since	请求	检查新鲜度的时间和日期
If-None-Match	请求	先前为检查新鲜度而发送的标签
Host	请求	服务器的DNS名字
Authorization	请求	列出客户的信任凭据
Referer	请求	发出请求的先前URL
Cookie	请求	给服务器发回Cookie的先前URL
Set-Cookie	响应	客户存储的Cookie
Server	响应	关于服务器的信息
Content-Encoding	响应	内容如何编码（比如，gzip）
Content-Language	响应	页面使用的自然语言
Content-Length	响应	页面以字节计的长度
Content-Type	响应	页面的MIME类型
Content-Range	响应	标识了页面内容的一部分
Last-Modified	响应	页面最后修改的时间和日期
Expires	响应	页面不再有效的时间和日期
Location	响应	告诉客户向谁发送请求
Accept-Ranges	响应	指出服务器能接受的请求的字节范围
Date	请求/响应	发送消息的日期和时间
Range	请求/响应	标识一个页面的一部分
Cache-Control	请求/响应	指示如何处理缓存
ETag	请求/响应	页面内容的标签
Upgrade	请求/响应	发送方希望切换的协议

图 7-39　某些 HTTP 消息头

7.3 万维网

- HTTP超文本传输协议—缓存

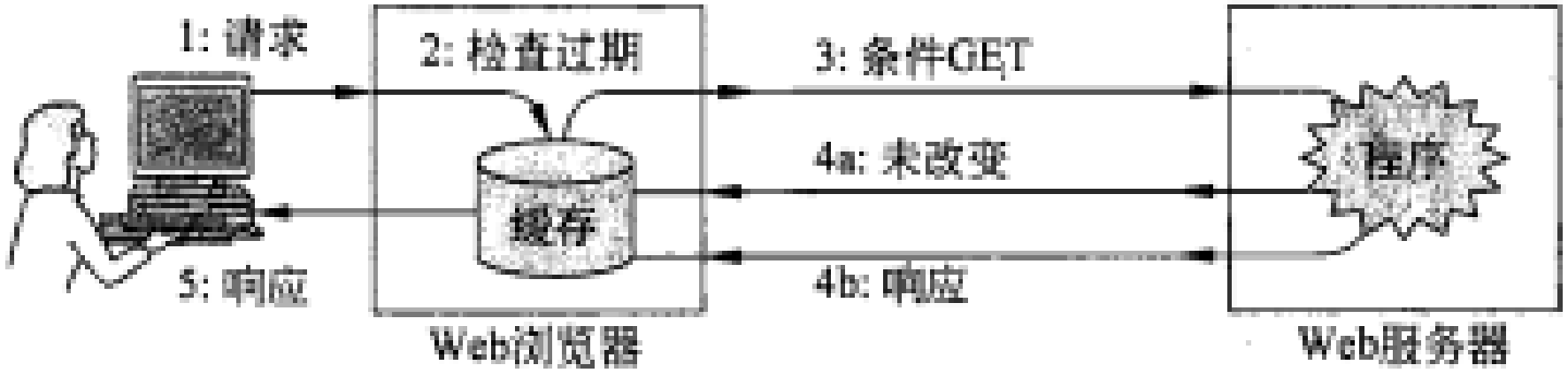


图 7-40　HTTP 缓存

7.3 万维网

- Web搜索

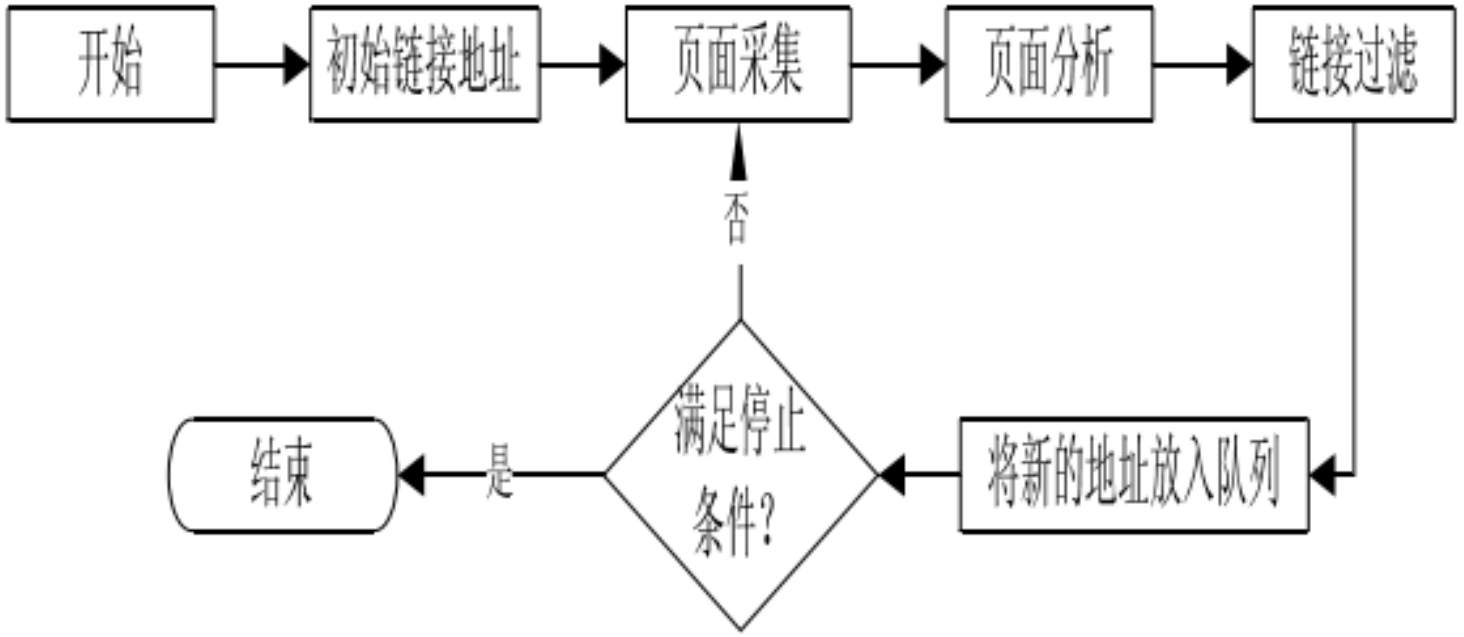


图　　通用网络爬虫工作流程

7.4 流式音视频

- 2000年开始实时音频和实时视频流量才有来真正极度的增长
- 实时流量和web流量的区别：必须以预先设定的速率播放才有用。
- Web可以有短暂的中断，而且页面加载可以花费更多或更少的时间，在一定限度内这不是主要的问题。
- 两件促进实时流量增长的事情：1、计算机更加强大，并且配备来麦克风和摄像头，能够输入、处理和输出音频和视频数据；2、Internet带宽扩张可以大大提高Internet的可用性

7.4 流式音视频

- 数字音频

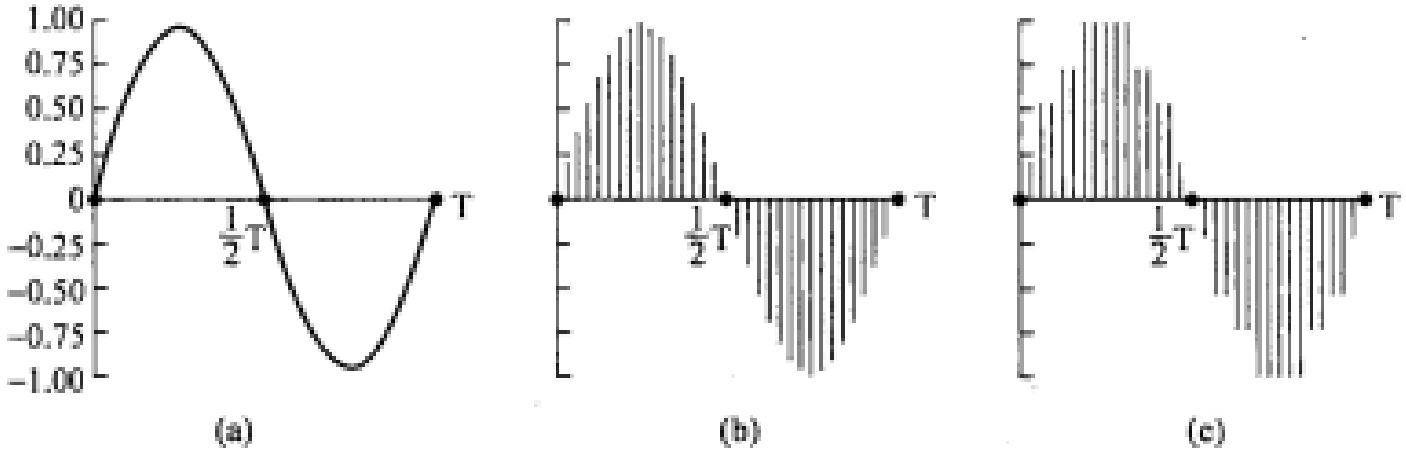


图 7-42

(a) 一个正弦波 (b) 正弦波样值 (c) 量化成 4 位

7.4 流式音视频

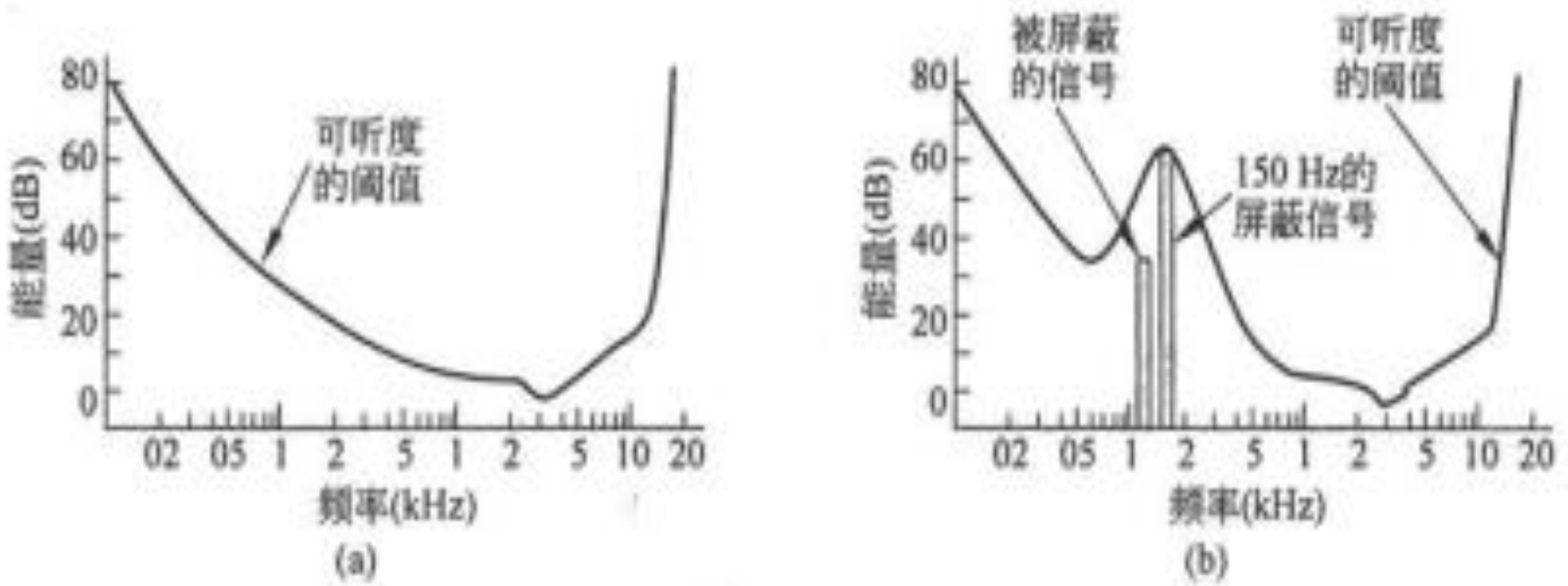


图 7-43

(a) 作为频率函数的可听度阈值；(b) 屏蔽的效果

7.4 流式音视频

- JPEG标准

$$Y = 16 + 0.26R + 0.50G + 0.09B$$

$$Cb = 128 + 0.15R - 0.29G - 0.44B$$

$$Cr = 128 + 0.44R - 0.37G + 0.07B$$

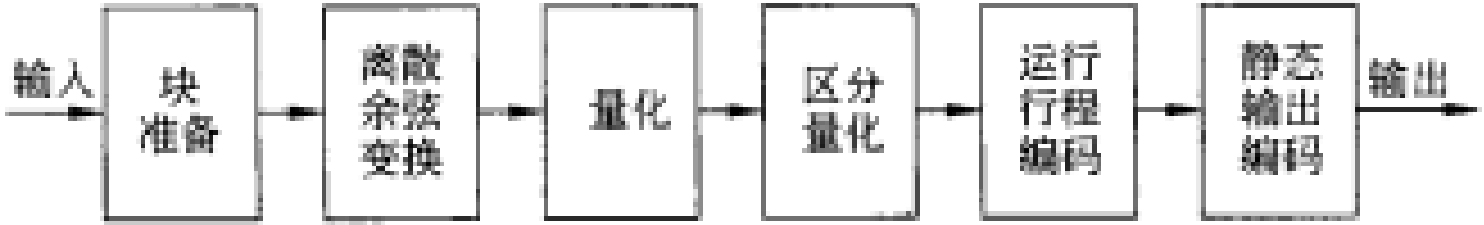


图 7-44　JPEG 有损顺序编码的步骤

7.4 流式音视频

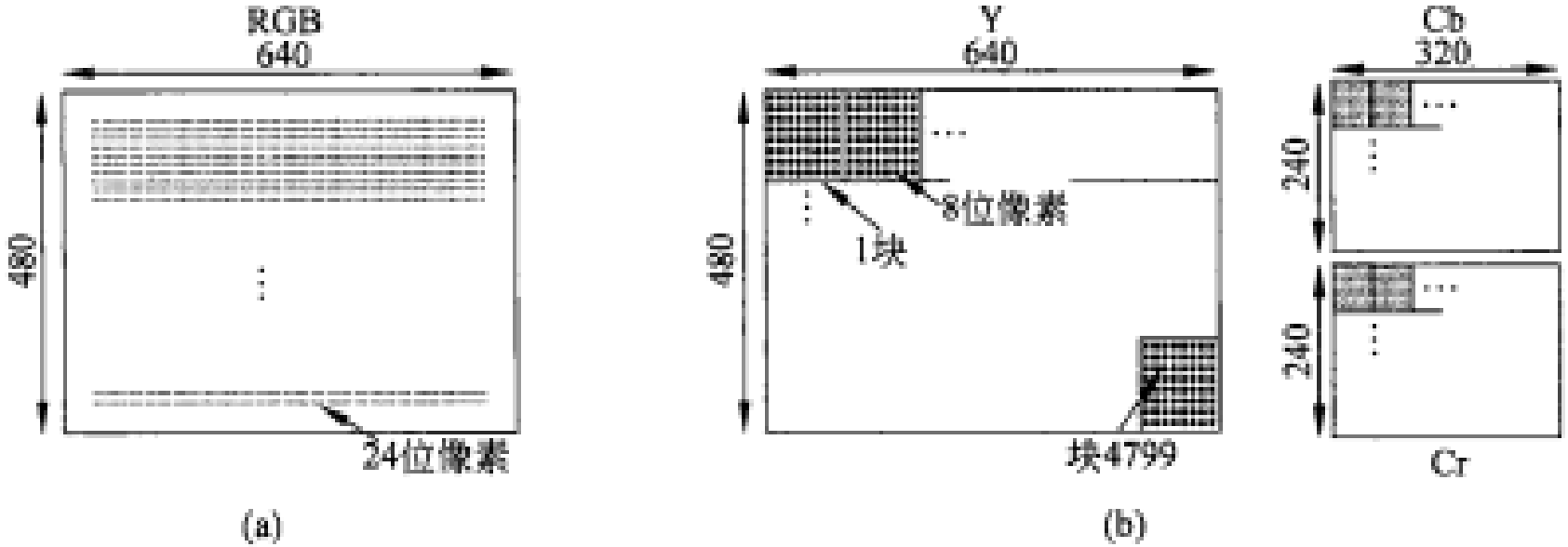


图 7-45

(a) RGB 输入数据；(b) 块准备之后

7.4 流式音视频

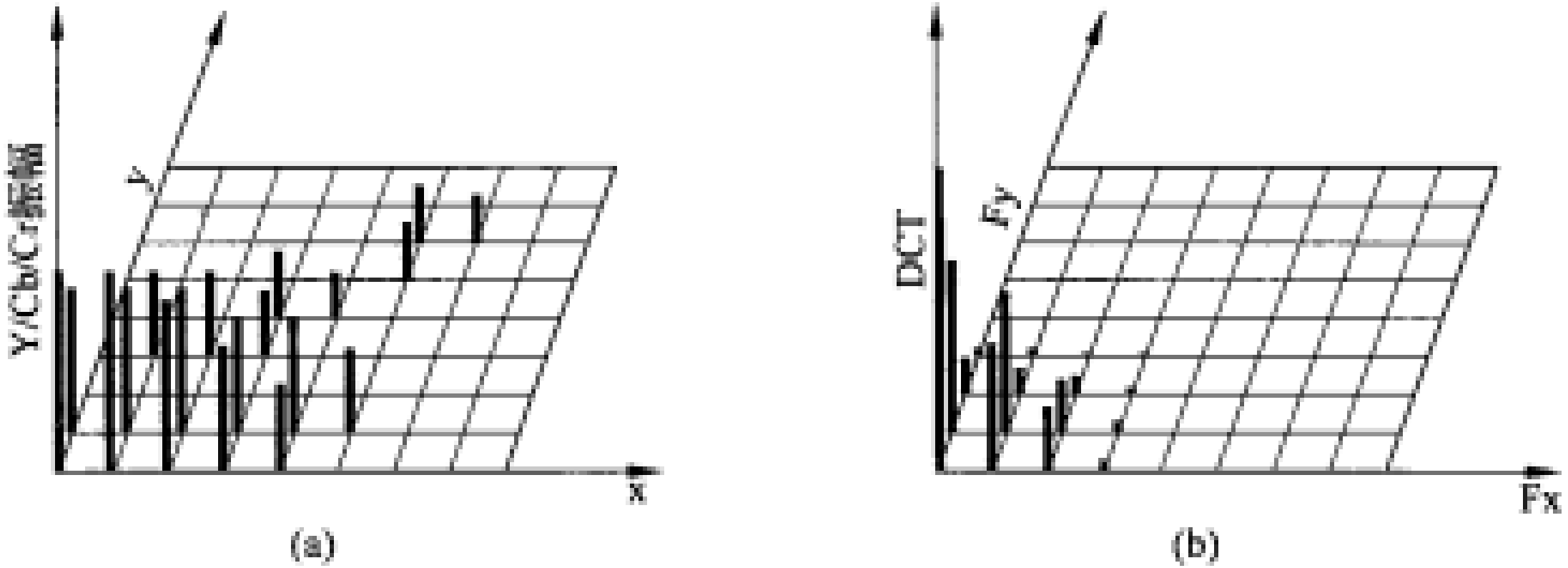


图 7-46

(a) Y 矩阵的一块；(b) DCT 系数

7.4 流式音视频

- MPEG标准

图 7-49　3 个连续帧

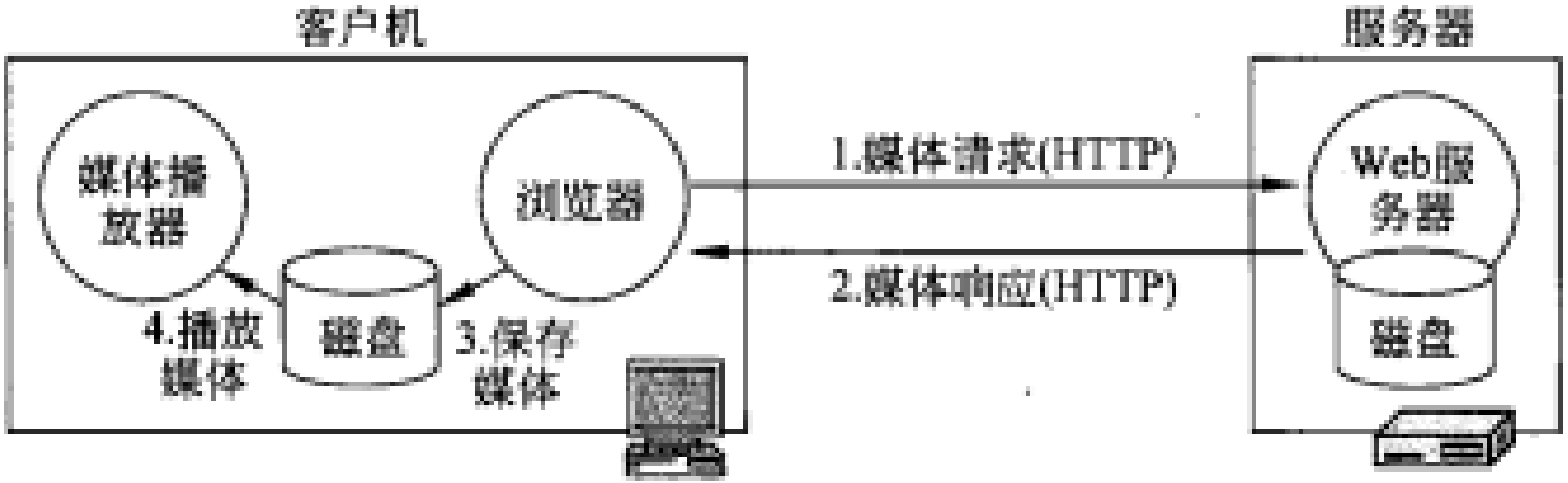


图 7-50　通过简单的下载在 Web 上播放媒体

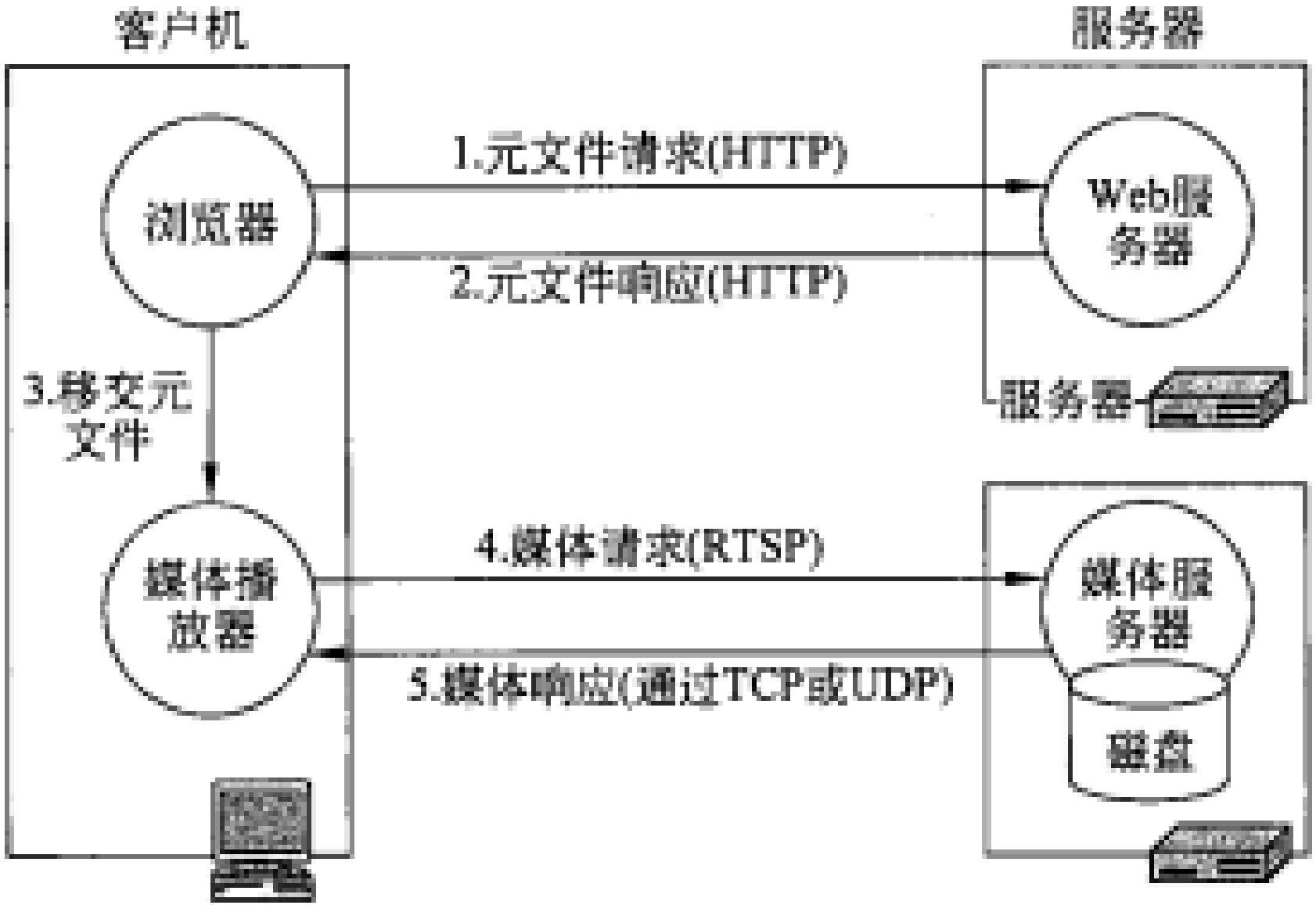
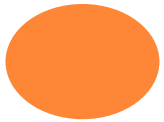


图 7-51　使用了 Web 和媒体服务器的流式媒体

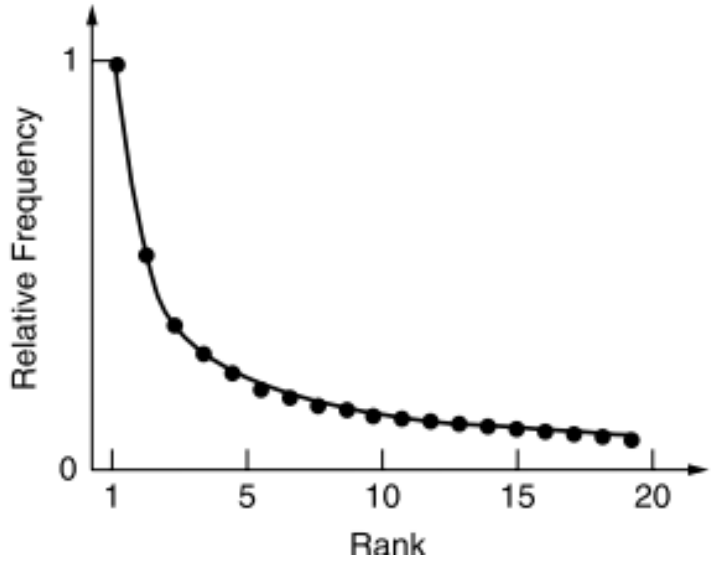
7.5 内容分发

- 计算机网络通信有不同的需求，对于访问WEB站点，视频和语音通信等，需要特定的目的地址或用户。而对一些内容的获取，用户更关心内容本身，对于通信对方并不关注。
- 两种不同的内容分发体系结构
- 内容分发网络（Content Distribution network CDN）
- 对等网络(Peer-to-Peer,P2P)



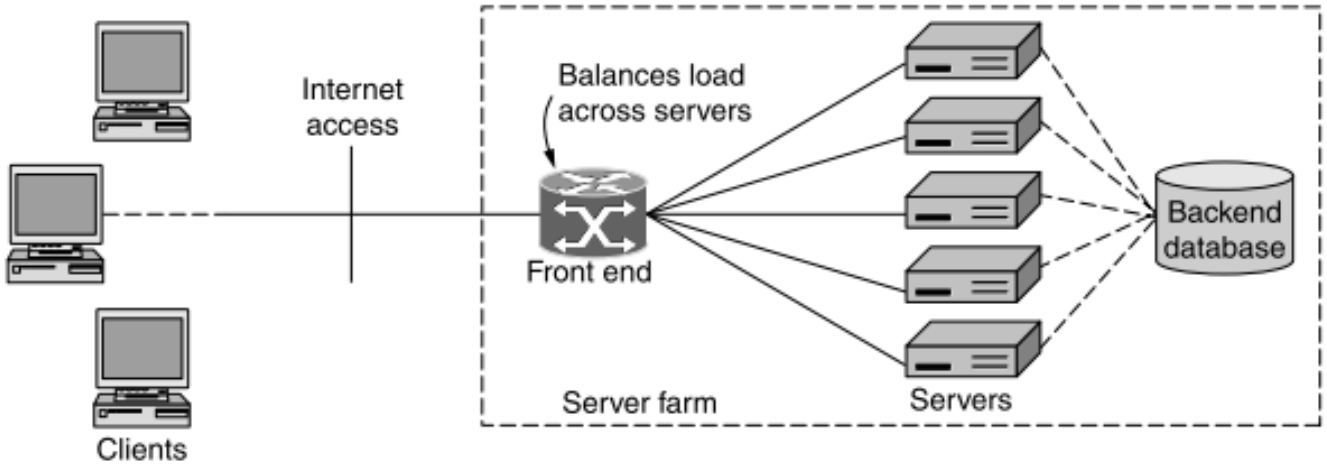
7.5.1 内容和INTERNET流量

- Internet流量的变化 1994FTP-2000WEB-2003P2P-2014视频
- 网页访问的幂定律



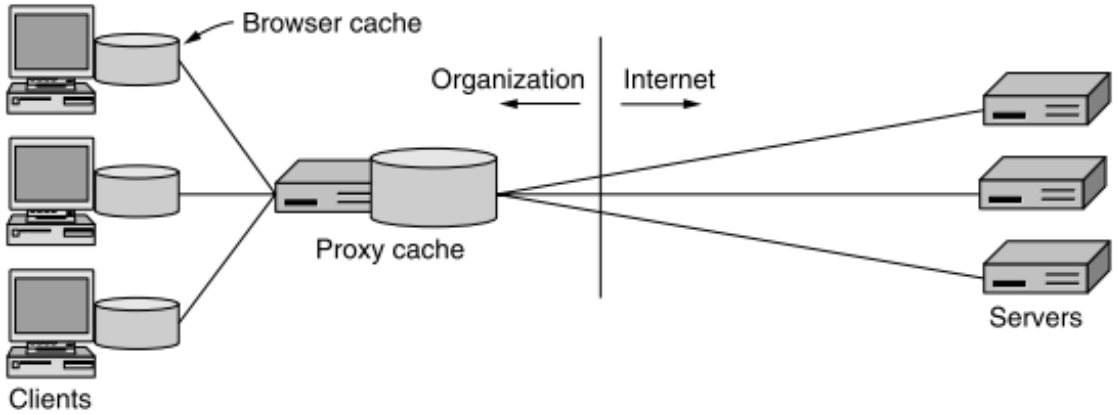
7.5.2 服务器农场与WEB代理

- 为了提高服务端性能，使用多台计算机充当WEB服务器，构建服务器农场模型，使得客户端看起来像是单个逻辑网站。
- 方法1：使用DNS将请求分散到不同服务器上，DNS返回IP的循环列表，客户端选取其中之一。
- 方法2：前端把请求分散到服务器池，采用广播或负载均衡的方式，选择服务器。



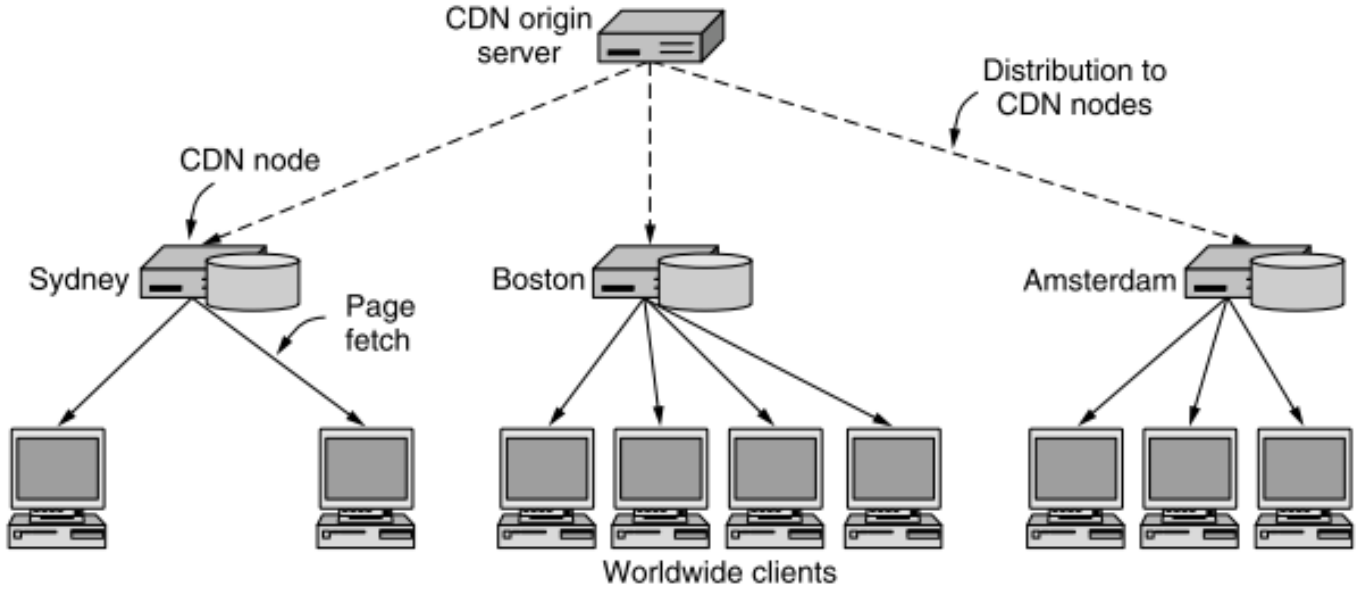
7.5.2 服务器农场与WEB代理

- WEB代理（代理缓存）会接收浏览器发出 HTTP 请求，然后代替原主机将请求发送给站点。。
- WEB代理把最近的一些请求和响应暂存在本地磁盘（缓存）中。
- 当与暂时存放的请求相同的新请求到达时， WEB代理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。



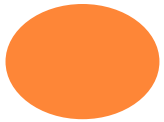
7.5.3 内容分发网络

- CDN为客户提供信息，在一组分布在不同地点的节点中哪个节点拥有所请求页面的副本，然后指示客户端使用附近的节点作为服务器。



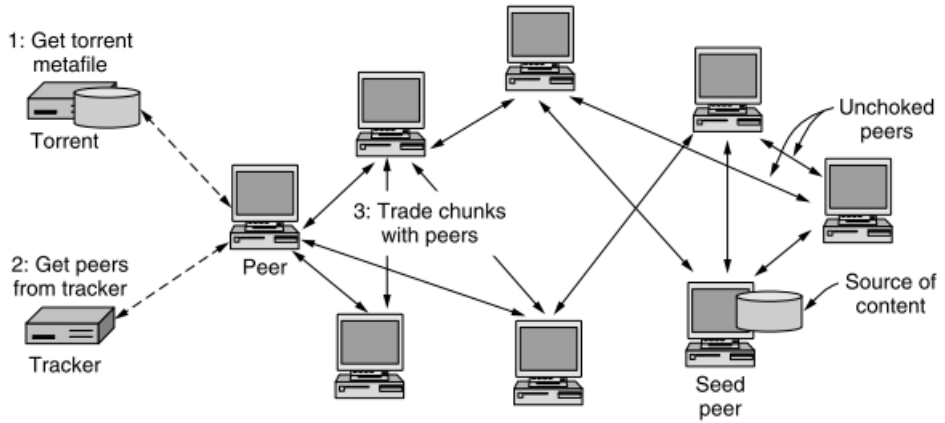
7.5.3 内容分发网络

- 1、分发树：数据按照树形结构分发。
- 2、镜像：CDN为提供镜像的技术。源服务器将内容复制给CDN节点，不同网络区域的CDN节点称为镜像点。
- 3、DNS重定向：DNS服务器作为CDN节点运行，对于相同网站的请求，返回不同的IP地址，将最靠近客户端的CDN内容站节点的IP返回给客户



7.5.4 对等网络

- P2P基本思路就是将多台计算机连接起来，共享资源。每个P2P网络中的节点，既是客户端可下载文件，也是服务器，提供内容给其他对等节点。
- BitTorrent
- 利用种子文件描述内容，获取需要的内容信息，并联系跟踪器节点，加入用户集群，下载和上传。



第七章作业:

1，2 ，3，4，6