

Chapter 7

The Application Layer (应用层)

Contents

- Network Application Model
- DNS
- Electronic Mail
- World Wide Web
- FTP [*]
- Multimedia [*]
- Content Delivery [*]

0

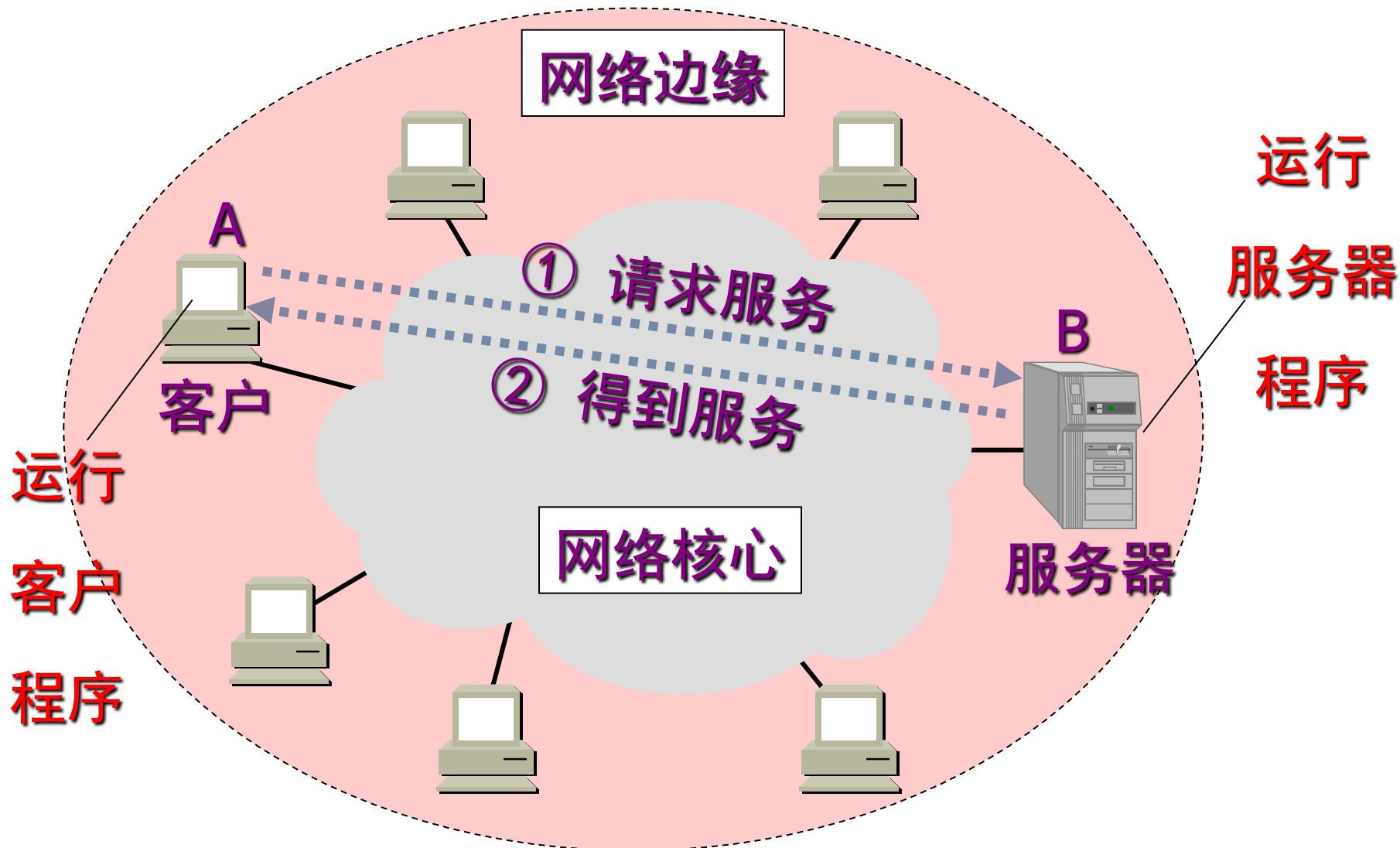
网络应用模型

网络应用模型

- 客户服务器模型(**C/S**)
 - 即**Client/Server**模型
- 对等模型(**P2P**)
 - 即**Peer-to-Peer**模型

客户服务器方式

- 客户(client)和服务器(server)都是指通信中所涉及的两个应用进程。
- 客户服务器方式所描述的是进程之间服务和被服务的关系。
- 客户是服务的请求方， 服务器是服务的提供方。



客户 A 向服务器 B 发出请求服务，
而服务器 B 向客户 A 提供服务。

客户端的特点

- 被用户调用后运行，在打算通信时主动向远地服务器发起通信（请求服务）。因此，客户程序必须知道服务器程序的地址。
- 不需要特殊的硬件和很复杂的操作系统。

服务端的特点

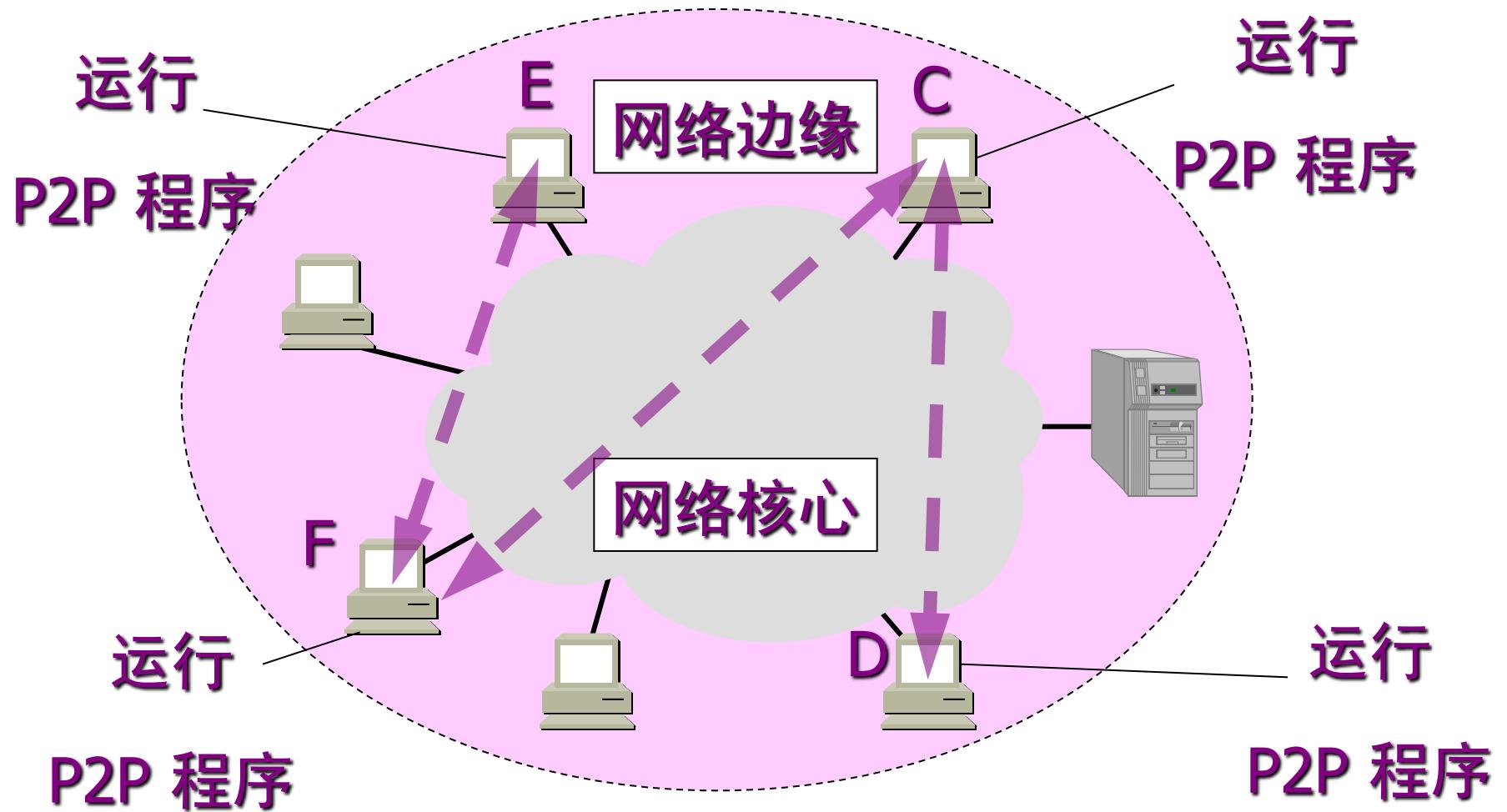
- 一种专门用来提供某种服务的程序，可同时处理多个远地或本地客户的请求。
- 系统启动后即自动调用并一直不断地运行着，被动地等待并接受来自各地的客户的通信请求。因此，服务器程序不需要知道客户程序的地址。
- 一般需要强大的硬件和高级的操作系统支持。

对等模型

- 对等连接(peer-to-peer，简写为 P2P)是指两个主机在通信时并不区分哪一个是服务请求方还是服务提供方。
- 只要两个主机都运行了对等连接软件(P2P 软件)，它们就可以进行平等的、对等连接通信。
- 双方都可以下载对方已经存储在硬盘中的共享文档。

对等连接方式的特点

- 对等连接方式从本质上讲仍然是使用客户服务器方式，只是对等连接中的每一个主机既是客户又同时是服务器。
- 例如主机 C 请求 D 的服务时，C 是客户，D 是服务器。但如果 C 又同时向 F 提供服务，那么 C 又同时起着服务器的作用。



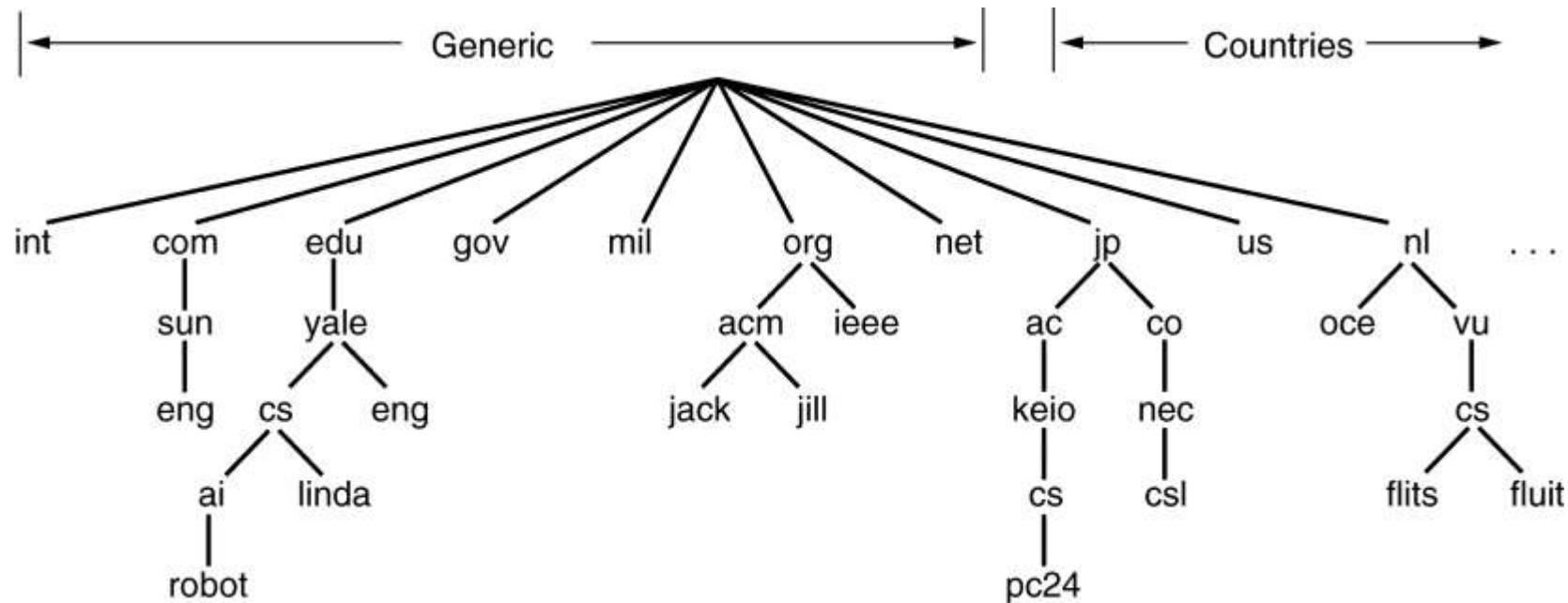
2

域名系统DNS

7.1 DNS – The Domain Name System

- The DNS Name Space**
- Resource Records**
- Name Servers**

7.1.1 The DNS Name Space



A portion of the Internet domain name space.

The DNS Name Space(2)

- 顶层域分为两大类：一般的和国家的
 - 一般的： com, edu, gov, int, mil, net, org ...
 - 国家的： cn, ca, uk, jp, fr, ...
- 每个域由它往上到根的路径命名，成员由句点“.”分隔。
cs. zju. edu. cn
- 域名对大小写不敏感(**case insensitive**)：
 - edu和EDU相同
- 成员名最长63个字符，路径名不能超过255个字符
- Domain names can be either **absolute or relative**

The DNS Name Space(3)

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Generic top-level domains

7.1.2 Resource Records

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

The principal DNS resource records types.

- DNS的实际功能就是把域名映射到资源记录上
- 资源记录的格式：
 - Domain_name, Time_to_live, Type, Class, Value
 - Type : 记录类型
 - 最重要的是A(地址)

Resource Records (2)

```
; 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 IISi" Proprietary
```

A portion of a possible DNS database for *cs.vu.nl.*

Example of Resource records

mail.xyz.com.cn	86400	IN	A	202.10.108.44
unix1.xyz.com.cn	86400	IN	A	202.10.108.45
smtp.xyz.com.cn	86400	IN	A	202.10.108.46
hr.xyz.com.cn	86400	IN	A	202.10.108.47
xyz.com.cn	86400	IN	MX	1 mail.xyz.com.cn
xyz.com.cn	86400	IN	MX	2 smtp.xyz.com.cn
hr.xyz.com.cn	86400	IN	MX	sun.xyz.com.cn
sun.xyz.com.cn	86400	IN	CNAME	unix1.xyz.com.cn

Q: What is the mail server address for zhang3@hr.xyz.com.cn?

Mydns - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

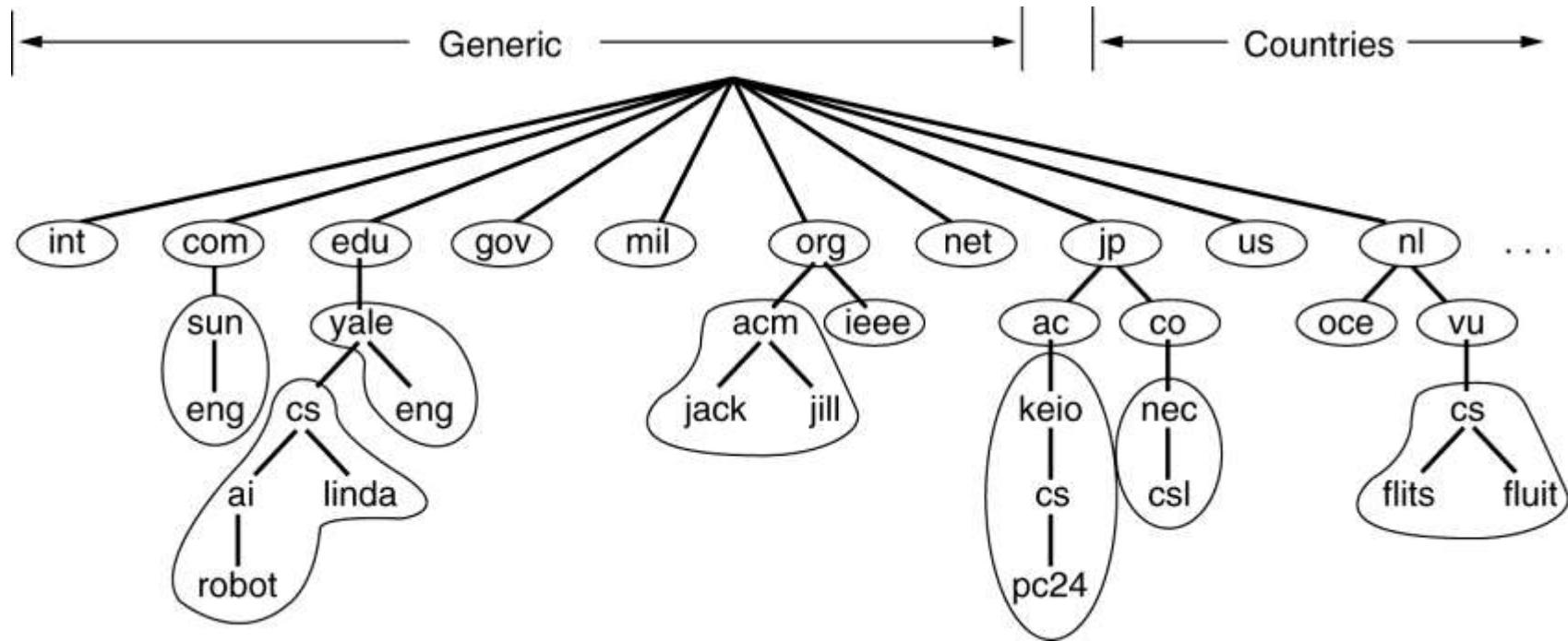
后退 前进 停止 搜索 收藏夹 媒体 邮件 图文 网址 地址栏: http://mydns.xinnet.com/cgi-bin/mydnsinfo.cgi?DomainName=zjunbsn.com&mystring=2d7a8e545d60178e2ef64efbe4a5a361&mystring2=1103096 链接

1、MYDNS功能，是DNS功能服务的扩展，通过这一深层服务，您可以独立完成对域名服务器各项记录的设置。在您没有仔细阅读以下说明、或对域名解析记录（如：A记录，MX记录，CNAME记录等）没有详细了解的情况下，为了确保您的网站和邮件正常使用，请您不要随意修改MYDNS中的各项内容，否则引起的一切后果我们不承担任何责任。
备注：含有[*.xinnet.com]、以及[210.51.168.*][210.51.170.*][210.51.172.*]的记录，不允许用户添加、修改。如因误删除而需要恢复，请参考[相关说明](#)。

2、如果您确实需要修改DNS记录，但又对DNS记录不够熟悉，这个工作可以交由我们来完成。[办理方法](#)

DNS解析记录		暂停解析		DNS解析		URL转发	
域名 (NS)(最多允许20条)	域名服务器	TTL	操作	帮助			
zjunbsn.com	ns.xinnetdns.com	3600	不可修改				
zjunbsn.com	ns.xinnet.cn	3600	不可修改				
添加新的Nameserver记录							
<input type="button" value="提交"/> 注：只提交新加纪录							
主机名 (A)(最多允许20条)	IP 地址	TTL	操作	帮助			
www.zjunbsn.com	221.12.59.221	3600	修改 - 删除				
zjunbsn.com	221.12.59.221	3600	修改 - 删除				
添加新的A记录							
<input type="button" value="提交"/> 注：只提交新加纪录							
别名 (CNAME)(最多允许20条)	别名主机	TTL	操作	帮助			
ftp.zjunbsn.com	www.zjunbsn.com	3600	修改 - 删除				
mail.zjunbsn.com	221.12.59.222	3600	修改 - 删除				
mysql.zjunbsn.com	mysql-g1.xinnetdns.com	3600	修改 - 删除				
science.zjunbsn.com	221.12.59.221	500	修改 - 删除				
添加新的别名							
<input type="button" value="提交"/> 注：只提交新加纪录							
邮件交换记录 (MX)(最多允许20条)	目标主机	优先级	TTL	操作	帮助		
zjunbsn.com	221.12.59.222	10	3600	修改 - 删除			
添加新的邮件记录							
<input type="button" value="提交"/> 注：只提交新加纪录							

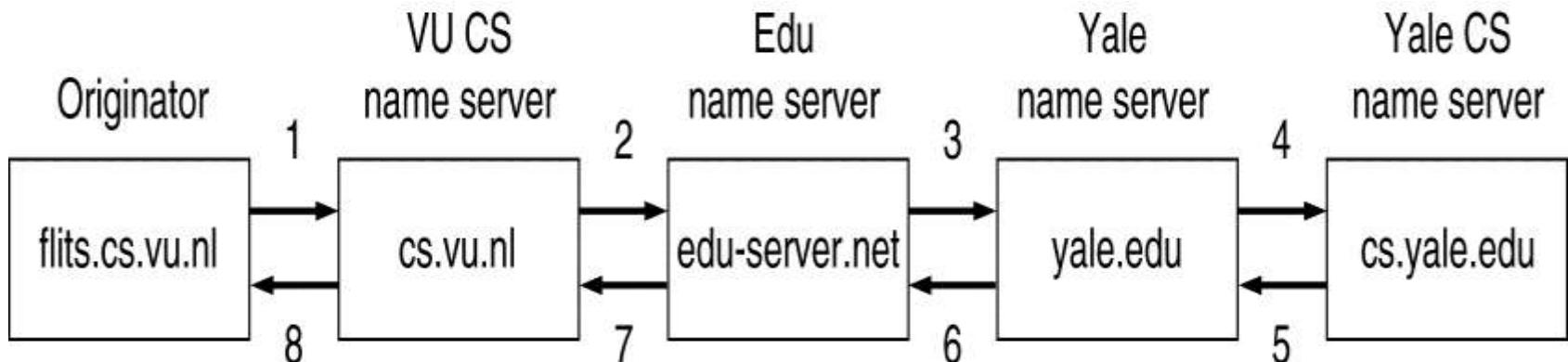
7.1.3 Name Servers



Part of the DNS name space showing the division into zones.
DNS名字空间被划分为一些不交叉的区域(zone)

Name Servers (2)

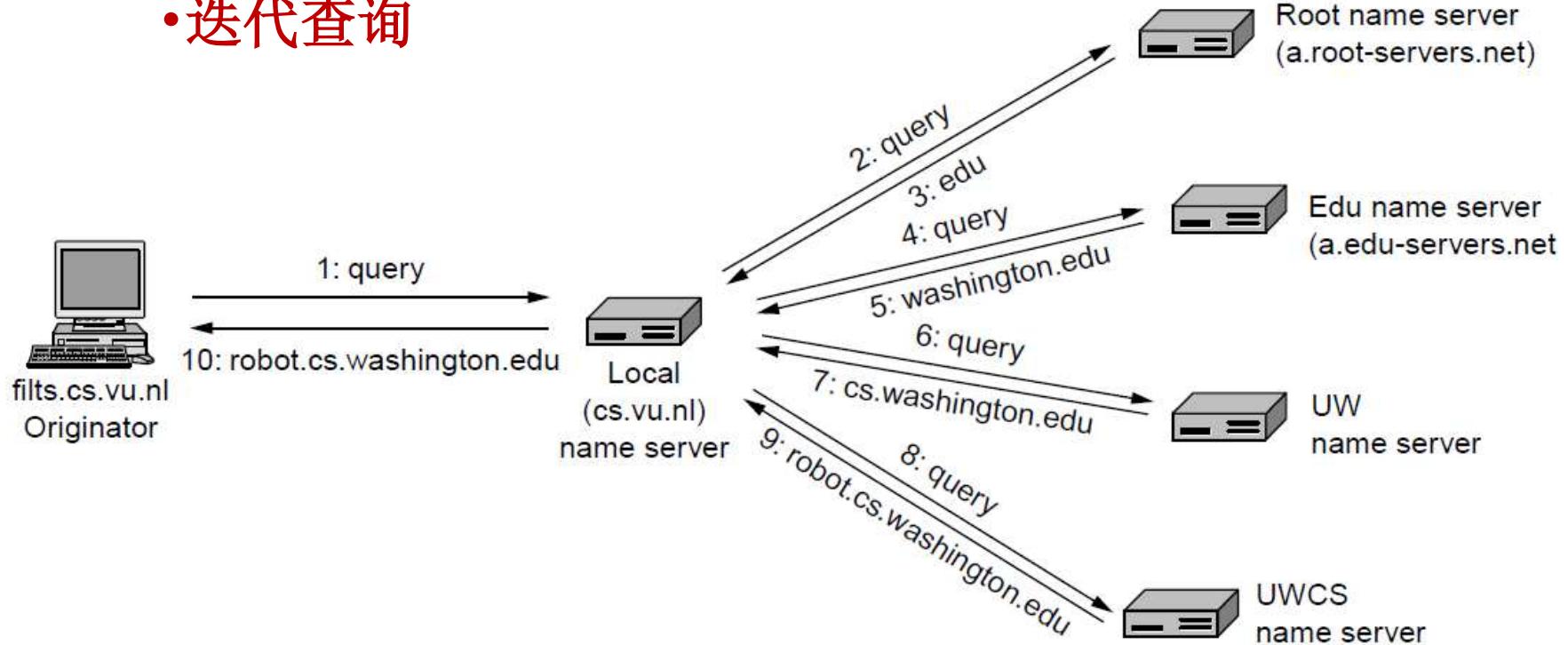
- Authoritative record(权威记录): 记录管理机构获得的、总是正确的记录
 - 缓存记录
- 远程名字的查询过程
 - 递归查询(比较少用)



How a resolver looks up a remote name in eight steps.

Name Servers (3)

- 迭代查询



Example of a resolver looking up a remote name in 10 steps.

7.2 Electronic Mail

- Architecture and Services
- The User Agent
- Message Formats
- Message Transfer
- Final Delivery

Electronic Mail(2)

- 主要协议
 - ARPANET: RFC 821(传输协议) 和 RFC 822(消息格式, new RFC5322)
 - CCITT X.400 ->OSI: MOTIS 现 X.400 已基本消失

Electronic Mail (3)

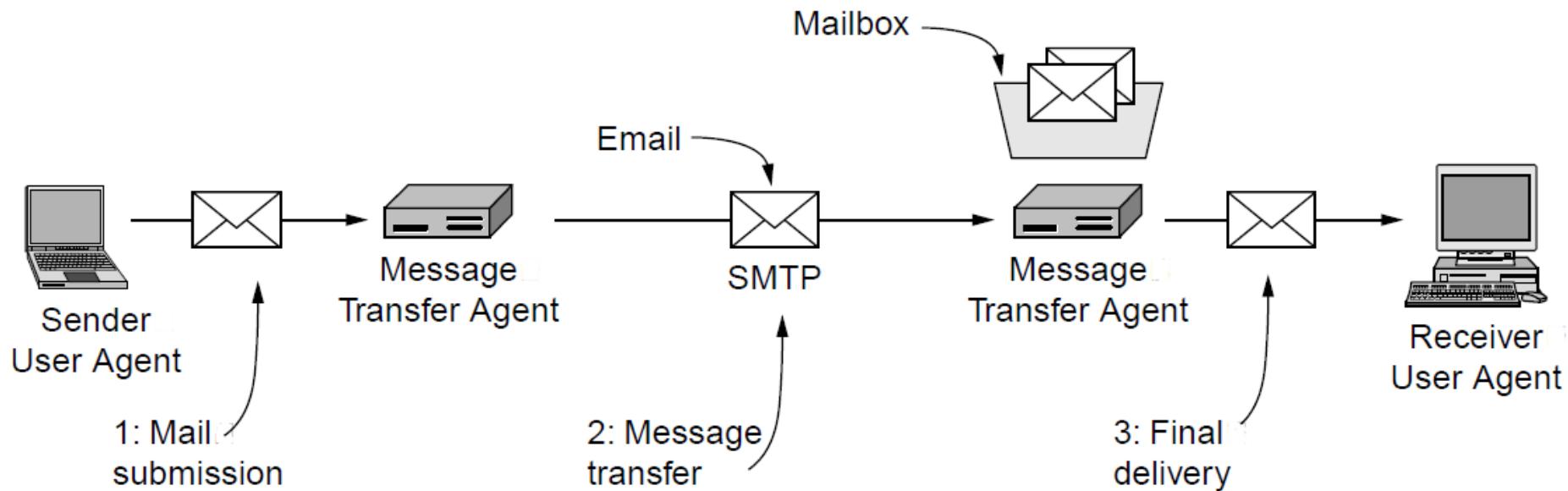
Smiley	Meaning	Smiley	Meaning	Smiley	Meaning
: -)	I'm happy	= : -)	Abe Lincoln	: +)	Big nose
: - (I'm sad/angry	=) : -)	Uncle Sam	: -))	Double chin
: -	I'm apathetic	* < : -)	Santa Claus	: - {)	Mustache
; -)	I'm winking	< : - (Dunce	# : -)	Matted hair
: - (O)	I'm yelling	(- :	Australian	8 -)	Wears glasses
: - (*)	I'm vomiting	:) X	Man with bowtie	C : -)	Large brain

Some smileys微笑图标 (emoticons, emotion icons情
绪图标). They will not be on the final exam :-).

7.2.1 Architecture and Services

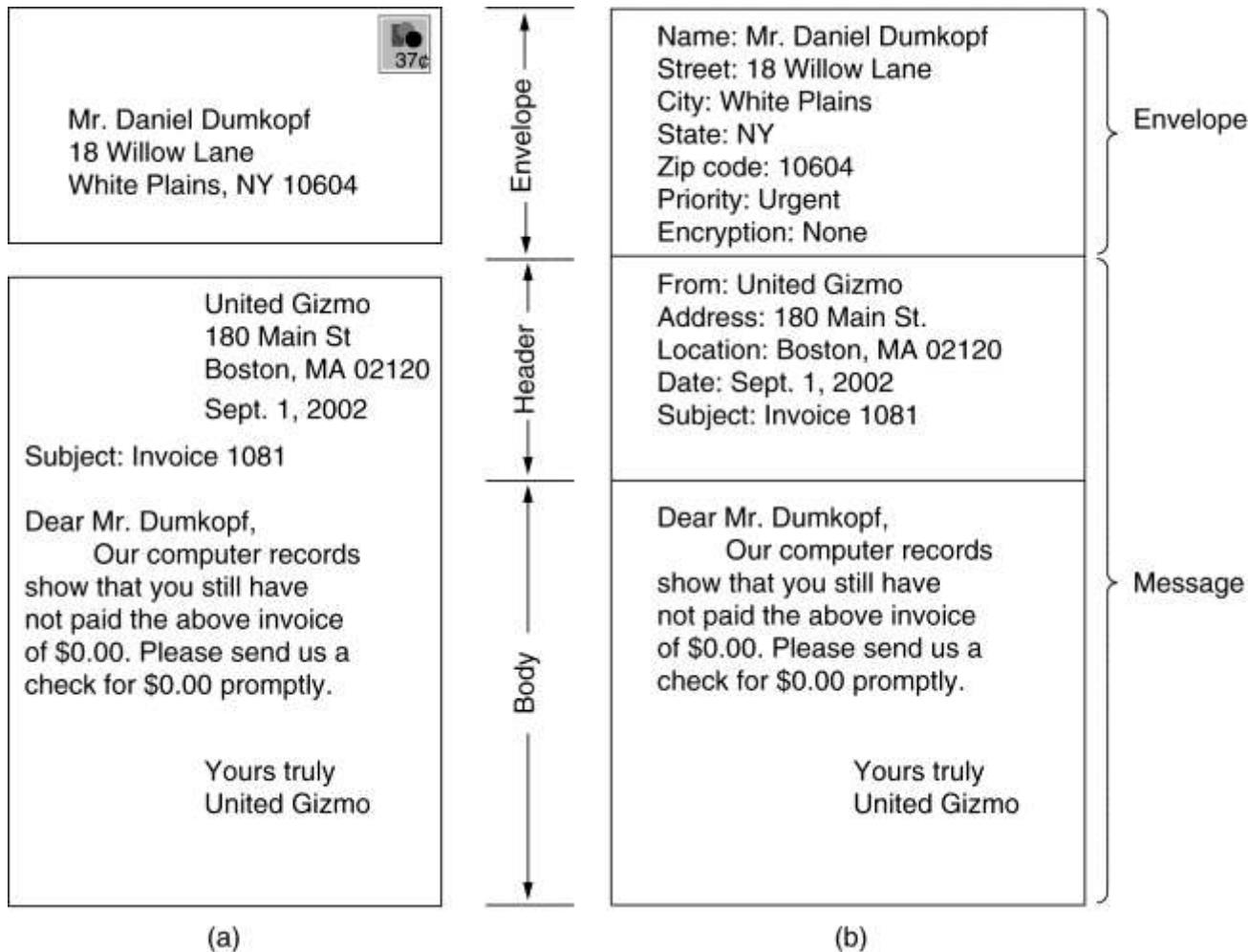
- 系统组成 --2个子系统
 - 用户代理(User Agent)
 - To allow people to read and send e-mail (sometimes called an **email reader**)
 - 消息传输代理(Message Transfer Agent, MTA)
 - To move messages from the source to the destination
- Basic functions
 - Composition(撰写)
 - Transfer(传输)
 - Reporting(报告)
 - Displaying(显示)
 - Disposition(处理)
 - 其它:转发、邮箱、收件人列表、挂号电子邮件、优先级、加密

Architecture and Services (2)



Architecture of the email system

Architecture and Services (3)



Envelopes and messages. (a) Paper mail. (b) Electronic mail.

7.2.2 The User Agent

Message folders

Mail Folders
All items
Inbox
Networks
Travel
Junk Mail

Message summary

From	Subject	Received
trudy	✉ Not all Trudys are nasty	Today
Andy	📎 Material on RFID privacy	Today
djw	❗ Have you seen this?	Mar 4
Amy N. Wong	Request for information	Mar 3
guido	Re: Paper acceptance	Mar 3
lazowska	More on that	Mar 2
lazowska	📎 New report out	Mar 2
...

Search 

Mailbox search

A. Student	Graduate studies?	Mar 1
Dear Professor,	I recently completed my undergraduate studies with distinction at an excellent university. I will be visiting your	...

Message

Typical elements of the user agent interface

Reading E-mail

#	Flags	Bytes	Sender	Subject
1	K	1030	asw	Changes to MINIX
2	KA	6348	trudy	Not all Trudys are nasty
3	K F	4519	Amy N. Wong	Request for information
4		1236	bal	Bioinformatics
5		104110	kaashoek	Material on peer-to-peer
6		1223	Frank	Re: Will you review a grant proposal
7		3110	guido	Our paper has been accepted
8		1204	dmr	Re: My student's visit

An example display of the contents of a mailbox.

- K – message is not new (be read), kept in the mailbox
- A – message has already been answered
- F – message has been forwarded to someone else

Mail Operations

- T – type message
- A – Answer message
- D – Delete message
- F – Forward message

7.2.3 Message Formats – RFC 5322

- 基本ASCII格式 和 多媒体扩展格式
- RFC 5322 (old RFC 822)
 - 消息：基本信封+头字段+空行+主体
 - 与消息传输有关的RFC5322头字段

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

RFC 5322 header fields related to message transport.

Message Formats – RFC 5322 (2)

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

Some fields used in the RFC 5322 message header.

Note the differences between RFC 822 and RFC 5322!

MIME – Multipurpose Internet Mail Extensions

Problems with international languages:

- Languages with accents
(French, German).
- Languages in non-Latin alphabets
(Hebrew, Russian).
- Languages without alphabets
(Chinese, Japanese).
- Messages not containing text at all
(audio or images).

MIME (2)

➤ MIME – 多用因特网邮件扩展

- 基本思想：继续使用RFC 822的形式，但对消息主体增加结构，并为非ASCII消息定义编码结构
- 5种新的消息头部：

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

Message headers added by MIME.

MIME (3)

➤MIME – 多用因特网邮件扩展

- 二进制消息编码方法:64个基本字符编码
(24bit→4个6bit字符)
- MIME类型和子类型
 - 标准通用标记语言SGML
 - 移动图像专家组MPEG

MIME (4)

Type	Example subtypes	Description
text	plain, html, xml, css	Text in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quicktime	Movies
model	vrml	3D model
application	octet-stream, pdf, javascript, zip	Data produced by applications
message	http, rfc822	Encapsulated message
multipart	mixed, alternative, parallel, digest	Combination of multiple types

The MIME types and subtypes defined in RFC 3023.

MIME (5)

From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/enriched

Happy birthday to you
Happy birthday to you
Happy birthday dear Carolyn
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
access-type="anon-ftp";
site="bicycle.abcd.com";
directory="pub";
name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--

A multipart message containing enriched and audio alternatives.

7.2.4 Message Transfer

- SMTP – 简单邮件传输协议
 - 一个简单的ASCII协议
 - 端口号： 25
 - TCP连接
- 发送 [fig 7-14] 消息的简单对话 – [see fig 7-15]

Message Transfer (2)

A multipart message containing HTML and audio alternatives.

From: alice@cs.washington.edu
To: bob@ee.uwa.edu.au
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@cs.washington.edu>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm

Content-Type: text/html

<p>Happy birthday to you

Happy birthday to you

Happy birthday dear Bob

Happy birthdav to vou

--qwertyuiopasdfghjklzxcvbnm

Content-Type: message/external-body;
access-type="anon-ftp";
site="bicycle.cs.washington.edu";
directory="pub";
name="birthday.snd"

content-type: audio/basic

content-transfer-encoding: base64

--qwertyuiopasdfghjklzxcvbnm--

Message Transfer (3)

```
S: 220 ee.uwa.edu.au SMTP service ready
C: HELO abcd.com
      S: 250 cs.washington.edu says hello to ee.uwa.edu.au
C: MAIL FROM: <alice@cs.washington.edu>
      S: 250 sender ok
C: RCPT TO: <bob@ee.uwa.edu.au>
      S: 250 recipient ok
C: DATA
      S: 354 Send mail; end with "." on a line by itself
C: From: alice@cs.washington.edu
C: To: bob@ee.uwa.edu.au
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@ee.uwa.edu.au>
C: Content-Type: multipart/alternative; boundary=qwertuuiopasdfghjklzxcvbnm
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: --qwertuuiopasdfghjklzxcvbnm
C: Content-Type: text/html
C:
    . . .
```

Sending a message from *alice@cs.washington.edu* to
bob@ee.uwa.edu.au.

Message Transfer (4)

```
....  
C: <p>Happy birthday to you  
C: Happy birthday to you  
C: Happy birthday dear <b> Bob </b>  
C: Happy birthday to you  
C:  
C: --qwertyuiopasdfghjklzxcvbnm  
C: Content-Type: message/external-body;  
C: access-type="anon-ftp";  
C: site="bicycle.cs.washington.edu";  
C: directory="pub";  
C: name="birthday.snd"  
C:  
C: content-type: audio/basic  
C: content-transfer-encoding: base64  
C: --qwertyuiopasdfghjklzxcvbnm  
C: .  
S: 250 message accepted  
C: QUIT  
S: 221 ee.uwa.edu.au closing connection
```

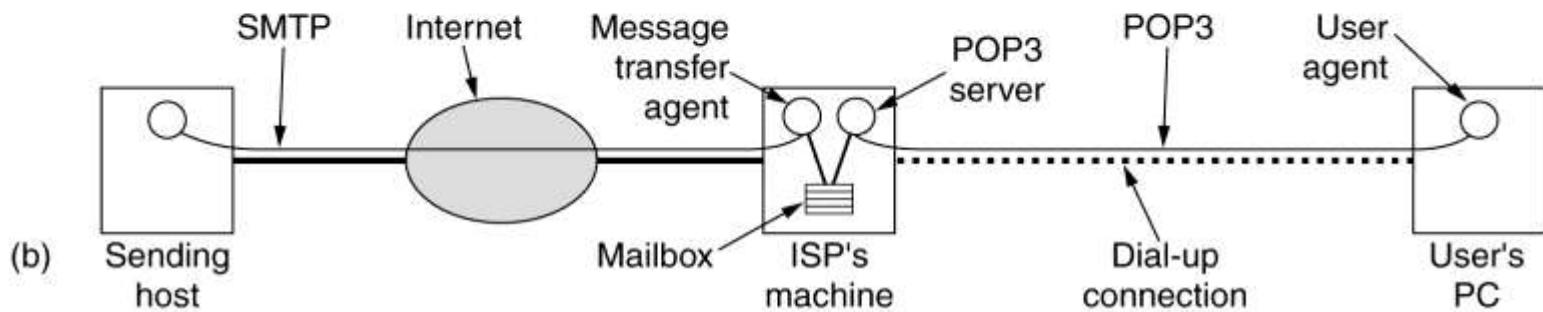
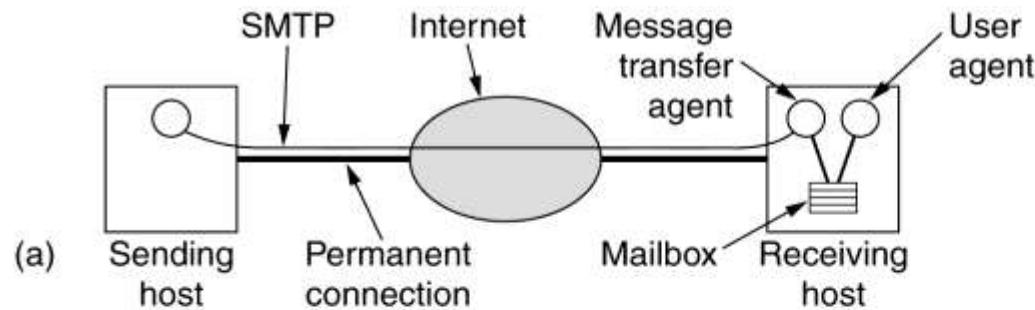
Sending a message from *alice@cs.washington.edu* to
bob@ee.uwa.edu.au.

Message Transfer (5)

Keyword	Description
AUTH	Client authentication
BINARYMIME	Server accepts binary messages
CHUNKING	Server accepts large messages in chunks
SIZE	Check message size before trying to send
STARTTLS	Switch to secure transport (TLS; see Chap. 8)
UTF8SMTP	Internationalized addresses

Some SMTP extensions.

7.2.5 Final Delivery



(a) Sending and reading mail when the receiver has a permanent Internet connection and the user agent runs on the same machine as the message transfer agent. **(b)** Reading e-mail when the receiver has a dial-up connection to an ISP.

Final Delivery(2)

- 最后的递交 – 解决非因特网上机器的消息递交
 - 邮局协议POP3
 - 从远程邮箱中读取电子邮件的简单协议
 - 交互式电子邮件访问协议IMAP
 - 邮件服务器维护一个中心数据库，能够被各种其它机器访问
 - 分布式电子邮件系统协议DMSP
 - 将邮件从服务器上下载(download)到机器上，断开连接后再阅读
- 其它传输特性
 - 过滤器
 - 转发
 - 假期幽灵(vacation daemon)程序
 - 邮件自动处理机(robot)，处理收到的每封邮件

POP3

```
S: +OK POP3 server ready
C: USER carolyn
    S: +OK
C: PASS vegetables
    S: +OK login successful
C: LIST
    S: 1 2505
    S: 2 14302
    S: 3 8122
    S: .
C: RETR 1
    S: (sends message 1)
C: DELE 1
C: RETR 2
    S: (sends message 2)
C: DELE 2
C: RETR 3
    S: (sends message 3)
C: DELE 3
C: QUIT
    S: +OK POP3 server disconnecting
```

Using POP3 to fetch three messages.

IMAP

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is e-mail stored?	User's PC	Server
Where is e-mail read?	Off-line	On-line
Connect time required?	Little	Much
Use of server resources?	Minimal	Extensive
Multiple mailboxes?	No	Yes
Who backs up mailboxes?	User	ISP
Good for mobile users?	No	Yes
User control over downloading?	Little	Great
Partial message downloads?	No	Yes
Are disk quotas a problem?	No	Could be in time
Simple to implement?	Yes	No
Widespread support?	Yes	Growing

A comparison of POP3 and IMAP.

Final Delivery (1)

Command	Description
CAPABILITY	List server capabilities
STARTTLS	Start secure transport (TLS; see Chap. 8)
LOGIN	Log on to server
AUTHENTICATE	Log on with other method
SELECT	Select a folder
EXAMINE	Select a read-only folder
CREATE	Create a folder
DELETE	Delete a folder
RENAME	Rename a folder
SUBSCRIBE	Add folder to active set
UNSUBSCRIBE	Remove folder from active set

...

IMAP (version 4, **RFC 3501**) commands.

Final Delivery (2)

LIST	List the available folders
LSUB	List the active folders
STATUS	Get the status of a folder
APPEND	Add a message to a folder
CHECK	Get a checkpoint of a folder
FETCH	Get messages from a folder
SEARCH	Find messages in a folder
STORE	Alter message flags
COPY	Make a copy of a message in a folder
EXPUNGE	Remove messages flagged for deletion
UID	Issue commands using unique identifiers
NOOP	Do nothing
CLOSE	Remove flagged messages and close folder
LOGOUT	Log out and close connection

IMAP (version 4, RFC 3501) commands.

Webmail

- Microsoft Hotmail
- Google Gmail
- Yahoo!Mail
-

7.3 The World Wide Web

- Architectural Overview
- Static Web Documents
- Dynamic Web Documents
- HTTP – The HyperText Transfer Protocol
- Performance Enhancements
- The Wireless Web

- 起源于1989年欧洲粒子物理研究室(CERN)
 - 1989. 3 最初计划, Tim Berners-Lee (CERN)
 - 1993. 2 Mosaic发布(Marc Andreessen,
Univ. of Illinois)
 - 1994 CERN和MIT共同建立WWW集团
<http://www.w3.org>
 - 1995 Netscape上市 (by Andreessen)
 - 1996 Internet Explorer 1. 0
- 编写语言: HTML、CGI和Java

Netscape明年2月退出历史舞台 AOL放弃开发

<http://www.sina.com.cn> 2007年12月29日 07:13 新浪科技

新浪科技讯 北京时间12月29日消息，据国外媒体报道，作为Netscape的所有者，时代华纳旗下AOL已决定不再继续开发Netscape并提供技术支持。这意味着拥有13年历史的Netscape将正式退出历史舞台。

曾经头戴“全球最早商业化的互联网浏览器”、“引发互联网热潮的功臣”、“挑战微软的勇士”等众多光环的Netscape，将于明年2月1日正式退出历史舞台，距其诞生正好13年。Netscape自微软杀入浏览器市场后便一蹶不振，而开源浏览器Firefox的出现更令其回天乏术。

Netscape主管汤姆·德雷派尔(Tom Drapeau)昨天在官方博客里写道：“尽管AOL内部团队投入了大量时间和精力，期望Netscape获得重生，但这些努力没能让我们从微软手中夺取市场份额。”

最近几年，Netscape看上去就像是正在走红的Firefox的翻版。Firefox在浏览器市场所占份额接近一成，其余几乎全被微软IE占据。目前，网民仍可以下载和使用Netscape，但从明年2月1日起，AOL不再发布任何安全补丁或更新。德雷派尔建议Netscape用户改用Firefox。

虽然结局凄凉，但Netscape的确有着辉煌的过去。1993年，在马克·安德森(Marc Andreessen)及其大学同事开发出Netscape的前身Mosaic之前，互联网上的内容基本上仅限于文字。Mosaic于1994年商业化后即化身为Netscape。

1995年，Netscape成功上市，发行价为28美元，上市首日便增长一倍，总市值达到年营收的100倍。Netscape的成功立即引起微软的重视，后者开始在Windows操作系统中捆绑免费的IE浏览器，并大获成功。虽然捆绑行为引来美国司法部对微软的反垄断诉讼，并且双方最终达成谅解，而Netscape也开始免费提供，但一切为时已晚。1999年，AOL以100亿美元收购了Netscape。

不过，Netscape却催生了一个开源项目——Mozilla。依靠全世界软件开发志愿者的齐心协力，Mozilla发布了自己的浏览器Firefox，并虎口夺食，从微软手中抢得了近10%的市场份额。

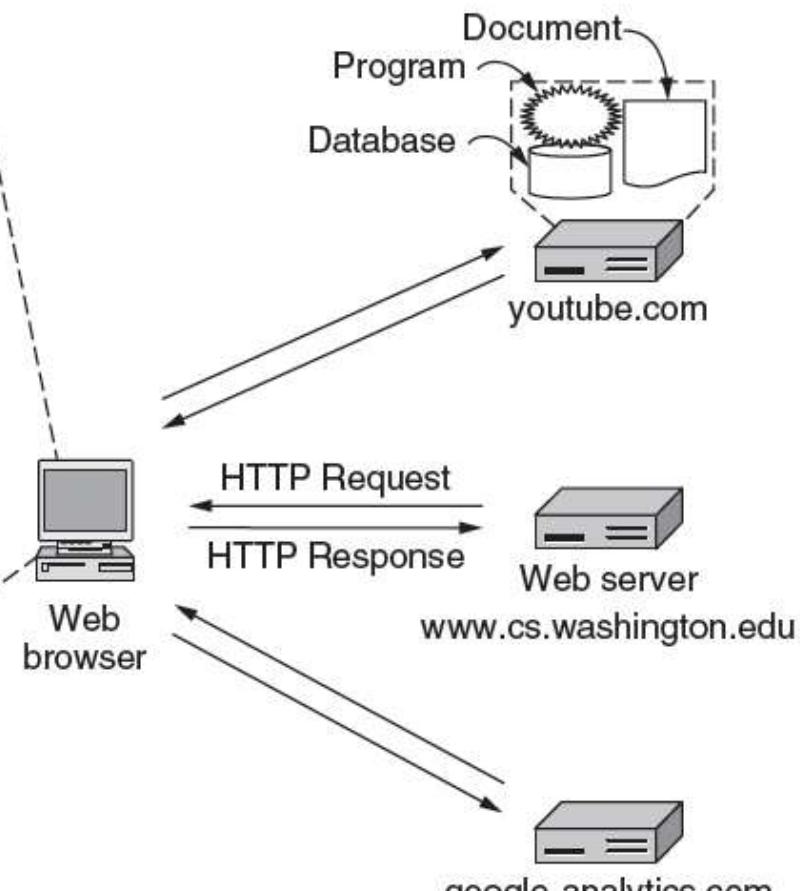
7.3.1 Architectural Overview

- Web由庞大的、世界范围的文档组成—页面（page）
- 超文本(hypertext)
- 浏览器(browser)：Mosaic、Netscape、IE
- 超链接(hyperlink)
- 书签(bookmark)
- 超媒体(hypermedia)
- TCP连接 (port number: 80)

Architectural Overview (2)



Web page



Architecture of the Web.

The Client Side

Three questions had to be answered before a selected page could be displayed:

1. What is the page called?
2. Where is the page located?
3. How can the page be accessed?

The Client Side (2)

Steps that occur when link is selected:

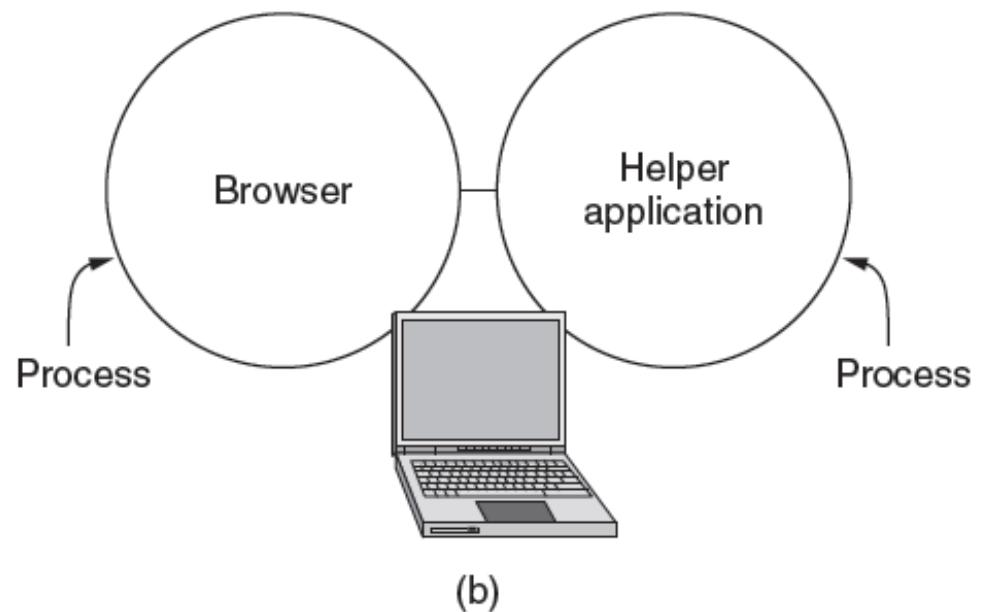
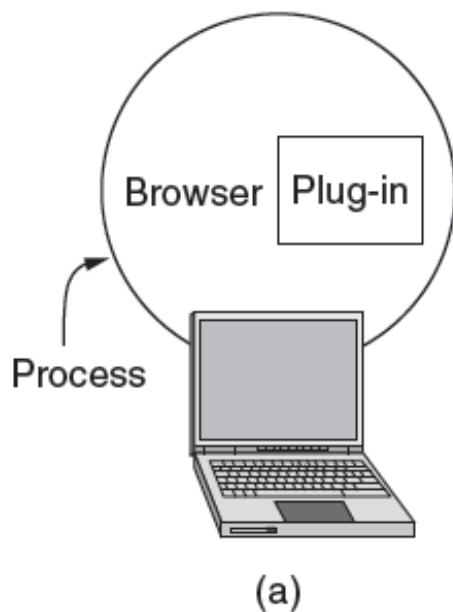
1. Browser determines the URL
2. Browser asks DNS for the IP address of the server
3. DNS replies
4. The browser makes a TCP connection
5. Sends HTTP request for the page
6. Server sends the page as HTTP response
7. Browser fetches other URLs as needed
8. The browser displays the page
9. The TCP connections are released

The Client Side (3)

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

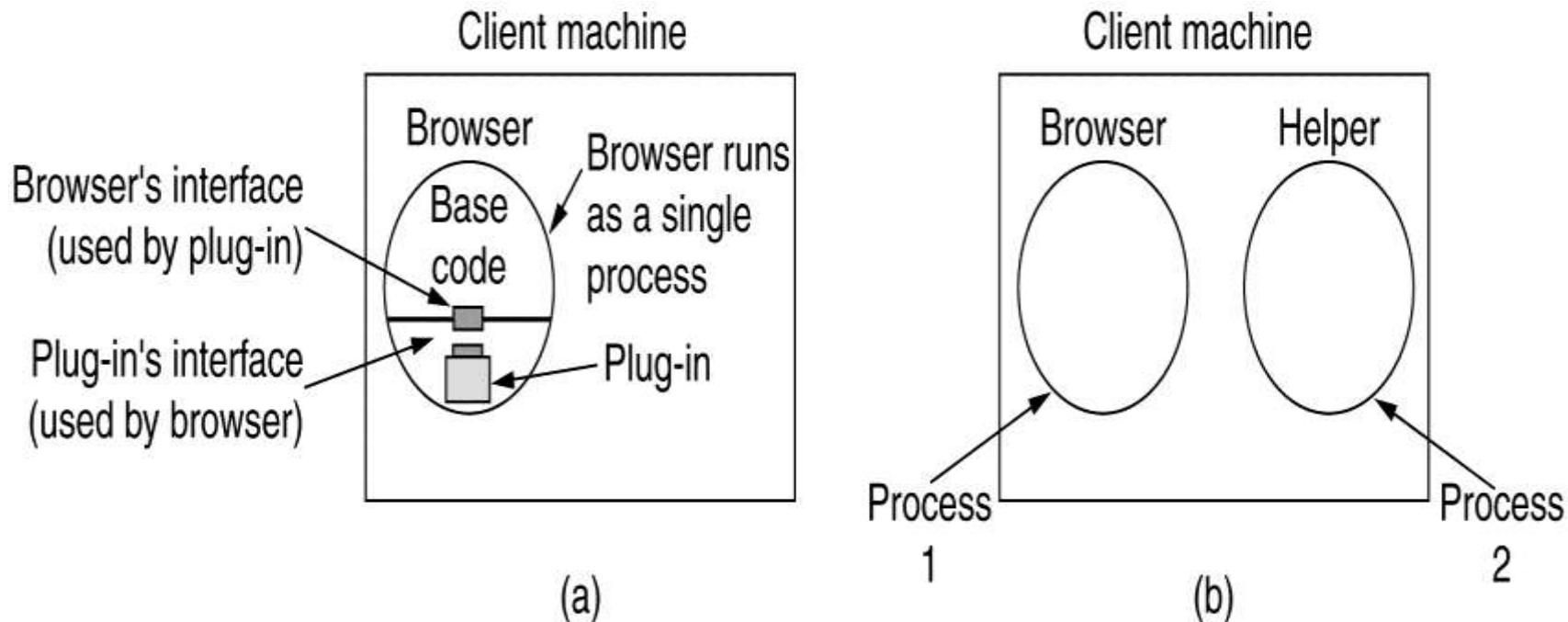
Some common URL schemes.

MIME Types



(a) A browser plug-in. (b) A helper application.

MIME Types(2)



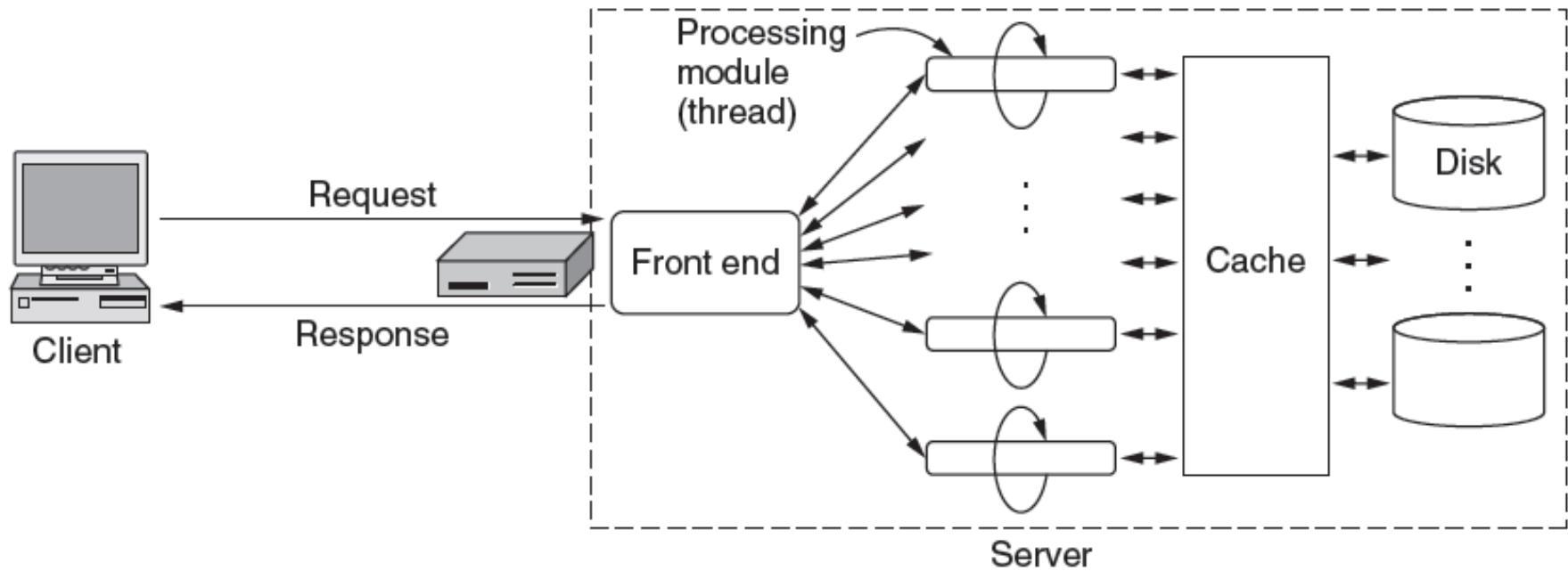
(a) A browser plug-in. (b) A helper application.
两种运行方式

The Server Side(1)

Steps server performs in main loop:

- 1.Accept a TCP connection from client
- 2.Get path to page, name of file requested.
- 3.Get the file (from disk).
- 4.Send contents of the file to the client.
- 5.Release the TCP connection.

The Server Side(2)



A multithreaded Web server with a front end and processing modules.

The Server Side(3)

A processing module performs a series of steps:

1. Resolve name of Web page requested.
2. Perform access control on the Web page.
3. Check the cache.
4. Fetch requested page from disk or run program
5. Determine the rest of the response
6. Return the response to the client.
7. Make an entry in the server log.

The Server Side(4)

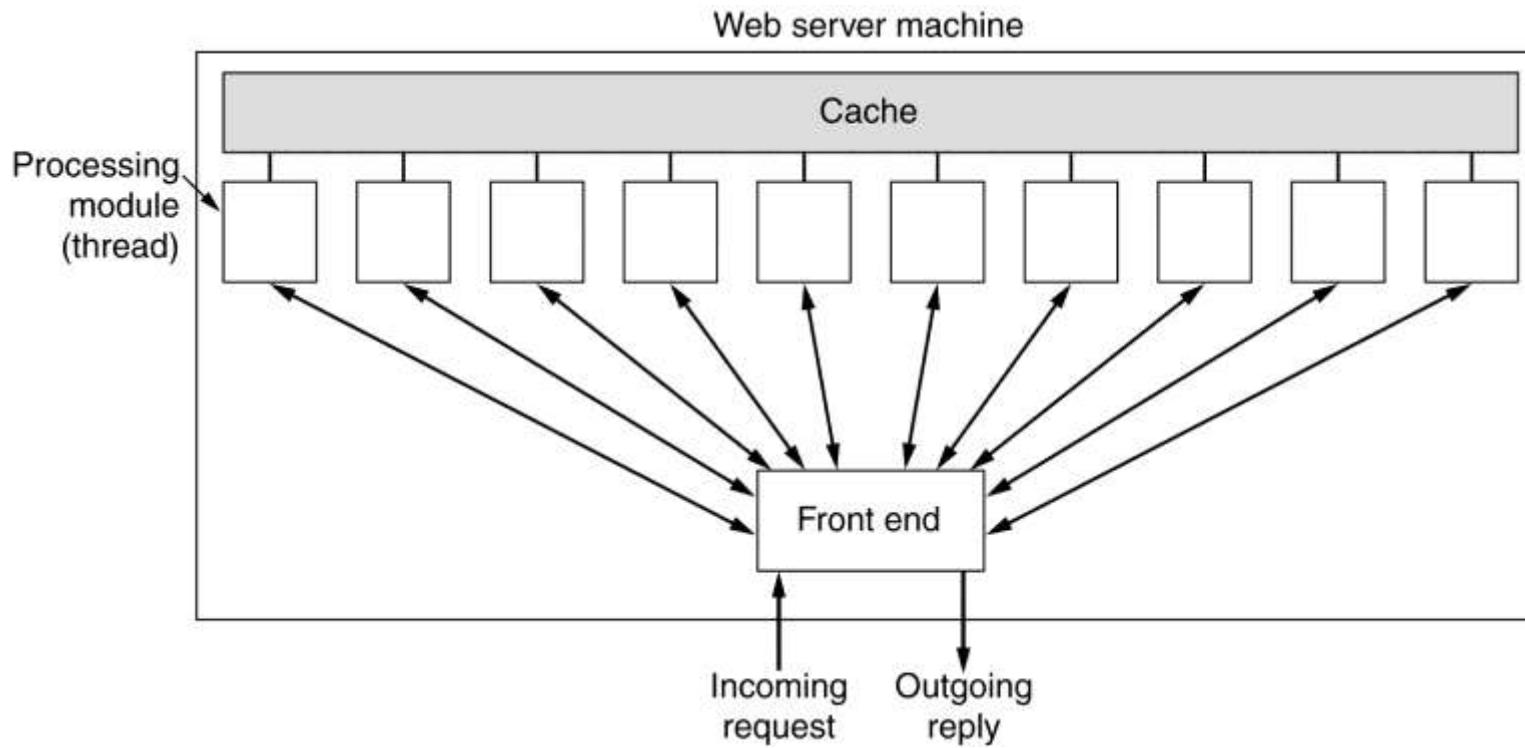
The Web server cannot serve more requests per second than it can make disk accesses.

- IDE disk
- SCSI disks
- RAID disks

How to improve the performance for a server

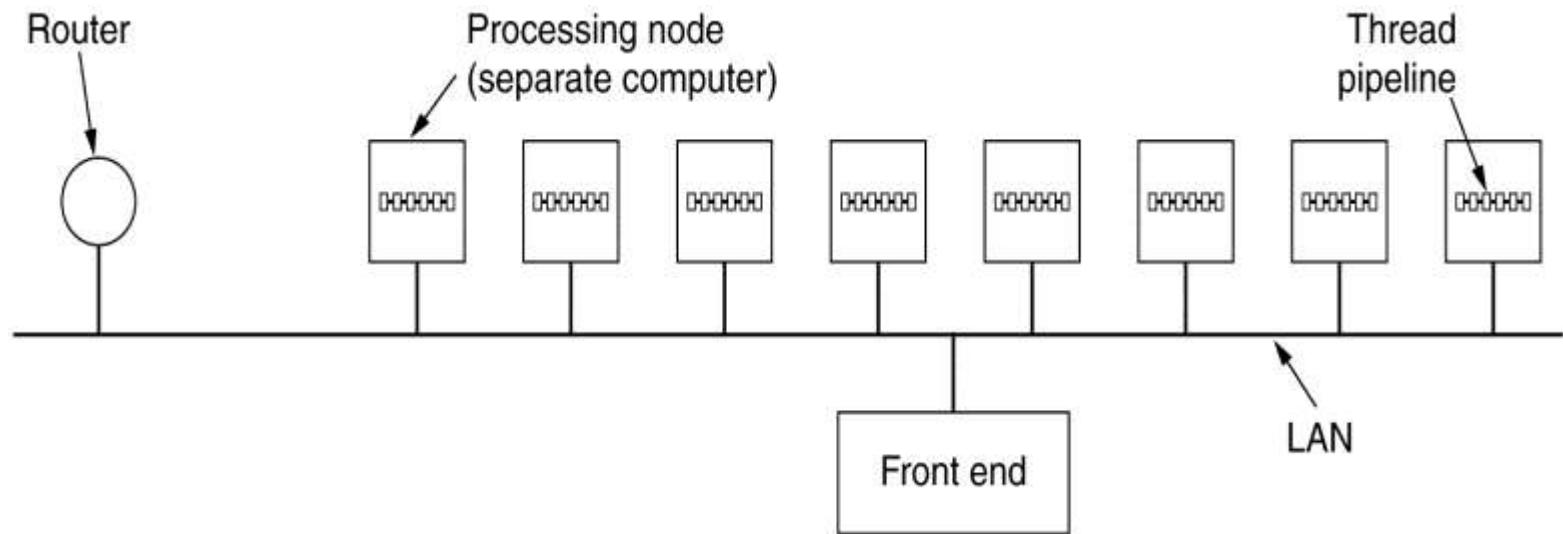
- To use caching. (multi-storages)
- To make the server multithreaded. (multi-threads)
- To use a server farm. (multi-procesors)
- To use distributed systems. (multi-systems)

The Server Side(5)



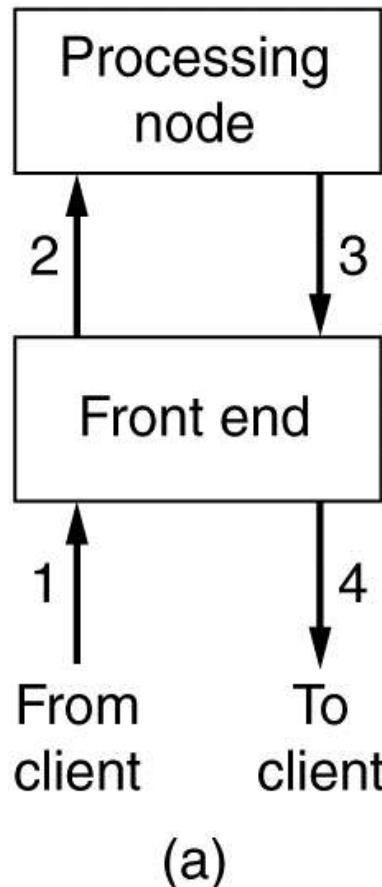
A multithreaded Web server with a front end and processing modules.

The Server Side(6)

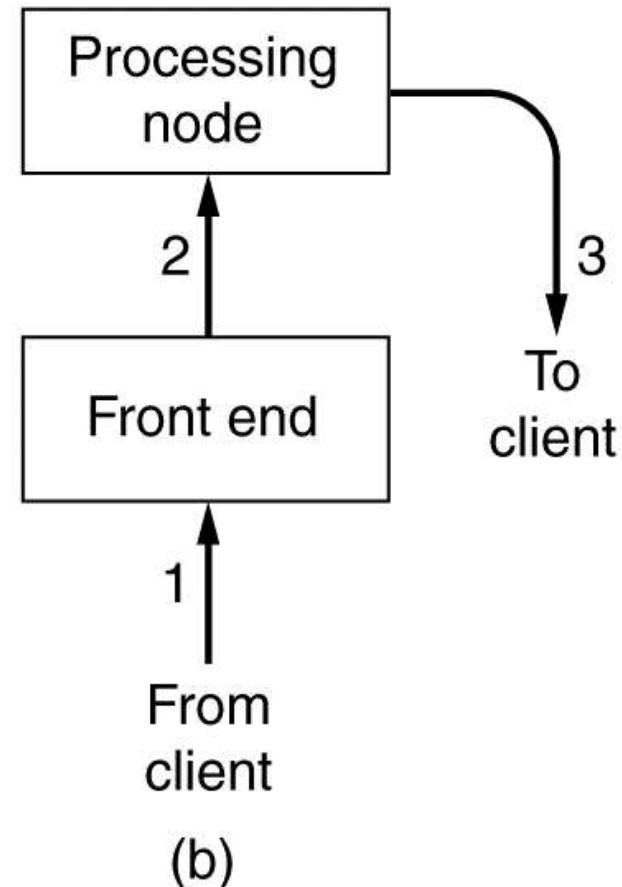


A server farm.

The Server Side(7)



(a)



(b)

(a) Normal request-reply message sequence.

(b) Sequence when TCP handoff(传递) is used.

URLs – Uniform Resource Locators

- URL consists of 3 parts:
 - the protocol (http), the DNS name of the host (www.zju.edu.cn), and the file name (index.html)
- URL中常用的协议：

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

Statelessness and Cookies

➤ Web is stateless

- No register
- How does the server keep track of the contents of the client accessing?
- To solve the problem, Netscape devised the technique: **cookies!**

➤ Cookies

- When a client requests a Web page, the server can supply additional information along with the requested page
- Cookies are just files or strings, not executable programs
- Cookies could contain virus information in data file

Statelessness and Cookies (2)

- A cookie may contain up to 5 fields:
 - Domain: where the cookie comes from
 - Path: which part of the server's file tree may use the cookie
 - Content: format as *name=value*
 - Expire: non-persistent cookie and persistent cookie
 - Secure: for e-commerce, banking or other secure applications
- Some examples of cookies:

Domain	Path	Content	Expires	Secure
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

7.3.2 Static Web Documents

● HTML – HyperText Markup Language

```
<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> 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)

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 FAX.

Product Information

- [Big widgets](http://widget.com/products/big)
- [Little widgets](http://widget.com/products/little)

(b)

Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

(a) The HTML for a sample Web page. (b) The formatted page.

HTML (2)

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h n > ... </h n >	Delimits a level n heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
	Starts a list item (there is no)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

A selection of common HTML tags. some can have additional parameters.

Forms

(a) An HTML table.

(b) A possible rendition
of this table.

```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item <th>HTML 1.0 <th>HTML 2.0 <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Forms <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Equations <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Toolbars <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Tables <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Accessibility features <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Object embedding <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Scripting <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
</table>
</body>
</html>
```

(a)

Some Differences between HTML Versions

Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0
Hyperlinks	x	x	x	x
Images	x	x	x	x
Lists	x	x	x	x
Active Maps and Images		x	x	x
Forms		x	x	x
Equations			x	x
Toolbars			x	x
Tables			x	x
Accessibility features				x
Object embedding				x
Scripting				x

(b)

Forms (2)

(a) The HTML for an order form.

(b) The formatted page.

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" 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>
```

(a)

Widget Order Form

Name

Street address

City State Country

Credit card # Expires M/C Visa

Widget size Big Little Ship by express courier

Thank you for ordering an AWI widget, the best widget money can buy!

(b)

Forms (3)

- When the user clicks the submit button, the browser packages the collected information into a single long line and sends it back to the server for processing.

```
customer=John+Doe&address=100+Main+St.&city=White+Plains&  
state=NY&country=USA&cardno=1234567890&expires=6/98&cc=mastercard&  
product=cheap&express=on
```

A possible response from the browser to the server with
information filled in by the user.

CSS

(Cascading Style Sheets, 层叠样式表)

- 采用CSS制作page，可以有效地对页面的布局、字体、颜色、背景和其它效果实现更加精确的控制。
- The original goal of HTML
 - structure
 - Style
- Some styles

```
<font face="helvetica"  
      size="24"  
      color="red"> Deborah's Photos  
</font>
```

CSS (2)

- Defining a CSS style sheet

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

- Using a CSS style sheet

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

XML and XSL

To enhance the ability for HTML to be structured for automated processing, W3C develop two new languages: XML and XSL

XML— eXtensible Markup Language (to describe Web content in a structured way)

XSL-- eXtensible Style Language (to describe the formatting of the content)

XML and XSL are still static

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>
<book_list>
  <book>
    <title> Computer Networks, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2003 </year>
  </book>
  <book>
    <title> Modern Operating Systems, 2/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2001 </year>
  </book>
  <book>
    <title> Structured Computer Organization, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 1999 </year>
  </book>
</book_list>
```

A simple Web page in XML.(book_list)

XML and XSL (2)

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

<html>
<body>

<table border="2">
  <tr>
    <th> Title</th>
    <th> Author</th>
    <th> Year </th>
  </tr>

  <xsl:for-each select="book_list/book">
    <tr>
      <td> <xsl:value-of select="title"/> </td>
      <td> <xsl:value-of select="author"/> </td>
      <td> <xsl:value-of select="year"/> </td>
    </tr>
  </xsl:for-each>
</table>

</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

A style sheet
in XSL.

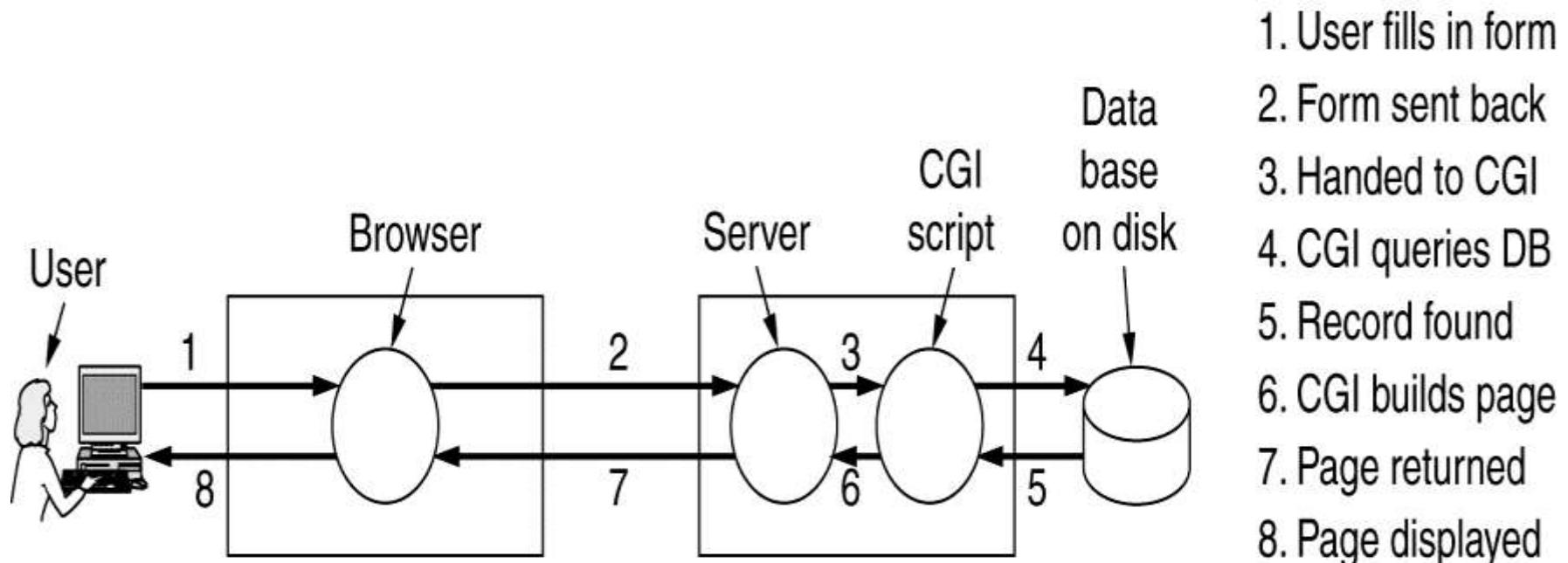
XHTML –eXtended HyperText Markup Language

- In order to fitting to the use of the Web-enabled devices, such wireless and handheld PDA-type devices
- Limited memory and small size display
- XHTML – a new Web standard
- Some differences between XHTML and HTML4

XHTML –eXtended HyperText Markup Language

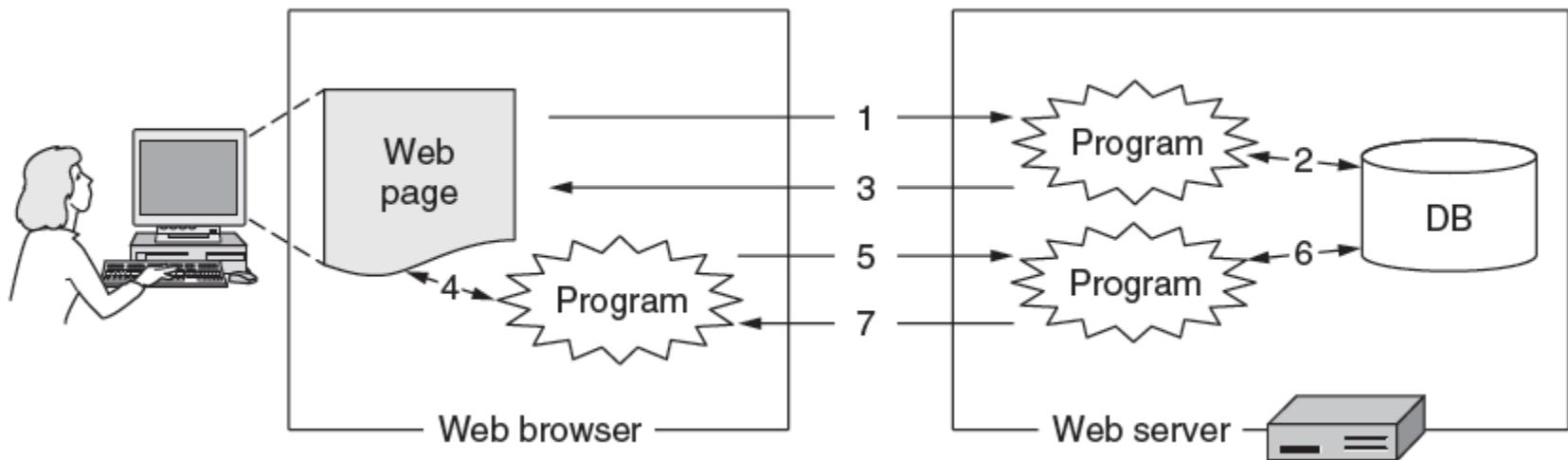
- Some differences between XHTML and HTML4
 - XHTML pages and browsers must strictly conform to the standard
 - all tags and attributes must be in lower case
 - closing tags are required
 - even for </p>. For tags that have no natural closing tag, such as
, <hr>, and , a slash must precede the closing ">,"
 - attributes must be contained within quotation marks
 - tags must nest properly
 - every document must specify its document type

7.3.3 Dynamic Web Pages and Web Applications



Steps in processing the information from an HTML form.

Dynamic Web Pages, Web Applications (2)



Dynamic pages

Server-Side Dynamic Web Page Generation

- Four techniques (languages) to generate dynamic content on the server side (**dynamic HTML**):

(1) CGI(Common Gateway Interface)

- provides an interface to allow Web servers to talk to back-end programs and scripts that can accept input (e.g. from forms) and generate HTML pages in response.
- A traditional way
- 在服务器 (cgi-bin) 上运行的一段脚本程序，最终将运行结果传回客户端
- Perl

(2) PHP(PHP: Hypertext Preprocessor)

- To embed little scripts inside HTML pages and have them be executed by the server itself to generate the page
- A popular language for writing these scripts
- <?php ... ?>tag contained in normal HTML file
- File extension *php*

Server-Side Dynamic Web Page Generation (2)

(3) JSP (Java Server Pages)

- **Similar to PHP**
- **Written in the Java language instead of in PHP**
- **File extension *jsp***

(4) ASP (Active Server Pages)

- **Microsoft's version of PHP and JSP**
- **File extension *asp***

Dynamic Web Documents (3)

```
<html>
<body>

<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>
```

A sample HTML page with embedded PHP.
(file: www.abcd.com/test.php)

Dynamic Web Documents (4)

```
<html>
<body>
<form action="action.php" method="post">

```

(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 25
</body>
</html>
```

(c)

- (a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

Client-Side Dynamic Web Page Generation

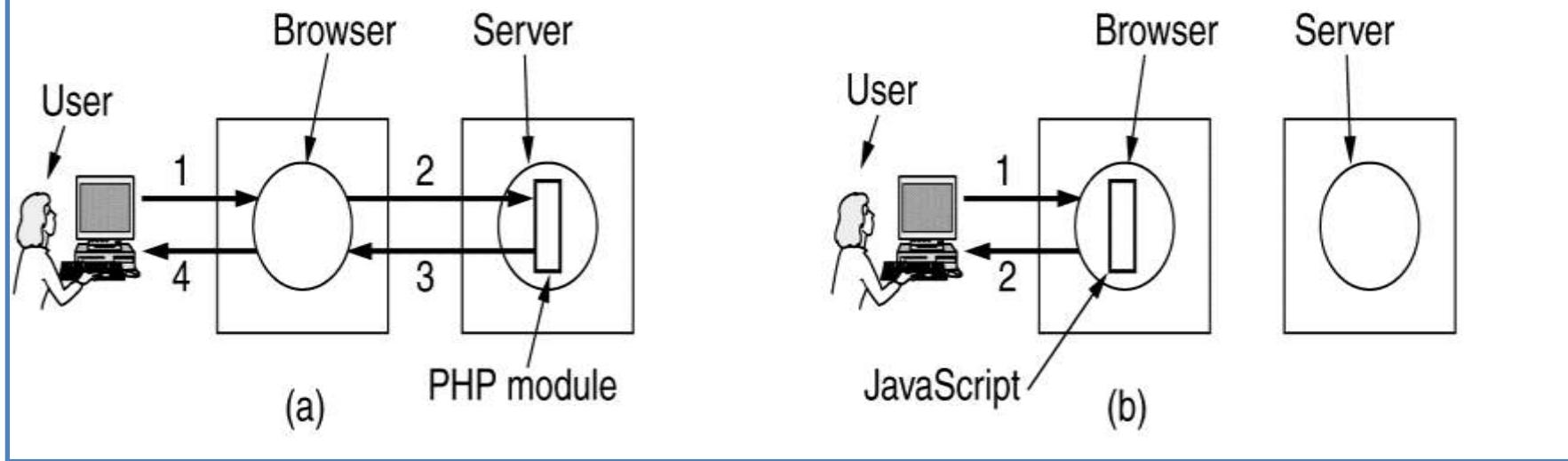
Use of JavaScript for processing a form.

Using tag *<script>*

```
<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>
```

Client-Side Dynamic Web Page Generation (2)



(a) Server-side scripting with PHP.

(b) Client-side scripting with JavaScript.

Client-Side Dynamic Web Page Generation (3)

```
<html>
<head>
<script language="javascript" type="text/javascript">

function response(test_form) {
    function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
    var r = eval(test_form.number.value);      // r = typed in argument
    document.myform.mytext.value = "Here are the results.\n";
    for (var i = 1; i <= r; i++)           // print one line from 1 to r
        document.myform.mytext.value += (i + "!" + factorial(i) + "\n");
}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

A JavaScript program for computing and printing factorials.

Client-Side Dynamic Web Page Generation (4)

```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.cs.vu.nl/~ast/im/kitten.jpg";
document.myurl[1] = "http://www.cs.vu.nl/~ast/im/puppy.jpg";
document.myurl[2] = "http://www.cs.vu.nl/~ast/im/bunny.jpg";
function pop(m) {
    var urx = "http://www.cs.vu.nl/~ast/im/cat.jpg";
    popupwin = window.open(document.myurl[m],"mywind","width=250,height=250");
}
</script>
</head>

<body>
<p> <a href="#" onMouseover="pop(0); return false;" > Kitten </a> </p>
<p> <a href="#" onMouseover="pop(1); return false;" > Puppy </a> </p>
<p> <a href="#" onMouseover="pop(2); return false;" > Bunny </a> </p>
</body>
</html>
```

An interactive Web page that responds to mouse movement.

Client-Side Dynamic Web Page Generation (5)

The various ways to generate and display content

- **JavaScript**
 - **Easier to write**
- **Java applet**
 - `<applet> ... </applet>`
 - **Execute faster**
- **ActiveX controls**
 - **Microsoft's answer to Sun's Java applets**
 - **Run fastest of all**

AJAX (Asynchronous JavaScript And XML)

- **Ajax**不是一个技术，它实际上是几种技术，每种技术都有其独特之处，合在一起就成了一个功能强大的新技术。
- **Ajax**包括：
 - **XHTML**和**CSS** 进行页面表达
 - 使用文档对象模型(**Document Object Model**)作动态显示和交互
 - 使用**XML**和**XSLT**做数据交互和操作
 - 使用**XMLHttpRequest**进行异步数据接收
 - 使用**JavaScript**将它们绑定在一起

AJAX (Asynchronous JavaScript And XML)

- **AJAX** is a set of technologies that work together to enable Web applications that are every bit as responsive and powerful as traditional desktop applications. The technologies are:
 1. **HTML and CSS** to present information as pages.
 2. **XML (eXtensible Markup Language)** to let programs exchange application data with the server.
 3. **DOM (Document Object Model)** to change parts of pages while they are viewed.
 4. **An asynchronous way** for programs to send and retrieve XML data.
 5. **JavaScript** as a language to bind all this functionality together.

AJAX (Asynchronous JavaScript And XML): XML

- XML (eXtensible Markup Language), is a language for specifying structured content.
- HTML mixes content with formatting because it is concerned with the presentation of information.
- XML allows Web content to be structured for automated processing.
- There are no defined tags for XML. Each user can define her own tags.

AJAX (Asynchronous JavaScript And XML): XML

A simple XML document.

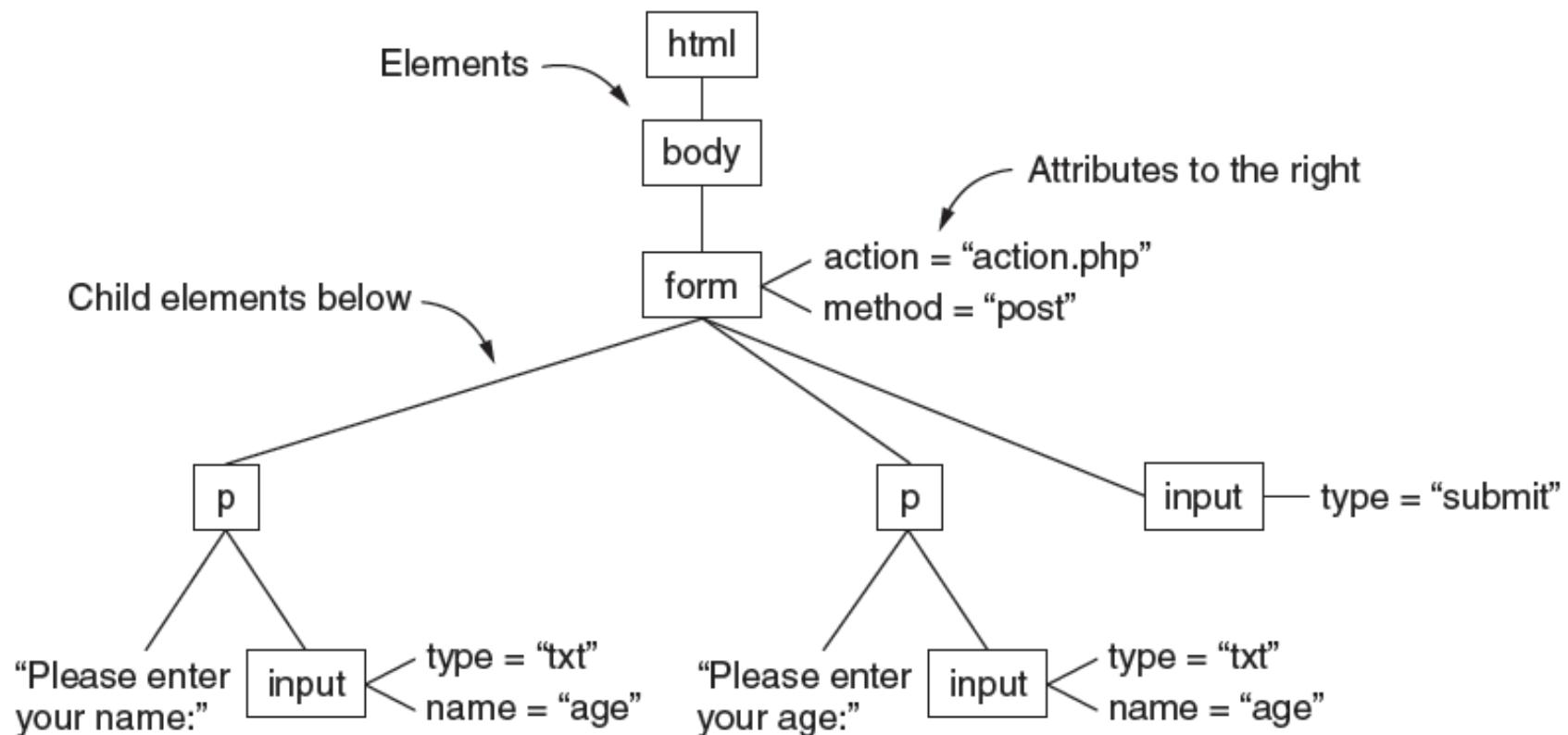
```
<?xml version="1.0" ?>
<book_list>
  <book>
    <title> Human Behavior and the Principle of Least Effort </title>
    <author> George Zipf </author>
    <year> 1949 </year>
  </book>
  <book>
    <title> The Mathematical Theory of Communication </title>
    <author> Claude E. Shannon </author>
 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

AJAX (Asynchronous JavaScript And XML): XML

- XSLT (eXtensible Stylesheet Language Transformations) can be used to define how XML should be transformed into HTML. XSLT is like CSS, but much more powerful.
- The data expressed in XML, instead of HTML, is that it is easier for programs to analyze.
- HTML is even being defined in terms of XML. → XHTML (eXtended HyperText Markup Language).
- XML has also proved popular as a language for communication between programs. → Web Service (SOAP, UDDI, WSDL)

AJAX (Asynchronous JavaScript And XML): DOM (Document Object Model)

A DOM tree example

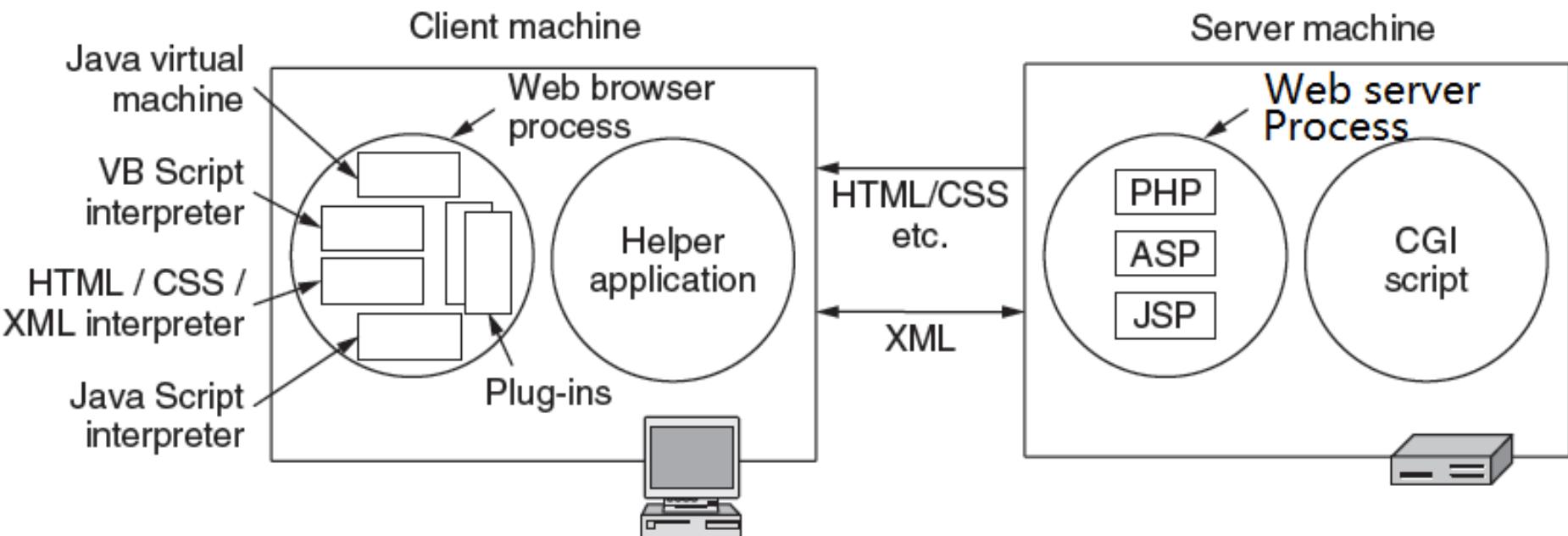


AJAX (Asynchronous JavaScript And XML): A-I/O

To provide a responsive interface in the browser while sending or receiving data, it must be possible for scripts to perform **asynchronous I/O** that does not block the display while awaiting the response to a request.

- For example, consider a map that can be scrolled in the browser. When it is notified of the scroll action, the script on the map page may request more map data from the server if the view of the map is near the edge of the data.
- The interface should not freeze while those data are fetched.
- Instead, the scrolling should continue smoothly.
- When the data arrive, the script is notified so that it can use the data. If all goes well, new map data will be fetched before it is needed.
- Modern browsers have support for this model of communication.

Client-Side Dynamic Web Page Generation (6)



The various ways to generate and display content.

7.3.4 HTTP– Hyper Text Transfer Protocol

- ASCII request
- RFC 822 MIME-like response
- TCP Connection: port 80 on server
- HTTP 1.1, supports persistent connections
 - To establish a TCP connection, send a request and get a response, and then send additional requests and get additional responses

HTTP (2)

- Connections
- Methods
- Message Headers (parameters)
 - Request headers
 - Response headers
- Caching
- Experimenting with HTTP
 - telnet www.ietf.org 80

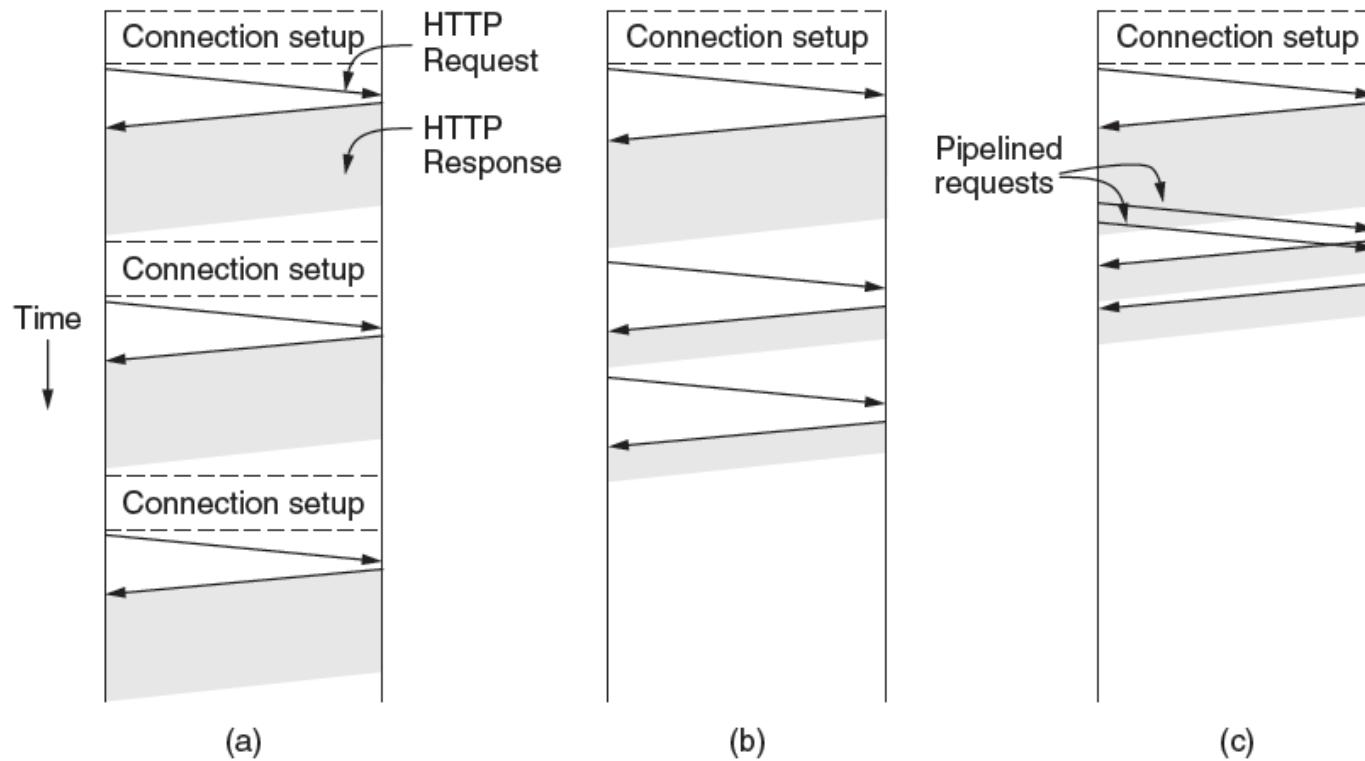
HTTP: Connections (3)

- HTTP 1.0 with multiple connections
 - Establish a TCP connection
 - Send a request
 - Get a single response
 - Terminate the TCP connection
- HTTP 1.1 with persistent connections
 - Establish a TCP connection
 - Looping for sending a request and getting a single response
 - Terminate the TCP connection
- HTTP 1.1 with persistent connections and pipeline requests.

HTTP: Connections (4)

HTTP with

- (a) multiple connections and sequential requests.
- (b) A persistent connection and sequential requests.
- (c) A persistent connection and pipelined requests.



HTTP Methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

The built-in HTTP request methods.

HTTP: Methods (2)

- Each request consists of one or more lines of ASCII text, with the first word on the first line being the name of the method requested. The method names are case sensitive, so **GET** is allowed but not **get**.
 - The **GET method** requests the server to send the page.
 - The **HEAD method** just asks for the message header, without the actual page. This method can be used to collect information for indexing purposes, or just to test a URL for validity.
 - The **POST method** is used when forms are submitted. Both it and GET are also used for SOAP Web services.
 - It uploads data to the server (i.e., the contents of the form or RPC parameters).
 - The server then does something with the data that depends on the URL, conceptually appending the data to the object.
 - Finally, the method returns a page indicating the result.

HTTP: Methods (3)

- The **PUT method** is the reverse of GET: instead of reading the page, it writes the page. This method makes it possible to build a collection of Web pages on a remote server.
- **DELETE** does what you might expect: it removes the page, or at least it indicates that the Web server has agreed to remove the page. As with PUT, authentication and permission play a major role here.
- The **TRACE method** is for debugging. It instructs the server to send back the request.
- The **CONNECT method** lets a user make a connection to a Web server through an intermediate device, such as a Web cache.
- The **OPTIONS method** provides a way for the client to query the server for a page and obtain the methods and headers that can be used with that page.

HTTP Methods (4)

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

The **status code** response groups.

HTTP Message Headers

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

Some HTTP message headers.

Example HTTP Usage

```
telnet www.ietf.org 80 >log
```

```
GET /rfc.html HTTP/1.1
```

```
Host:www.ietf.org
```

```
close
```

Example HTTP Usage

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^].
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211
Content-Type: text/html
X-Pad: avoid browser bug

<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript">
function url() {
    var x = document.form1.number.value
    if (x.length == 1) {x = "000" + x }
    if (x.length == 2) {x = "00" + x }
    if (x.length == 3) {x = "0" + x }
    document.form1.action = "/rfc/rfc" + x + ".txt"
    document.form1.submit
}
</script>

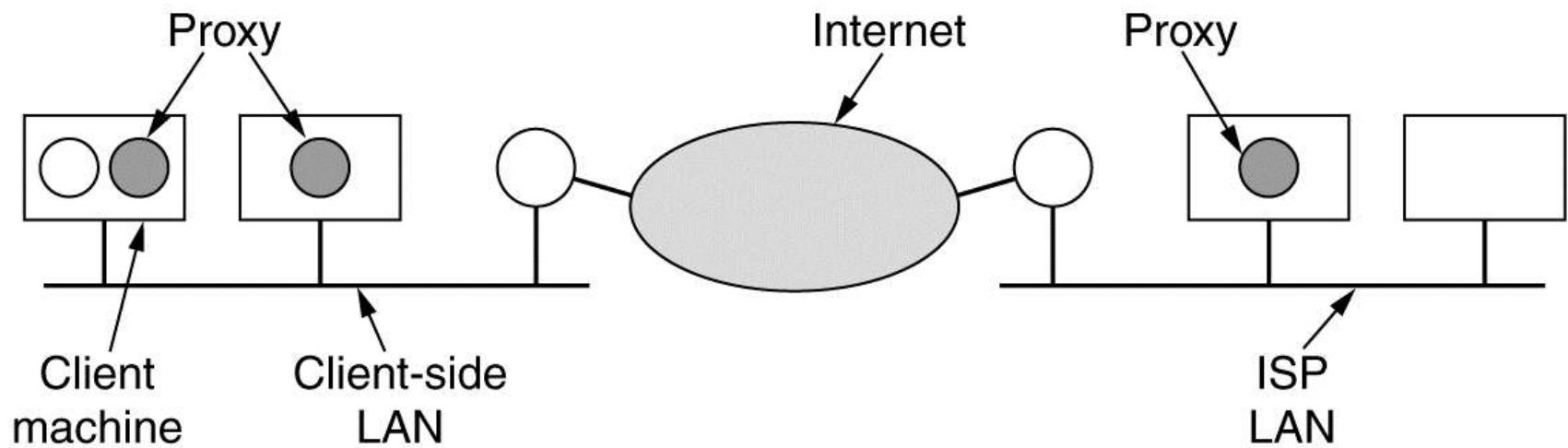
</head>
```

The start of the output of
www.ietf.org/rfc.html.

7.3.5 Performance Enhancements

- 3 methods to improve network performance
 - Caching
 - Server replication (or called Mirroring)
 - Content delivery networks (CDN, 内容分发网络)
 - Content providers
 - ICP vs. ISP

Caching— Client-side technique



Hierarchical caching with three proxies.

Mirroring— Server-side technique

- **Server Replication**, also called **Mirroring**
- **Static**
- **Flash crowds** (瞬间拥挤)

Content Delivery Networks

```
<html>
<head> <title> Furry Video </title> </head>
<body>
<h1> Furry Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="bears.mpg"> Bears Today </a> <br>
<a href="bunnies.mpg"> Funny Bunnies </a> <br>
<a href="mice.mpg"> Nice Mice </a> <br>
</body>
</html>
```

(a)

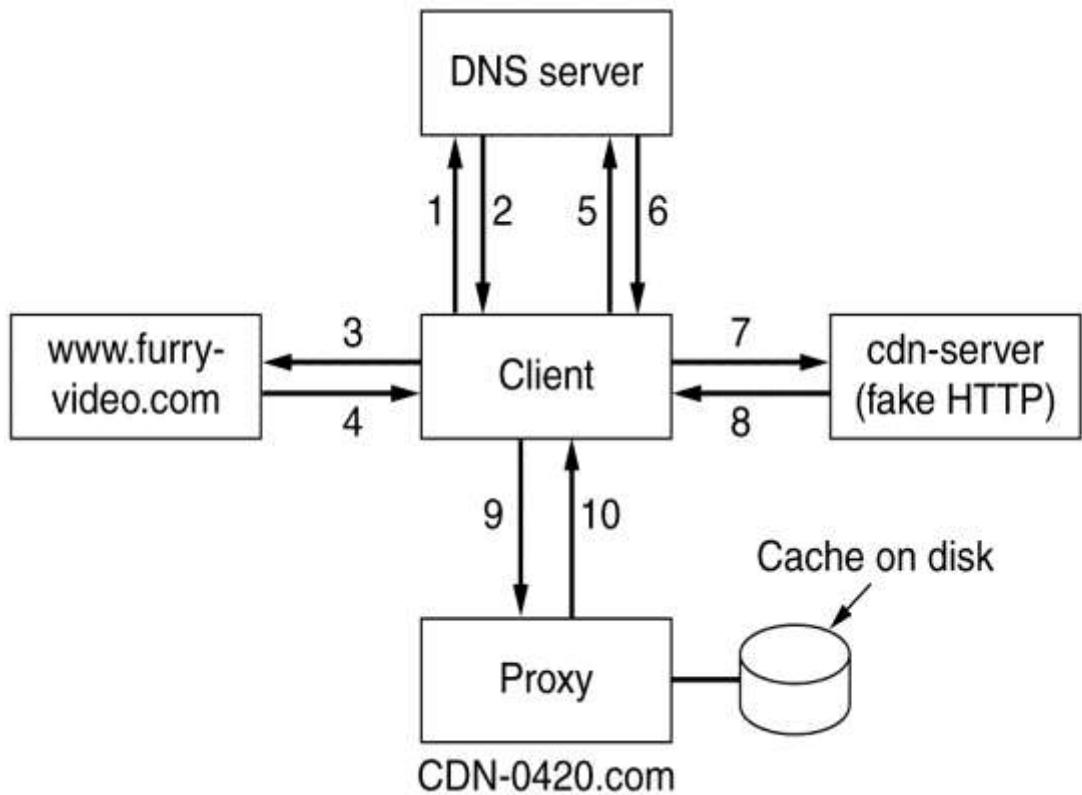
```
<html>
<head> <title> Furry Video </title> </head>
<body>
<h1> Furry Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="http://cdn-server.com/furryvideo/bears.mpg"> Bears Today </a> <br>
<a href="http://cdn-server.com/furryvideo/bunnies.mpg"> Funny Bunnies </a> <br>
<a href="http://cdn-server.com/furryvideo/mice.mpg"> Nice Mice </a> <br>
</body>
</html>
```

(b)

(a) Original Web page. (b) Same page after transformation (on www.furryvideo.com).

Content Delivery Networks(2)



1. Look up www.furryvideo.com
2. Furry's IP address returned
3. Request HTML page from Furry
4. HTML page returned
5. After click, look up cdn-server.com
6. IP address of cdn-server returned
7. Ask cdn-server for bears.mpg
8. Client told to redirect to CDN-0420.com
9. Request bears.mpg
10. Cached file bears.mpg returned

Steps in looking up a URL when a CDN is used.

7.3.5 The Mobile Web

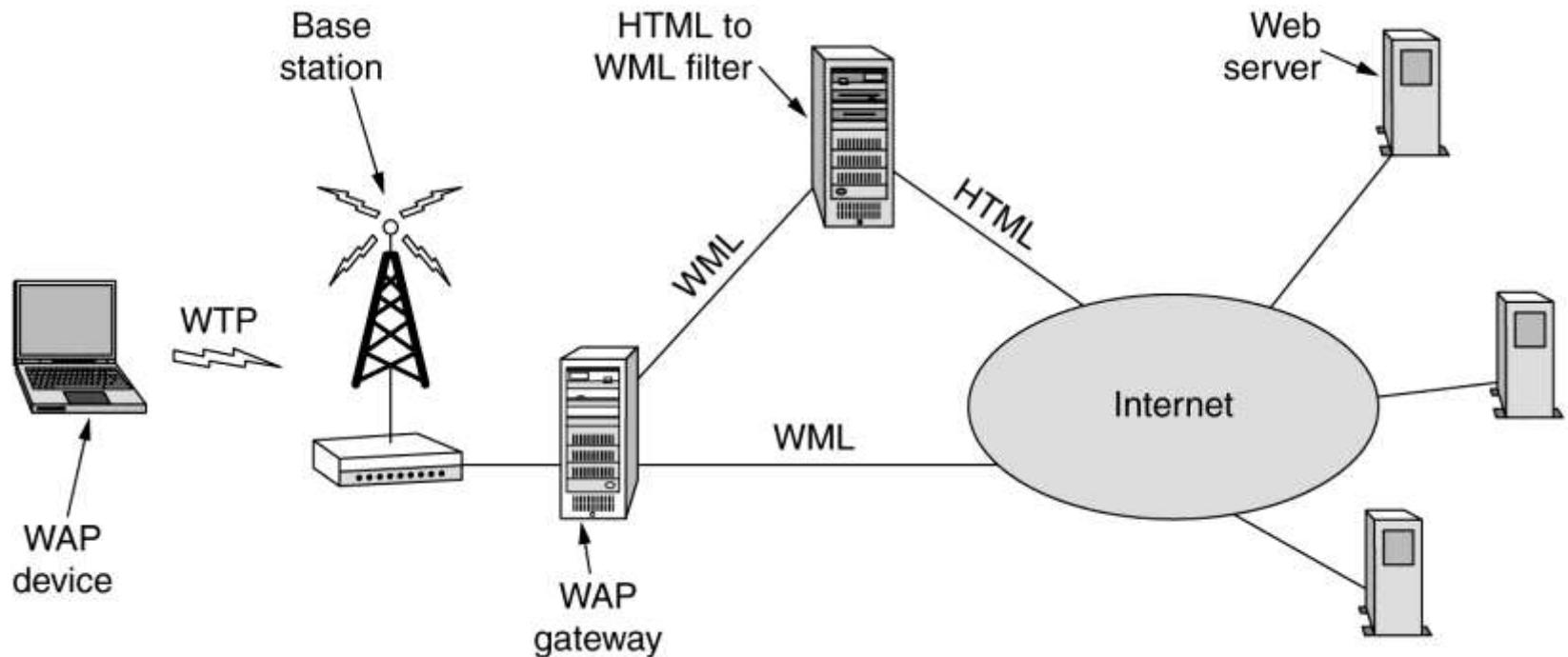
- WAP (Wireless Application Protocol) 1.0 failed!
- I-Mode successful only in Japan!
- WAP 2.0 useless!
- Any need for new protocol stack? **NO!!!**

The Mobile Web

Wireless application environment (WAE)
Wireless session protocol (WSP)
Wireless transaction protocol (WTP)
Wireless transport layer security (WTLS)
Wireless datagram protocol (WDP)
Bearer layer (GSM, CDMA, D-AMPS, GPRS, etc.)

The WAP protocol stack.

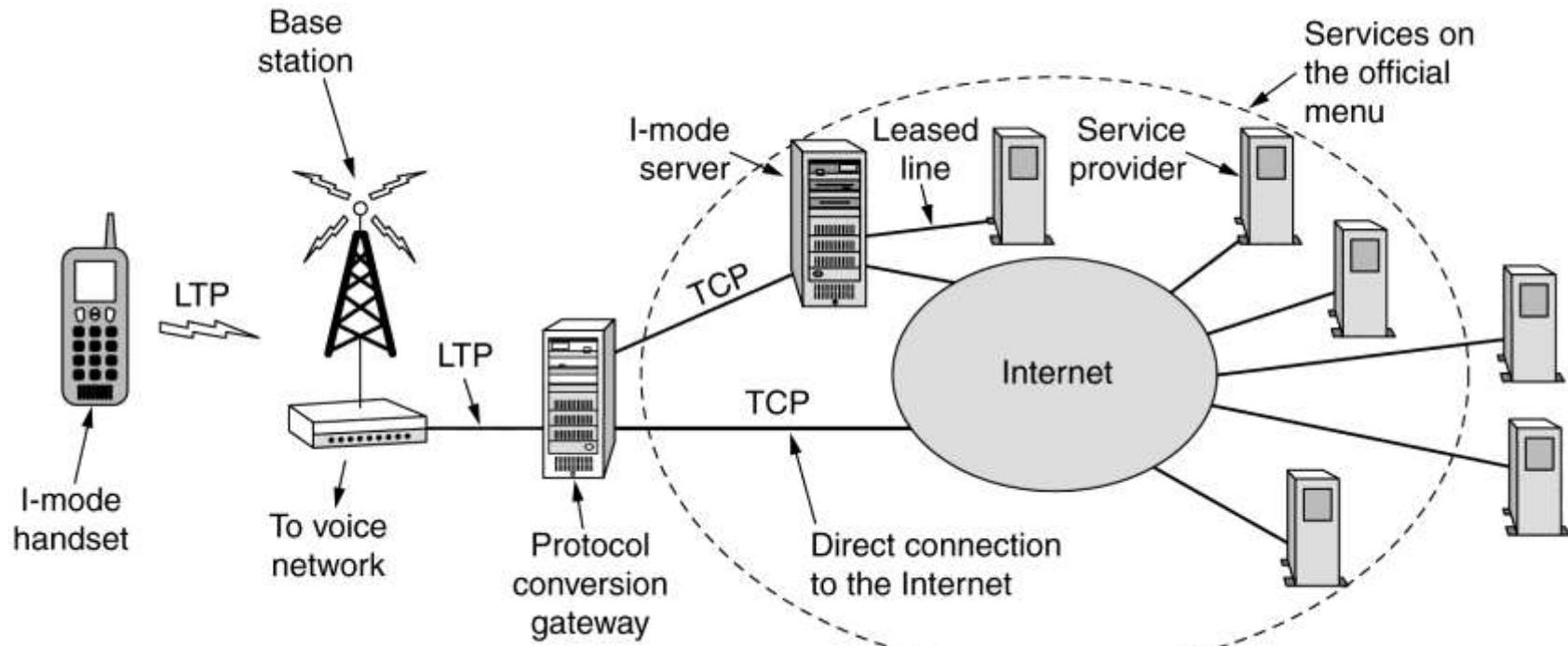
WAP (2)



The WAP architecture.

WML – Wireless Markup Language

I-Mode

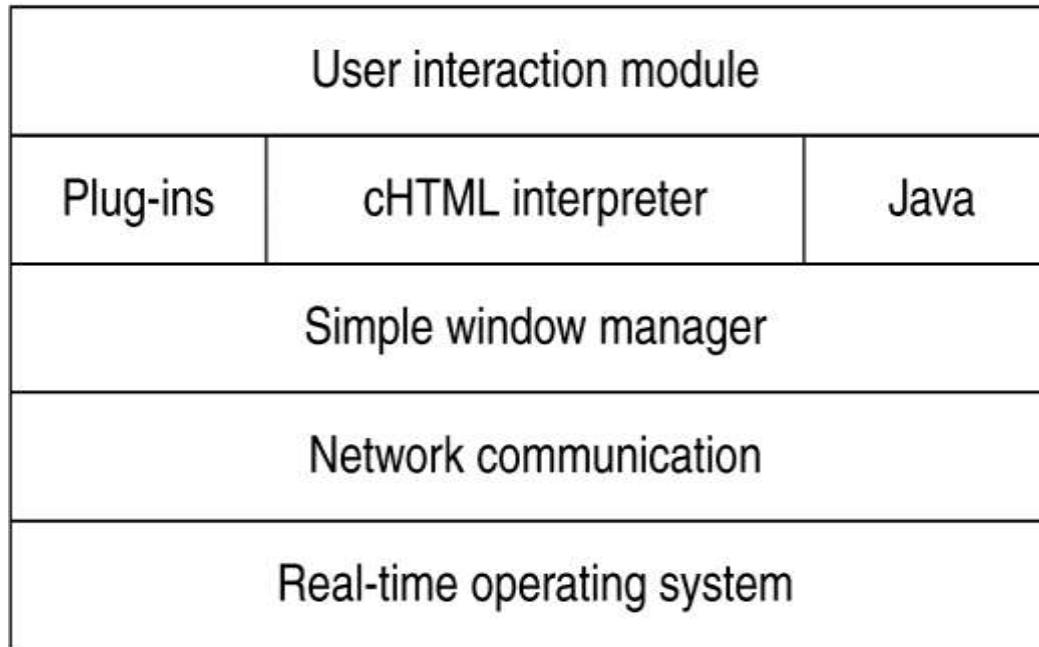


Structure of the i-mode data network showing the transport protocols.

I-Mode – Information mode (Japan, 一个日本女子发明)

It was a huge success in Japan, but WAP 1.0 was failed

I-Mode (2)



Structure of the i-mode software.

cHTML – compact HTML, a subset of HTML

I-Mode (3)

The time has com
e the walrus sai
d to talk of man
y things. Of sho
es and ships and
sealing wax of c

(a)

The time has
come the walrus
said to talk of
many things. Of
shoes and ships
and sealing wax

(b)

Lewis Carroll meets a 16 x 16 screen.

I-Mode (4)

```
<html>
<body>
<h1> Select an option </h1>
<a href="messages.chtml" accesskey="1"> Check voicemail </a> <br>
<a href="mail.chtml" accesskey="2"> Check e-mail </a> <br>
<a href="games.chtml" accesskey="3"> Play a game </a>
</body>
</html>
```

An example of cHTML file.

Second-Generation Wireless Web

Feature	WAP	I-mode
What it is	Protocol stack	Service
Device	Handset, PDA, notebook	Handset
Access	Dial up	Always on
Underlying network	Circuit-switched	Two: circuit + packet
Data rate	9600 bps	9600 bps
Screen	Monochrome	Color
Markup language	WML (XML application)	cHTML
Scripting language	WMLscript	None
Usage charges	Per minute	Per packet
Pay for shopping	Credit card	Phone bill
Pictograms	No	Yes
Standardization	WAP forum open standard	NTT DoCoMo proprietary
Where used	Europe, Japan	Japan
Typical user	Businessman	Young woman

A comparison of first-generation WAP and i-mode.

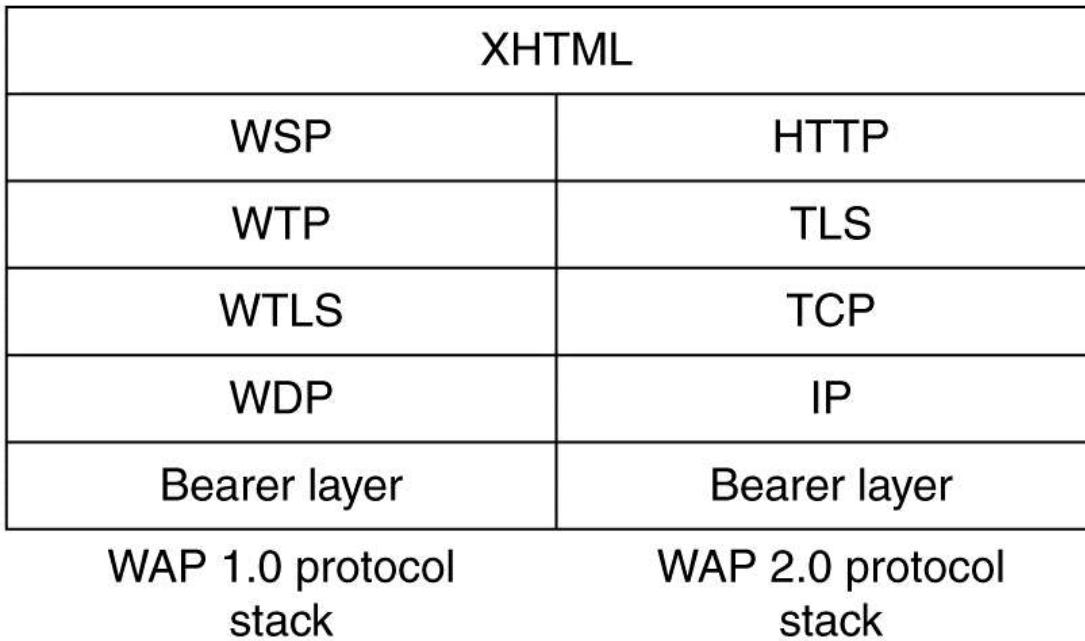
Second-Generation Wireless Web (2)

- WAP 1.0 虽然是国际标准， but it was failed
- I-Mode 很成功，但因其非标准，它只是日本年轻人玩得电子小产品

如何找到一种能够适合上述2种情况的新的技术与标准？ ➔

- New features of WAP 2.0.
- Push model as well as pull model.
- Support for integrating telephony into applications.
- Multimedia messaging.
- Inclusion of 264 pictograms(象形文字).
- Interface to a storage device. (flash ROM)
- Support for plug-ins in the browser.

Second-Generation Wireless Web (3)



WAP 2.0 supports two protocol stacks.

Second-Generation Wireless Web (4)

Module	Req.?	Function	Example tags
Structure	yes	Doc. structure	body, head, html, title
Text	yes	Information	br, code, dfn, em, hn, kbd, p, strong
Hypertext	yes	Hyperlinks	a
List	yes	Itemized lists	dl, dt, dd, ol, ul, li
Forms	No	Fill-in forms	form, input, label, option, textarea
Tables	No	Rectangular tables	caption, table, td, th, tr
Image	No	Pictures	img
Object	No	Applets, maps, etc.	object, param
Meta-information	No	Extra info	meta
Link	No	Similar to <a>	link
Base	No	URL starting point	base

The XHTML Basic modules and tags.

7.3.6 Web Search

- Web search companies
 - Google, Yahoo, Bing
 - Baidu
- How does web search works?
 - How to find web pages? **Web crawling**
 - How to store web pages? **Hard disks**
 - How to process web pages? **Indexing techniques**
 - How to make profit from web search? **Advertising**
 -

3

文件传送协议FTP

文件传送协议FTP概述

- 文件传送协议 **FTP (File Transfer Protocol)** 是因特网上使用得最广泛的文件传送协议。
- **FTP** 提供交互式的访问，允许客户指明文件的类型与格式，并允许文件具有存取权限。
- **FTP** 屏蔽了各计算机系统的细节，因而适合于在异构网络中任意计算机之间传送文件。
- **RFC 959** 很早就成为了因特网的正式标准。

网络环境下复制文件的复杂性

- (1) 计算机存储数据的格式不同。
- (2) 文件的目录结构和文件命名的规定不同。
- (3) 对于相同的文件存取功能，操作系统使用的命令不同。
- (4) 访问控制方法不同。

FTP 特点

- 文件传送协议 **FTP** 只提供文件传送的一些基本的服务，它使用 **TCP** 可靠的运输服务。
- **FTP** 的主要功能是减少或消除在不同操作系统下处理文件的不兼容性。
- **FTP** 使用客户服务器方式。一个 **FTP** 服务器进程可同时为多个客户进程提供服务。**FTP** 的服务器进程由两大部分组成：一个主进程，负责接受新的请求；另外有若干个从属进程，负责处理单个请求。

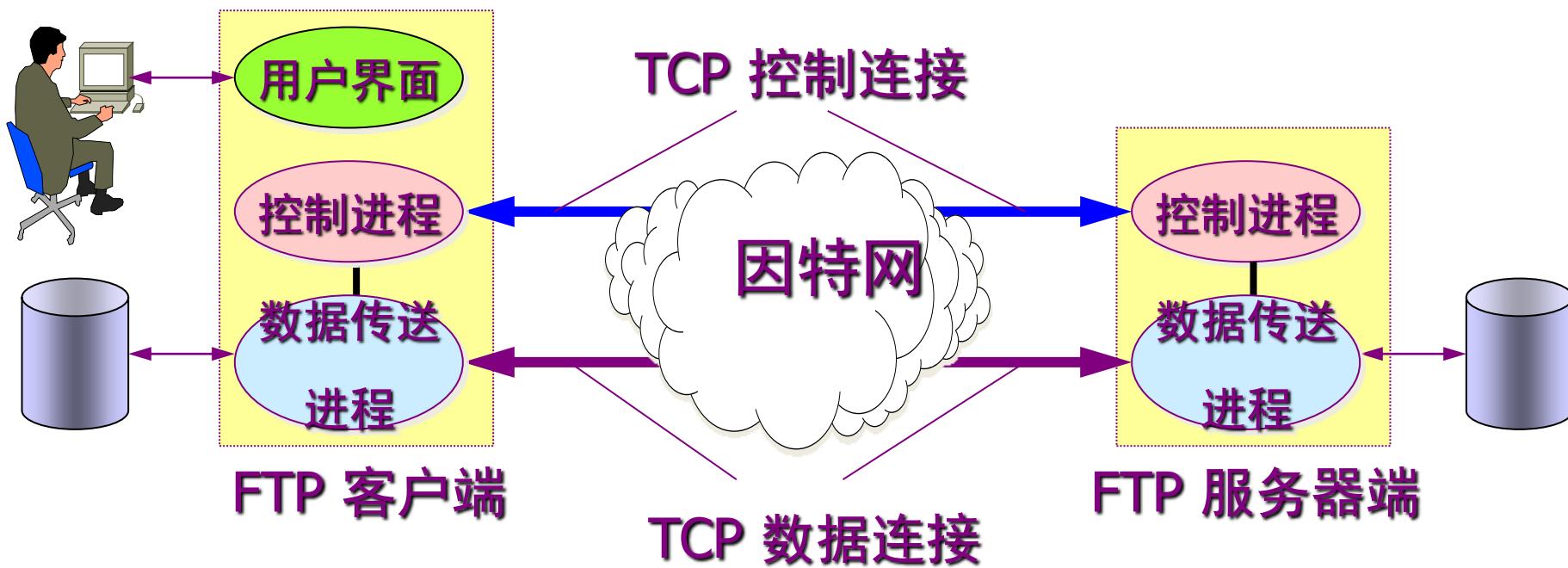
主进程的工作步骤如下

- 打开熟知端口（端口号为 21），使客户进程能够连接上。
- 等待客户进程发出连接请求。
- 启动从属进程来处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。
- 回到等待状态，继续接受其他客户进程发来的请求。主进程与从属进程的处理是并发地进行。

FTP使用两个连接

- **控制连接**在整个会话期间一直保持打开，FTP 客户发出的传送请求通过控制连接发送给服务器端的控制进程，但控制连接不用来传送文件。
- 实际用于传输文件的是“**数据连接**”。服务器端的控制进程在接收到 FTP 客户发送来的文件传输请求后就创建“数据传送进程”和“数据连接”，用来连接客户端和服务器端的数据传送进程。
- 数据传送进程实际完成文件的传送，在传送完毕后关闭“数据传送连接”并结束运行。

FTP 工作原理



两个不同的端口号

- 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熟知端口**(21)**，同时还要告诉服务器进程自己的另一个端口号，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熟知端口**(20)**与客户进程所提供的端口号建立数据传送连接。
- 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。

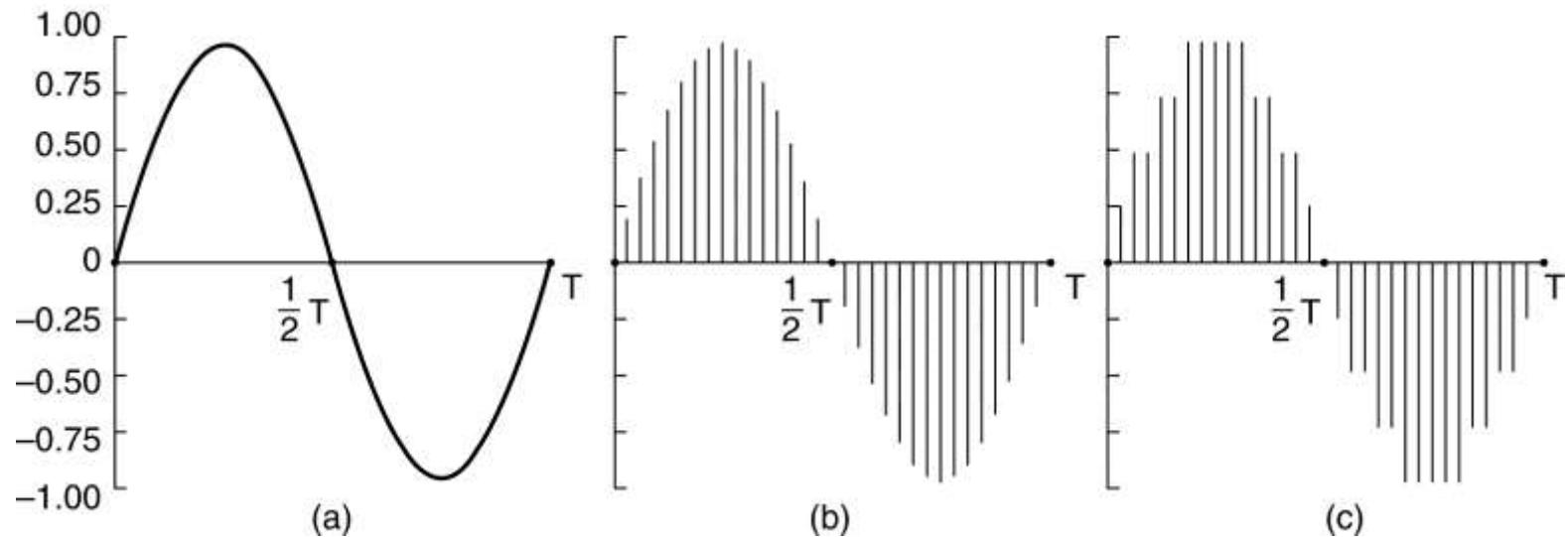
使用两个不同端口号的好处

- 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）。

7.4 Streaming Audio and Video

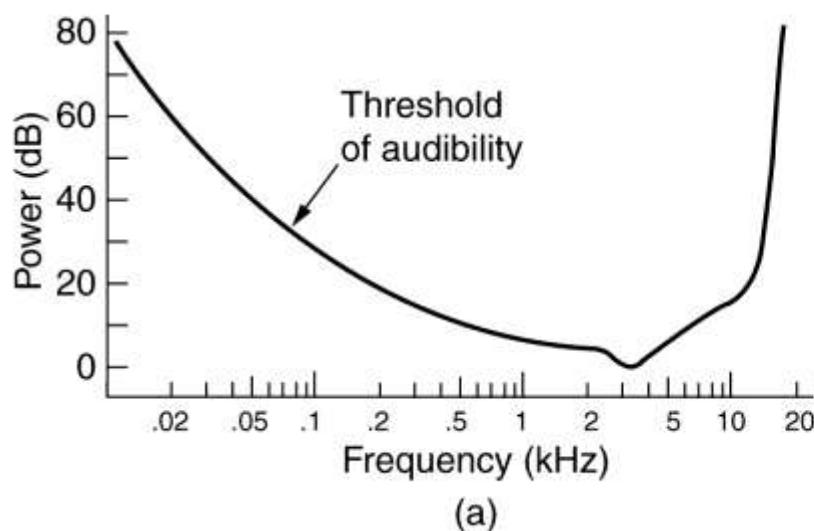
- **Introduction to Audio**
- **Audio Compression**
- **Streaming Audio**
- **Internet Radio**
- **Voice over IP**
- **Introduction to Video**
- **Video Compression**
- **Video on Demand**
- **The MBone – The Multicast Backbone**

Introduction to Audio

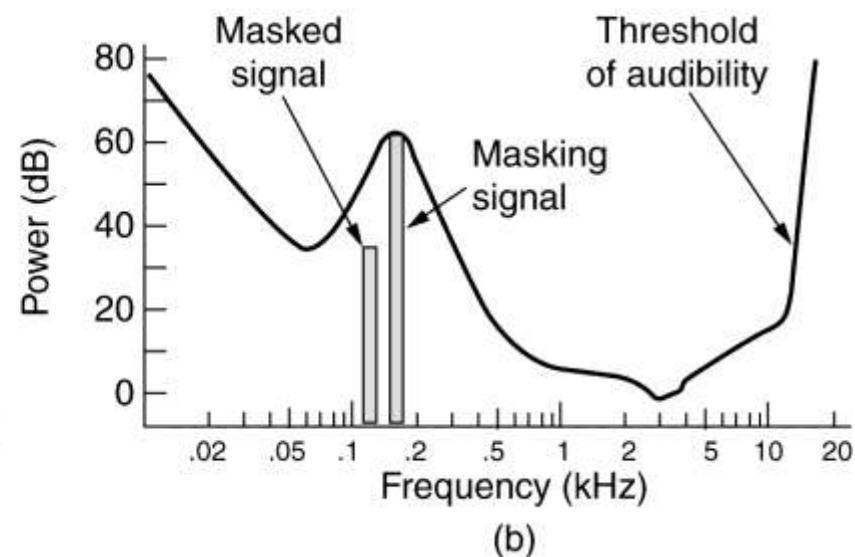


- (a) A sine wave. (b) Sampling the sine wave.
(c) Quantizing the samples to 4 bits.

Audio Compression



(a)

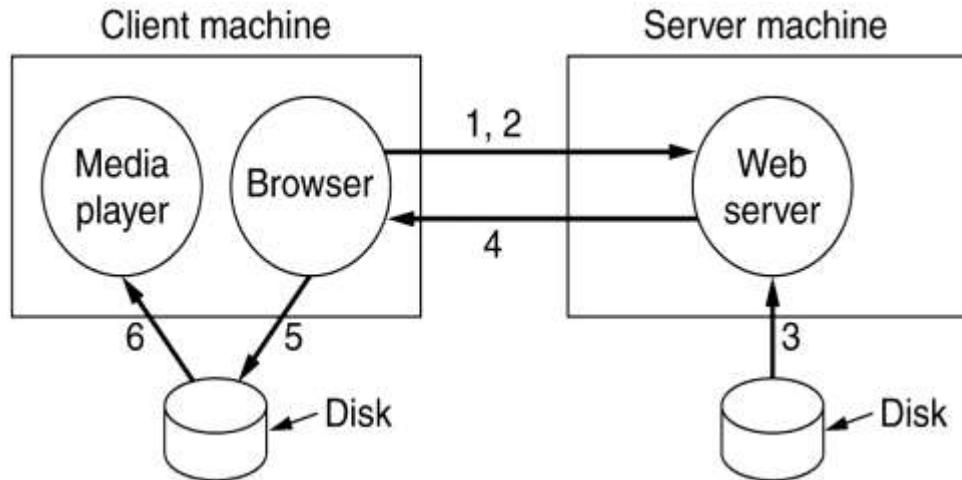


(b)

(a) The threshold of audibility as a function of frequency.

(b) The masking effect.

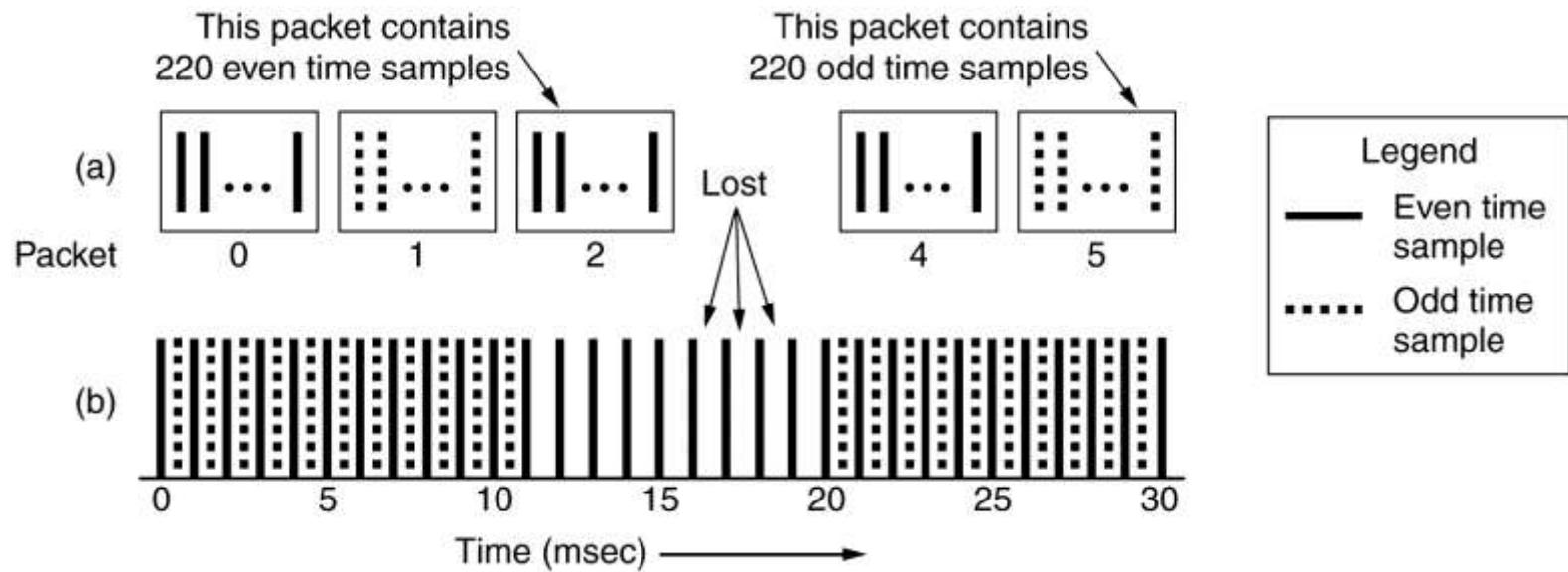
Streaming Audio



1. Establish TCP connection
2. Send HTTP GET request
3. Server gets file from disk
4. File sent back
5. Browser writes file to disk
6. Media player fetches file block by block and plays it

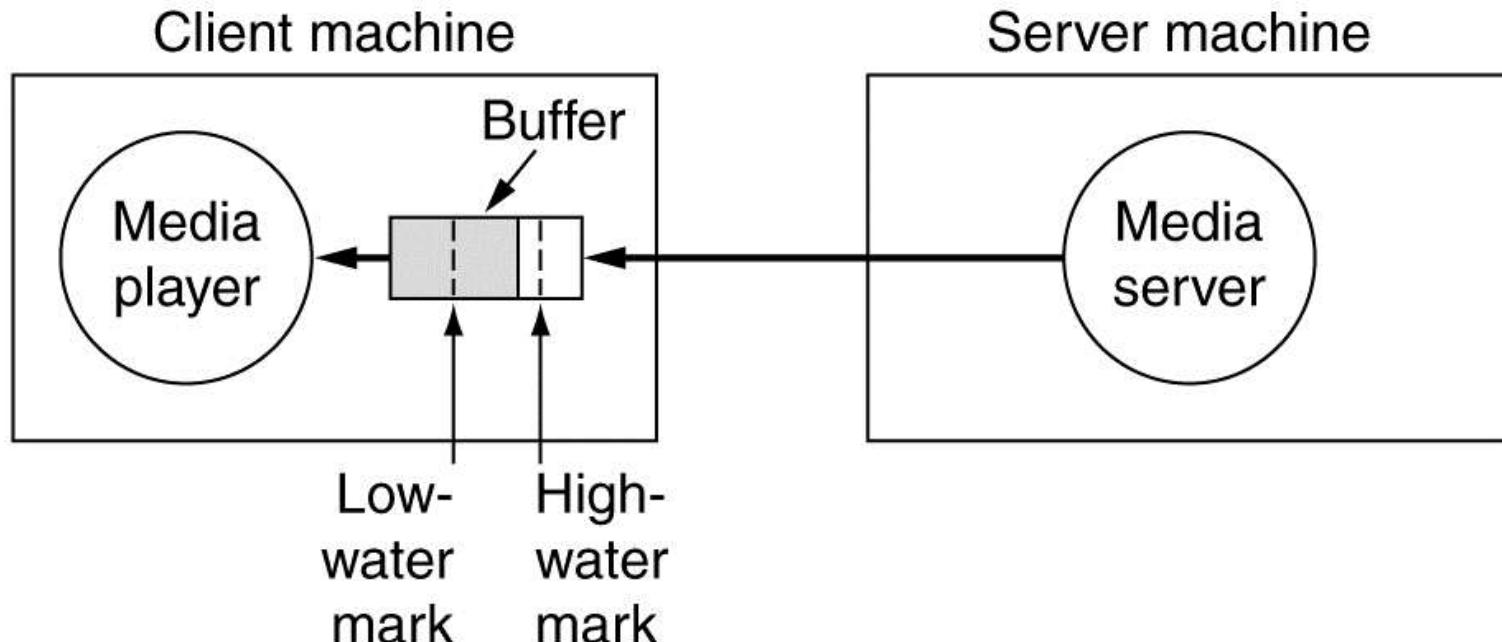
A straightforward way to implement
clickable music on a Web page.

Streaming Audio (2)



When packets carry **alternate samples**, the loss of a packet reduces the temporal resolution rather than creating a gap in time.

Streaming Audio (3)



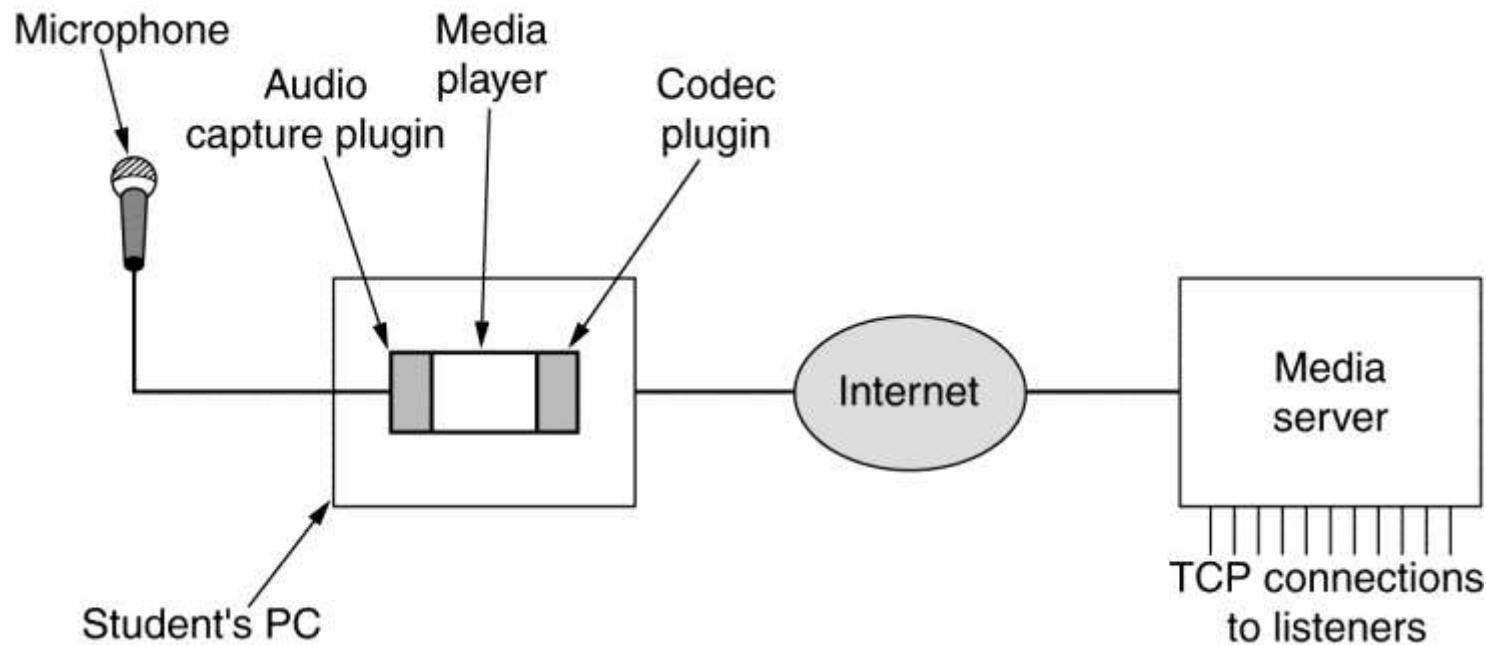
The media player **buffers** input from the media server and plays from the buffer rather than directly from the network.

Streaming Audio (4)

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

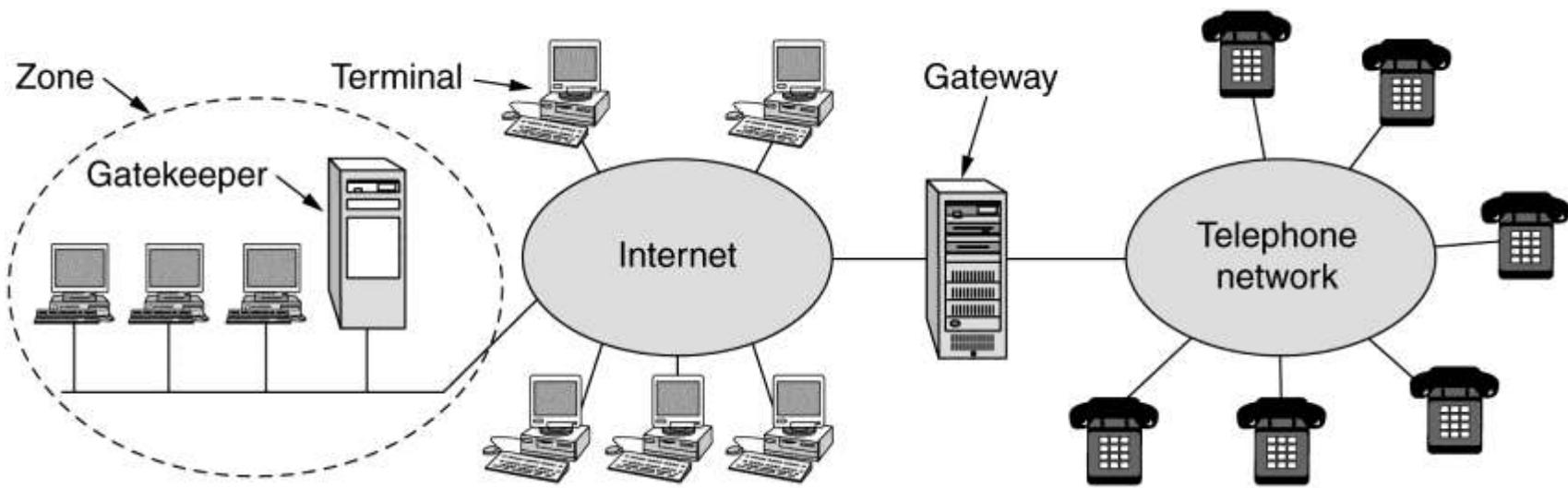
RTSP commands from the player to the server.

Internet Radio



A student radio station.

Voice over IP



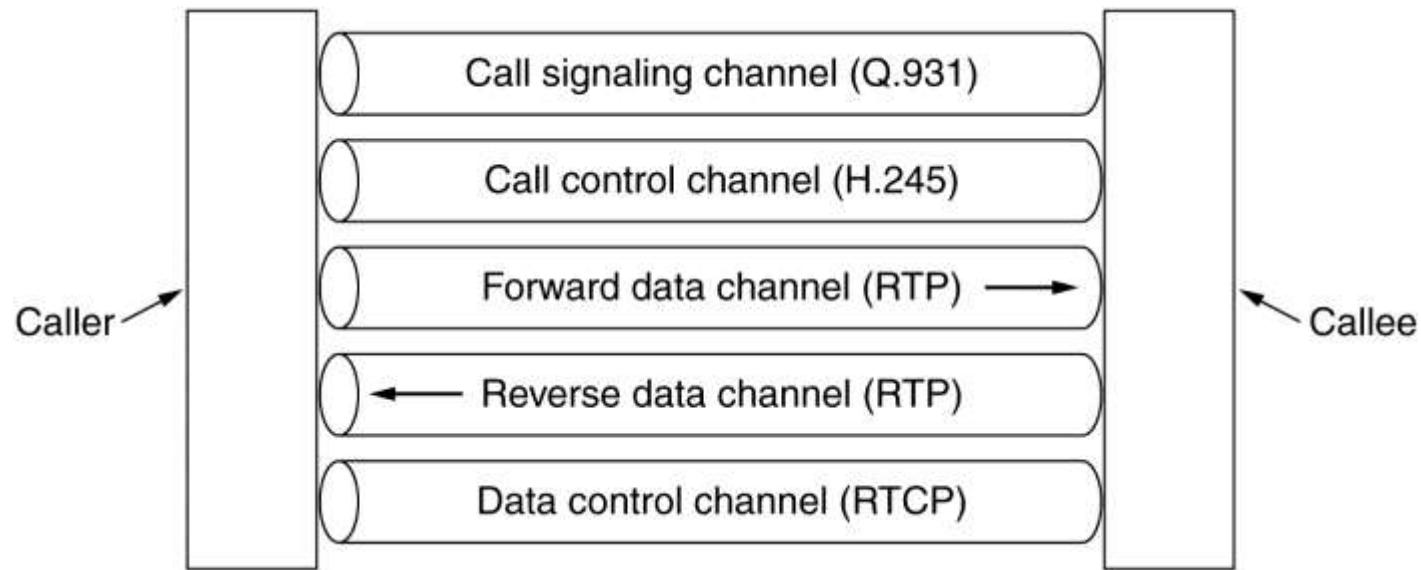
The H.323 architectural model for Internet telephony. [designed by ITU]

Voice over IP (2)

Speech	Control							
G.7xx	RTCP	H.225 (RAS)	Q.931 (Call signaling)	H.245 (Call control)				
RTP								
UDP		TCP						
IP								
Data link protocol								
Physical layer protocol								

The H.323 protocol stack.

Voice over IP (3)



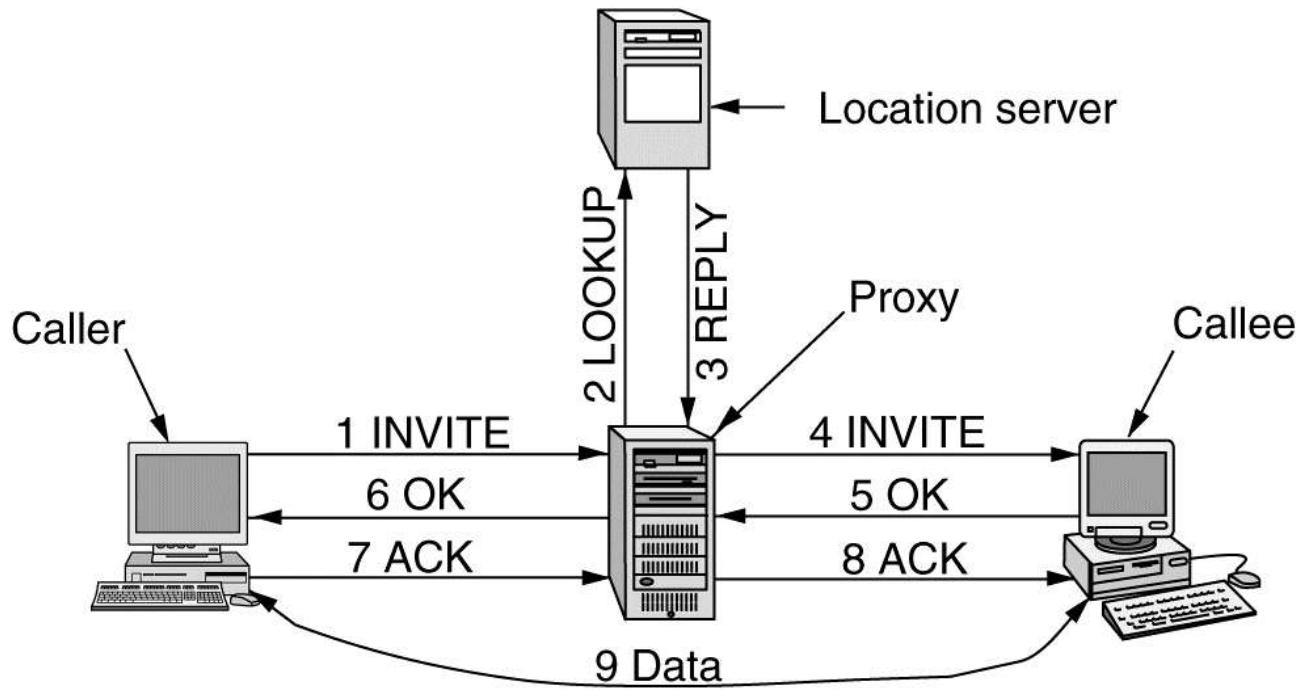
Logical channels between the caller and callee during a call.

SIP – The Session Initiation Protocol

Method	Description
INVITE	Request initiation of a session
ACK	Confirm that a session has been initiated
BYE	Request termination of a session
OPTIONS	Query a host about its capabilities
CANCEL	Cancel a pending request
REGISTER	Inform a redirection server about the user's current location

The SIP methods defined in the core specification. [designed by IETF]

SIP (2)

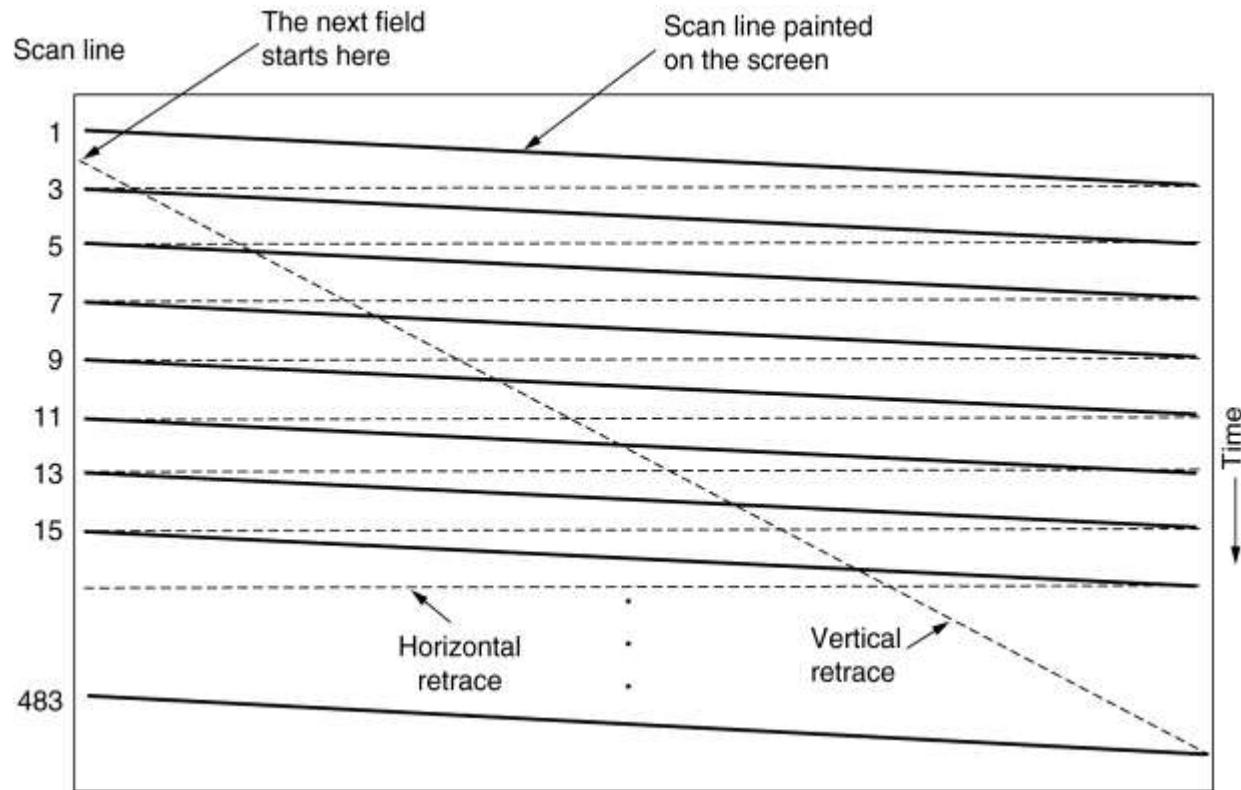


Use a proxy and redirection servers with SIP.

Comparison of H.323 and SIP

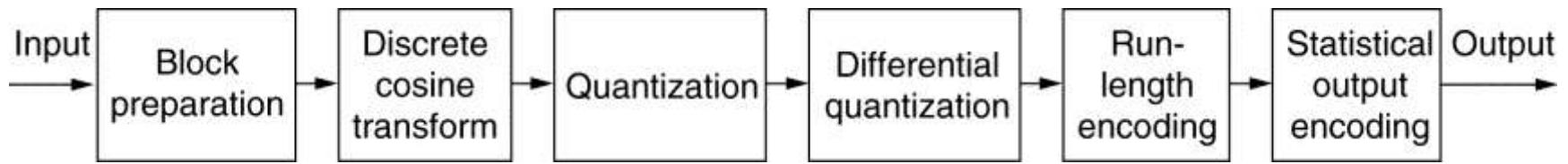
Item	H.323	SIP
Designed by	ITU	IETF
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	No	Yes
Architecture	Monolithic	Modular
Completeness	Full protocol stack	SIP just handles setup
Parameter negotiation	Yes	Yes
Call signaling	Q.931 over TCP	SIP over TCP or UDP
Message format	Binary	ASCII
Media transport	RTP/RTCP	RTP/RTCP
Multiparty calls	Yes	Yes
Multimedia conferences	Yes	No
Addressing	Host or telephone number	URL
Call termination	Explicit or TCP release	Explicit or timeout
Instant messaging	No	Yes
Encryption	Yes	Yes
Size of standards	1400 pages	250 pages
Implementation	Large and complex	Moderate
Status	Widely deployed	Up and coming

Video Analog Systems



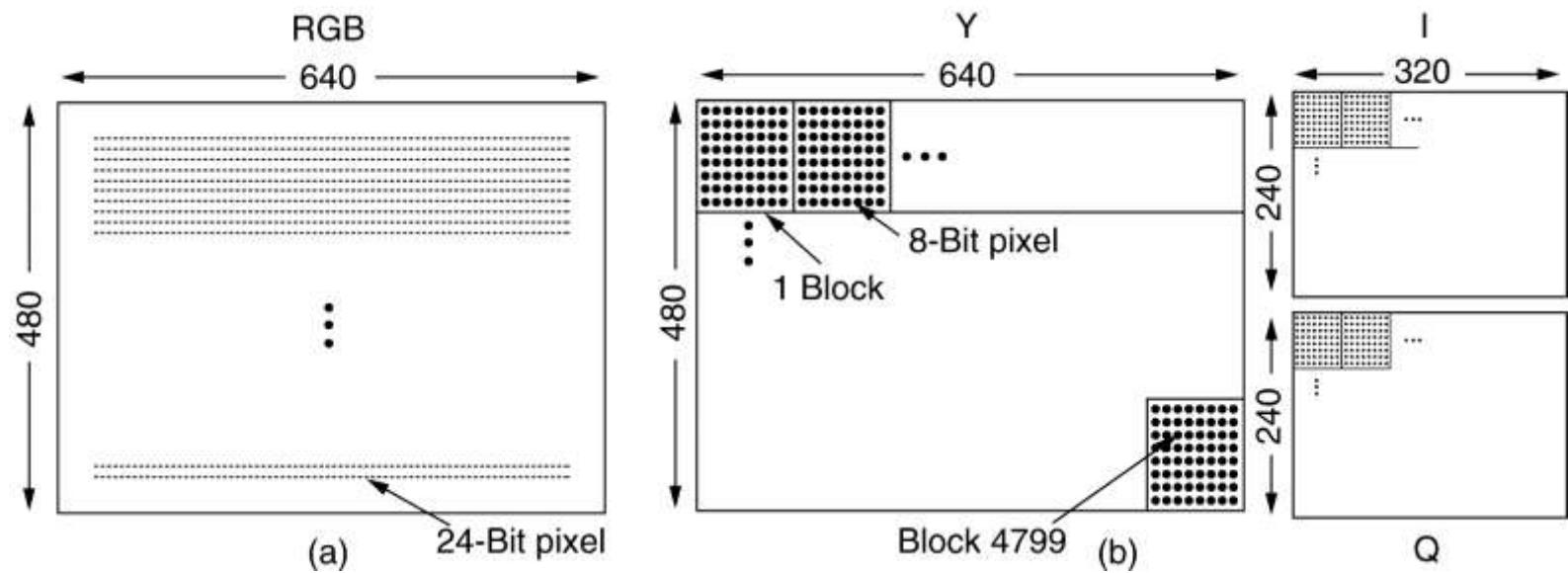
The scanning pattern used for NTSC video and television.

The JPEG Standard



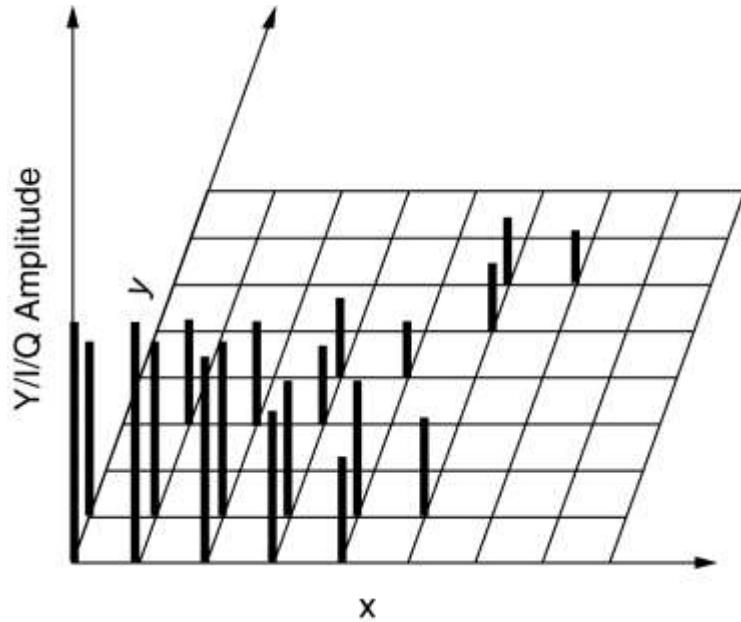
The operation of JPEG in lossy sequential mode.

The JPEG Standard (2)

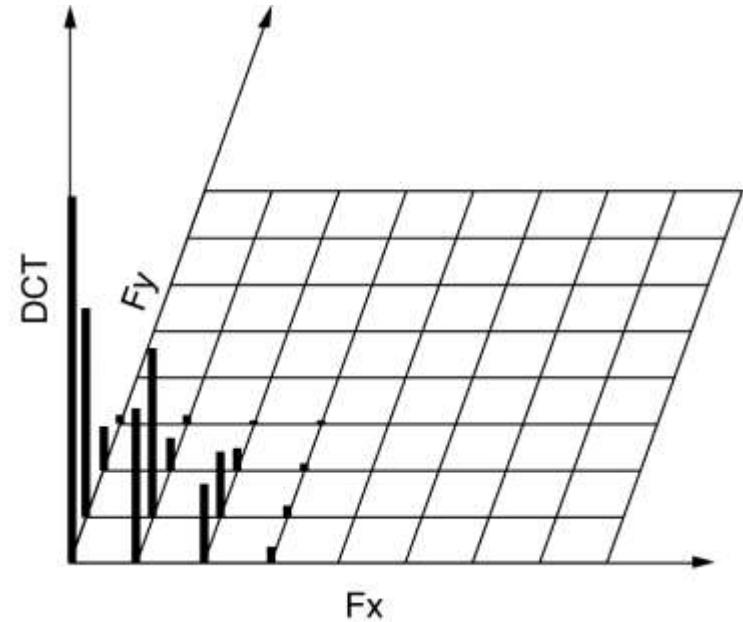


- (a) RGB input data.**
(b) After block preparation.

The JPEG Standard (3)



(a)



(b)

- (a) One block of the Y matrix.
- (b) The DCT coefficients.

The JPEG Standard (4)

DCT Coefficients

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

Quantization table

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

Quantized coefficients

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

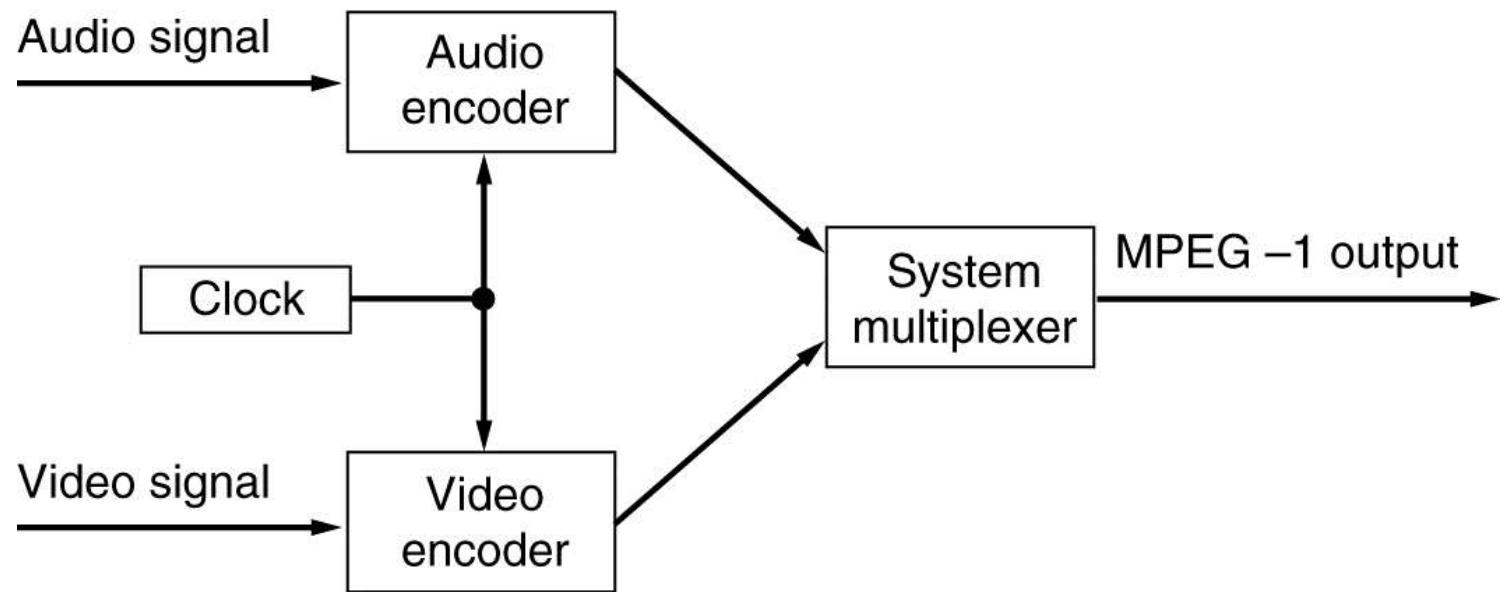
Computation of the quantized DTC coefficients.

The JPEG Standard (5)

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

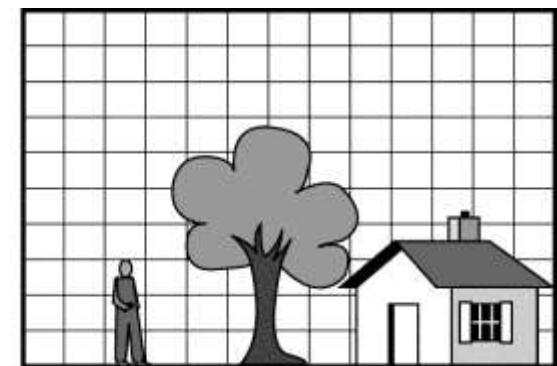
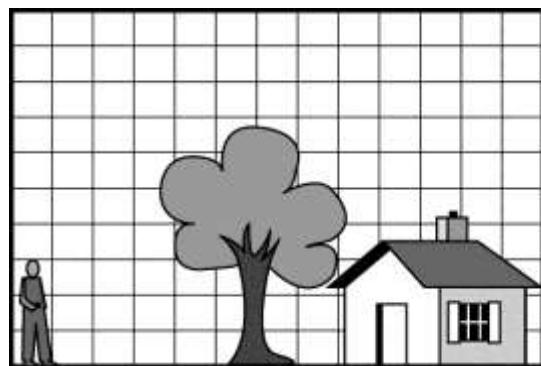
The order in which the quantized values are transmitted.

The MPEG Standard



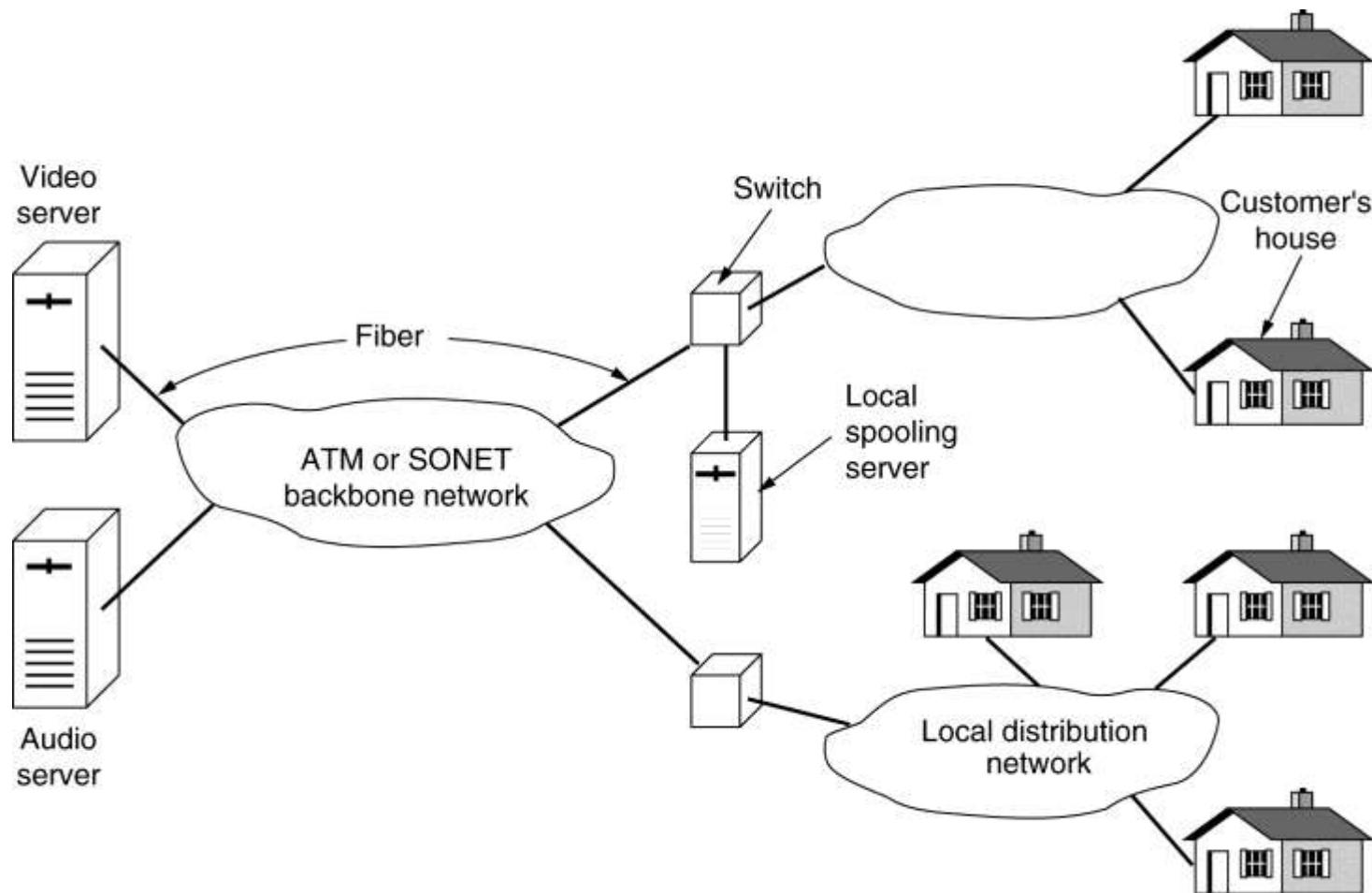
Synchronization of the audio and video streams in MPEG-1.

The MPEG Standard (2)



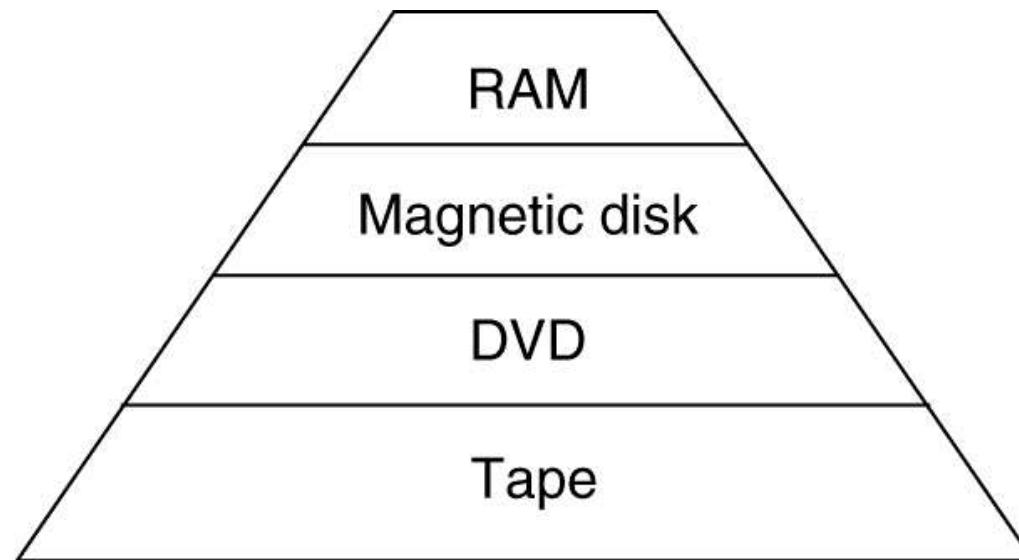
Three consecutive frames.

Video on Demand



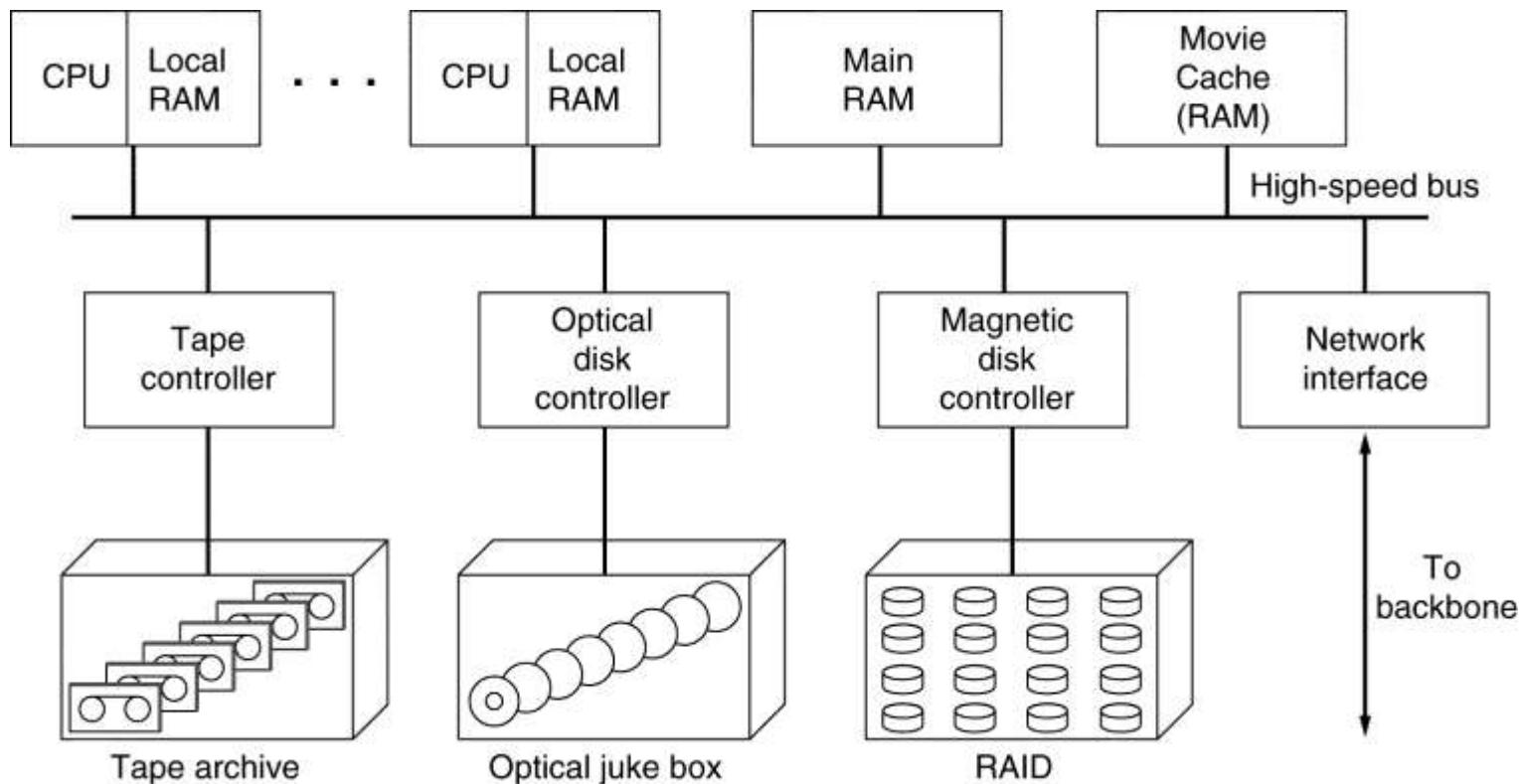
Overview of a video-on-demand system.

Video Servers



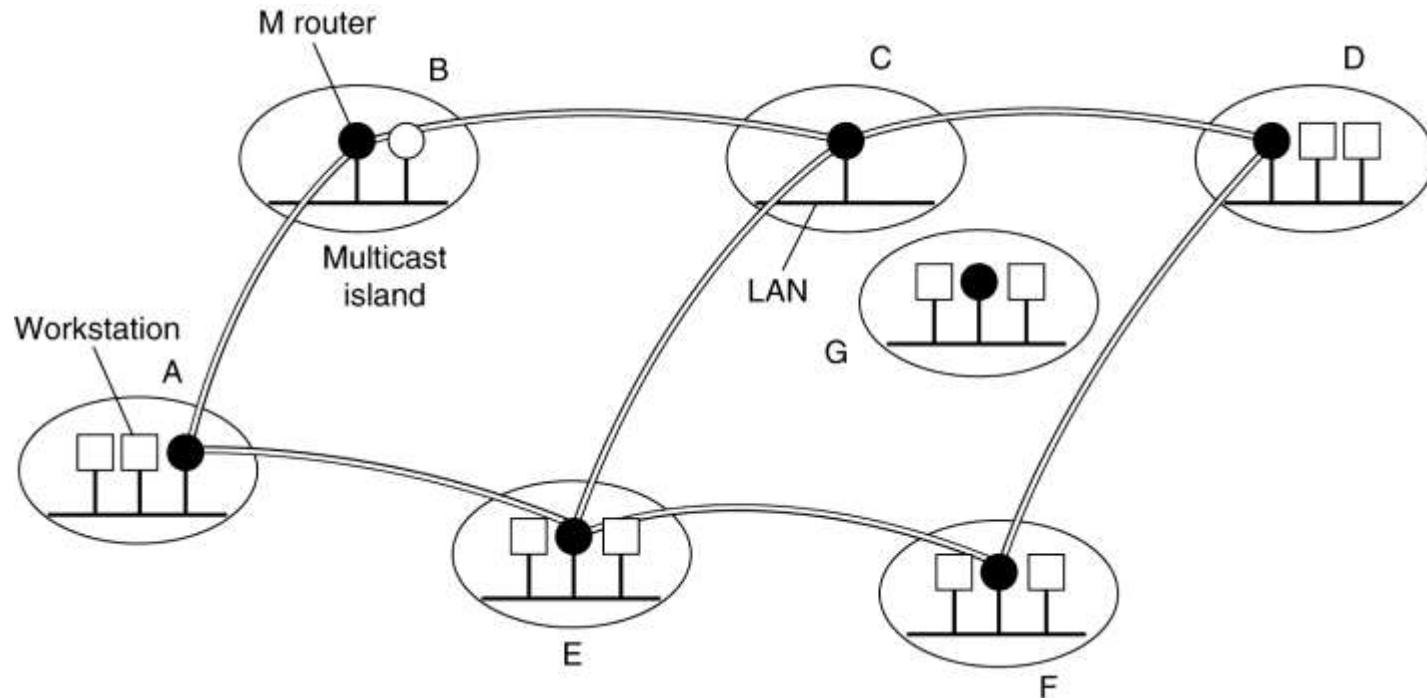
A video server storage hierarchy.

Video Servers (2)



The hardware architecture of a typical video server.

The MBone – The Multicast Backbone

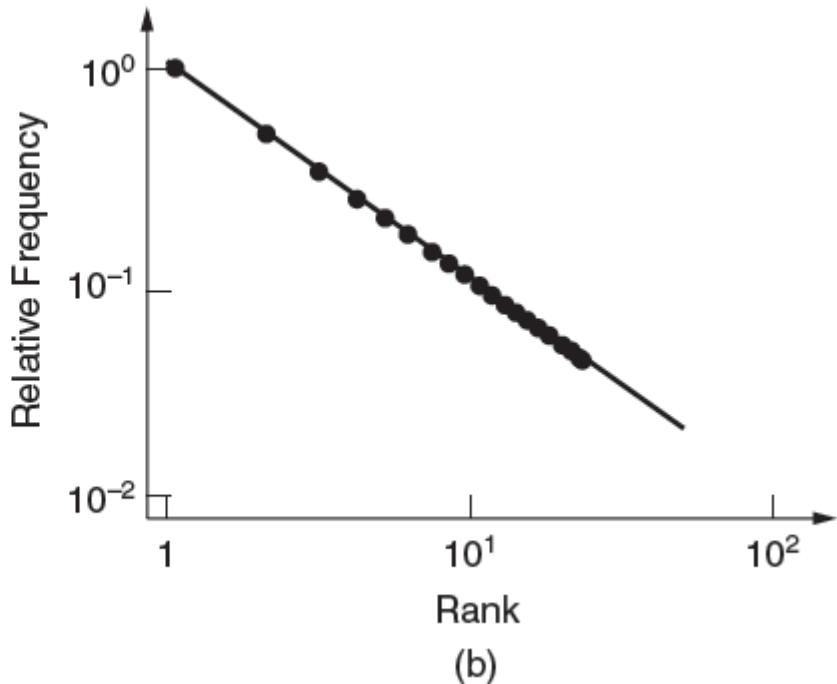
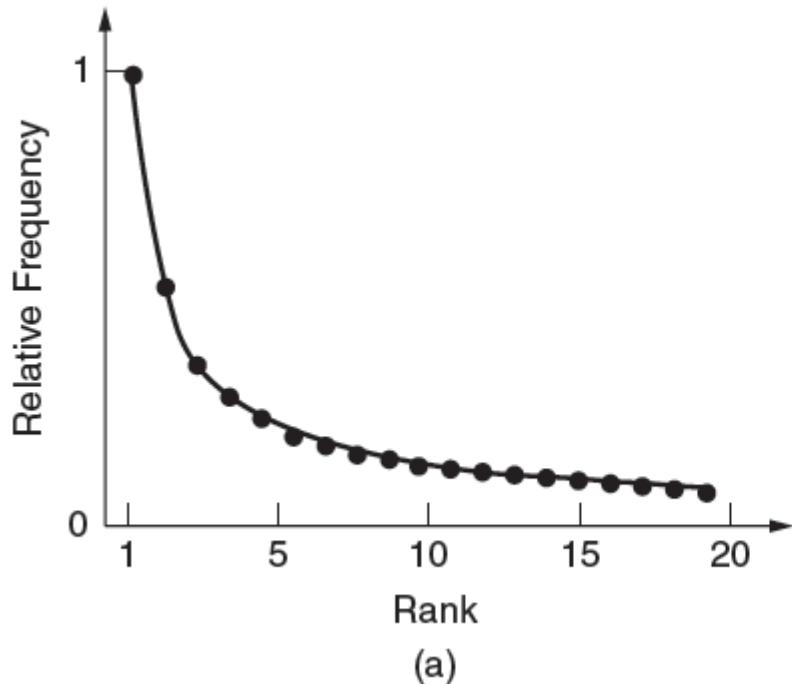


MBone consists of multicast islands connected by tunnels.

7.5 Content Delivery (*)

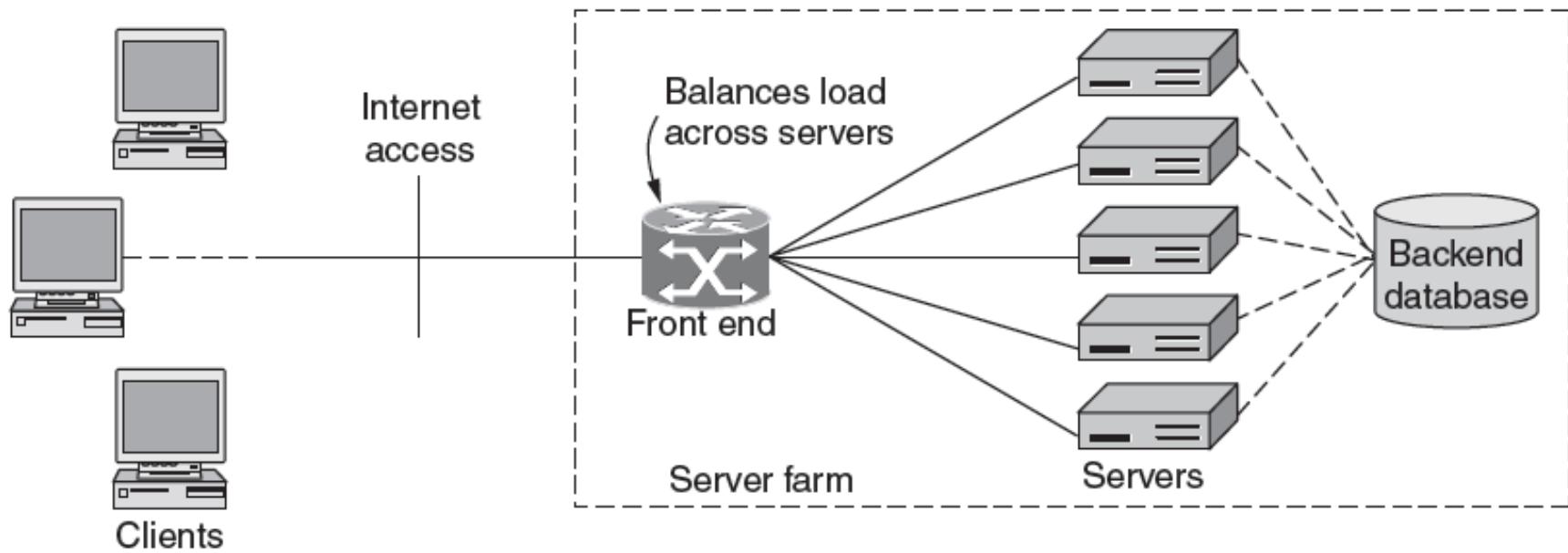
- **Content and internet traffic**
- **Server farms and web proxies**
- **Content delivery networks**
- **Peer-to-peer networks**

7.5.1 Content and Internet Traffic



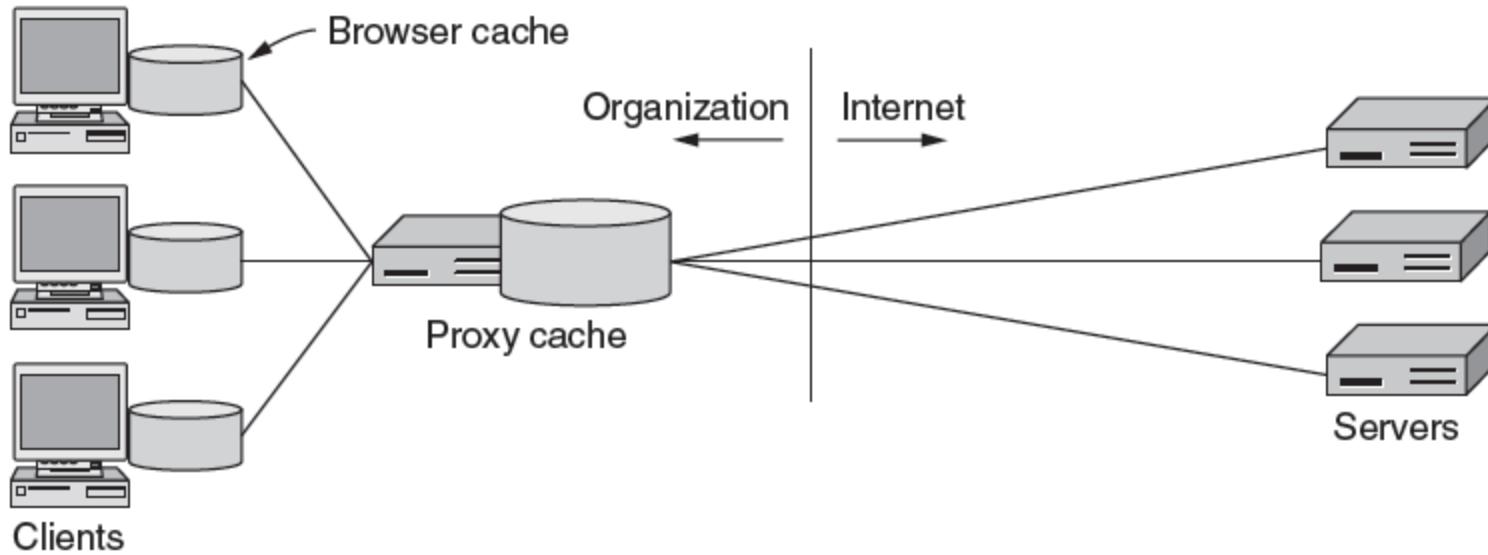
Zipf distribution (a) On a linear scale. (b) On a log-log scale.

7.5.2 Server Farms and Web Proxies (1)



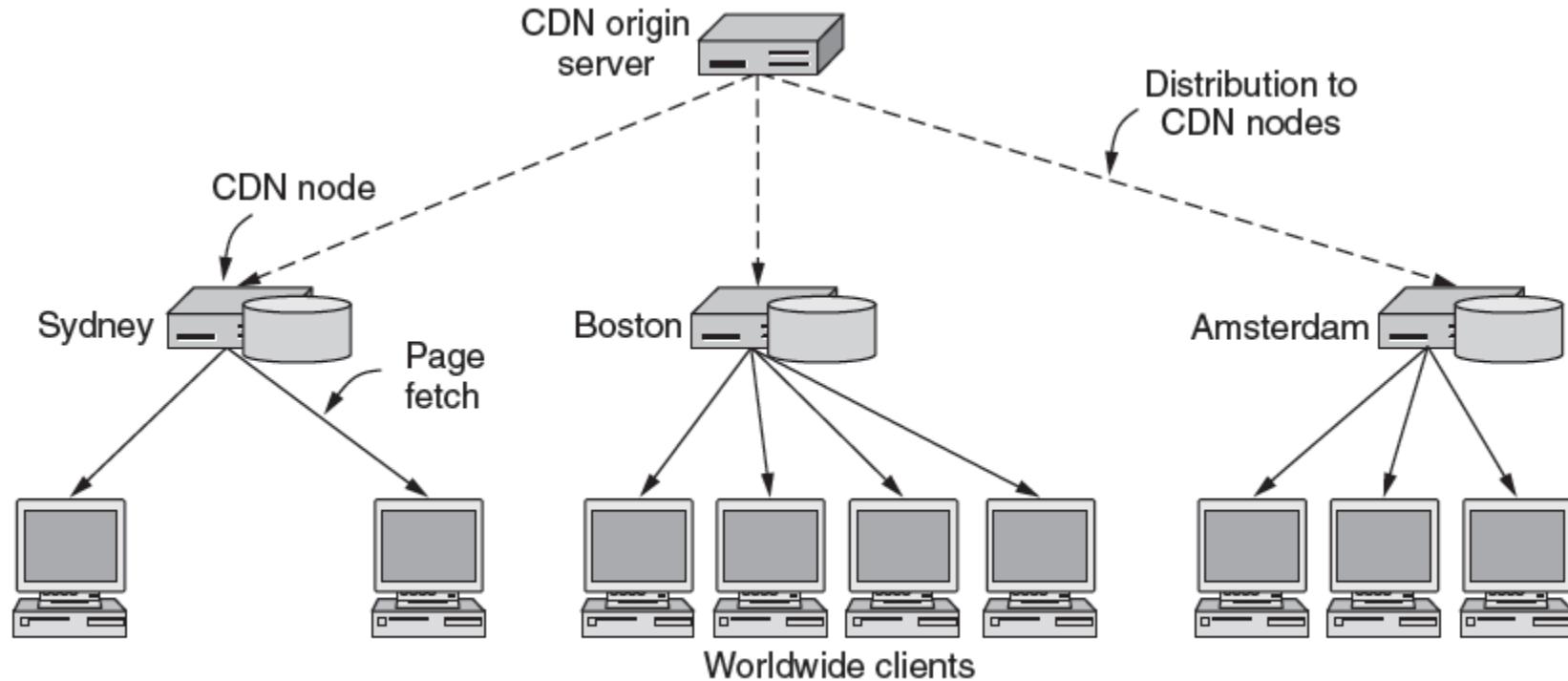
A server farm.

Server Farms and Web Proxies (2)



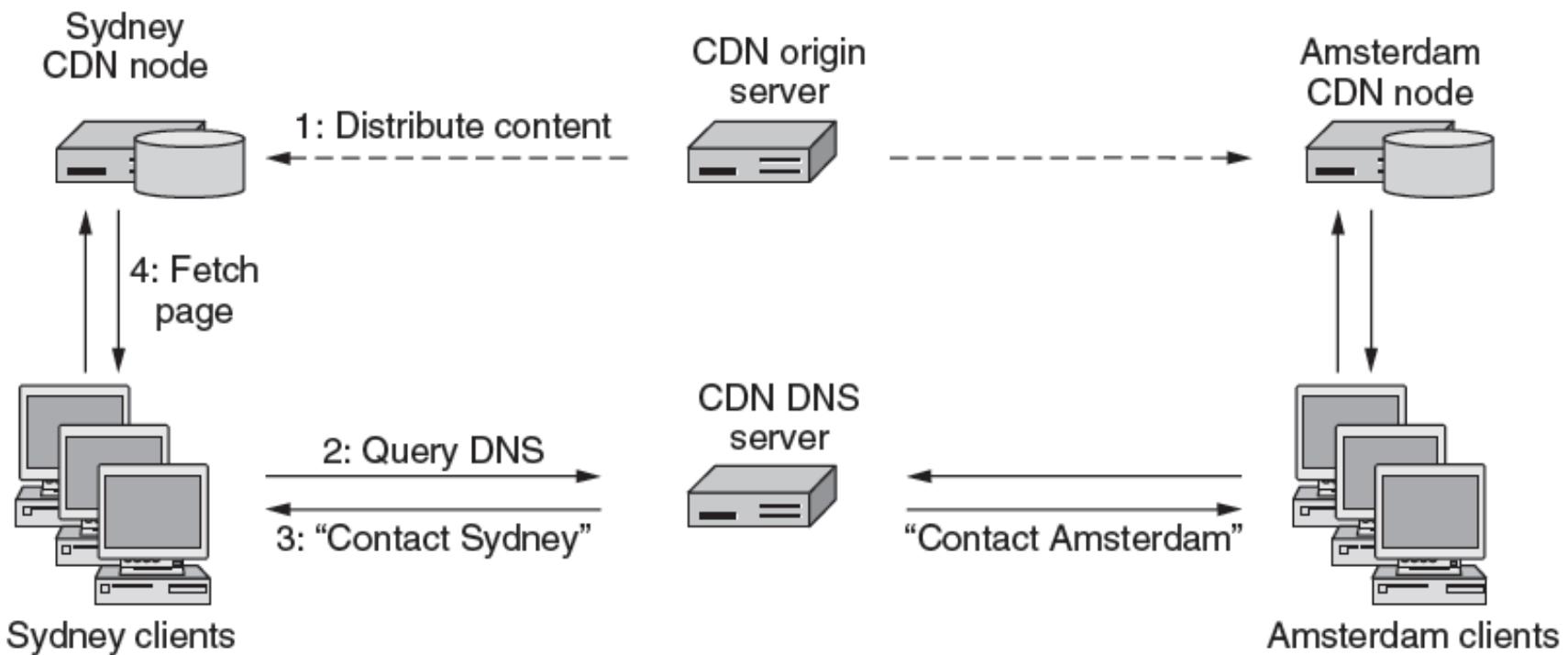
A proxy cache between Web browsers and Web servers.

7.5.3 Content Delivery Networks (1)



CDN distribution tree.

Content Delivery Networks (2)



Directing clients to nearby CDN nodes using DNS.

Content Delivery Networks (3)

```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>
<a href="koalas.mpg"> Koalas Today </a> <br>
<a href="kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

(a)

```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>
<a href="http://www.cdn.com/fluffyvideo/koalas.mpg"> Koalas Today </a> <br>
<a href="http://www.cdn.com/fluffyvideo/kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="http://www.cdn.com/fluffyvideo/wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

(b)

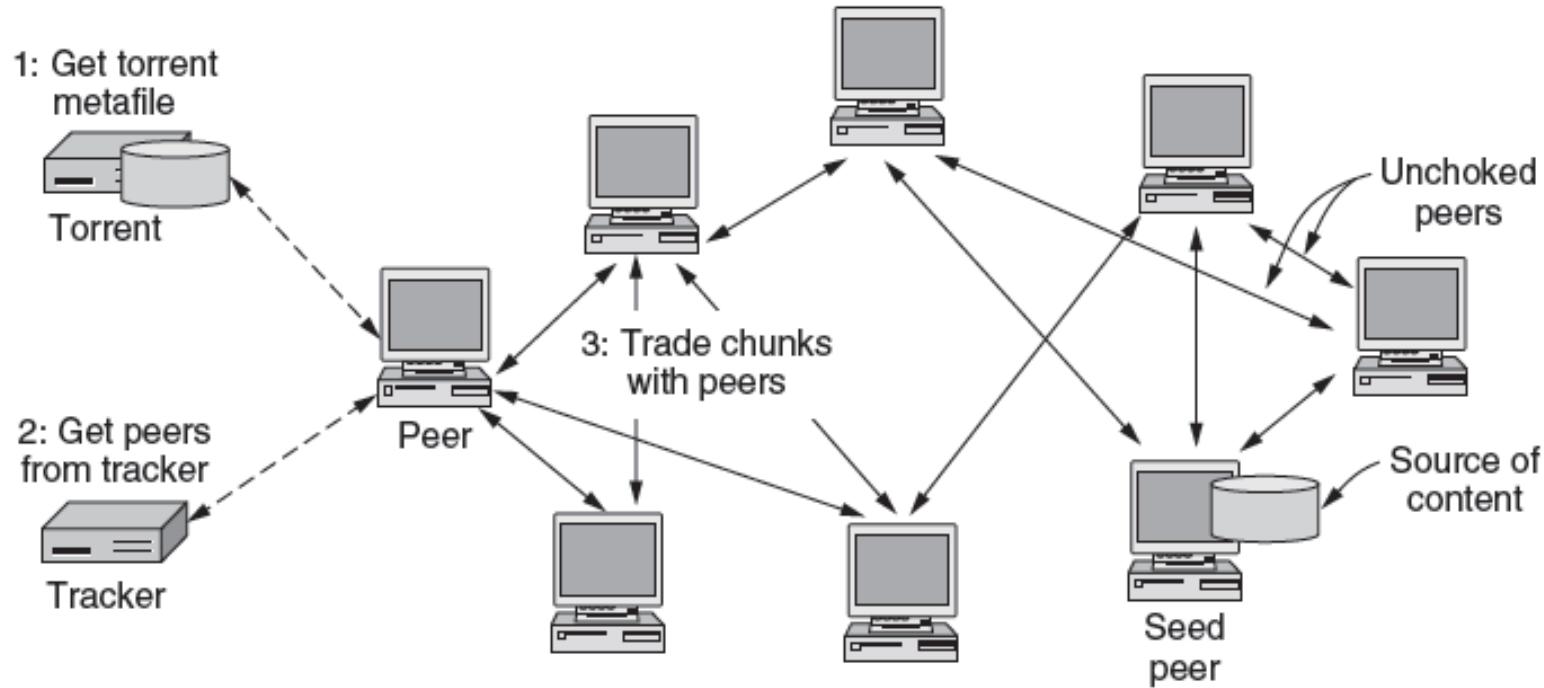
(a) Original Web page. (b) Same page after linking to the CDN

7.5.4 Peer-to-Peer Networks (1)

Problems to be solved with BitTorrent sharing

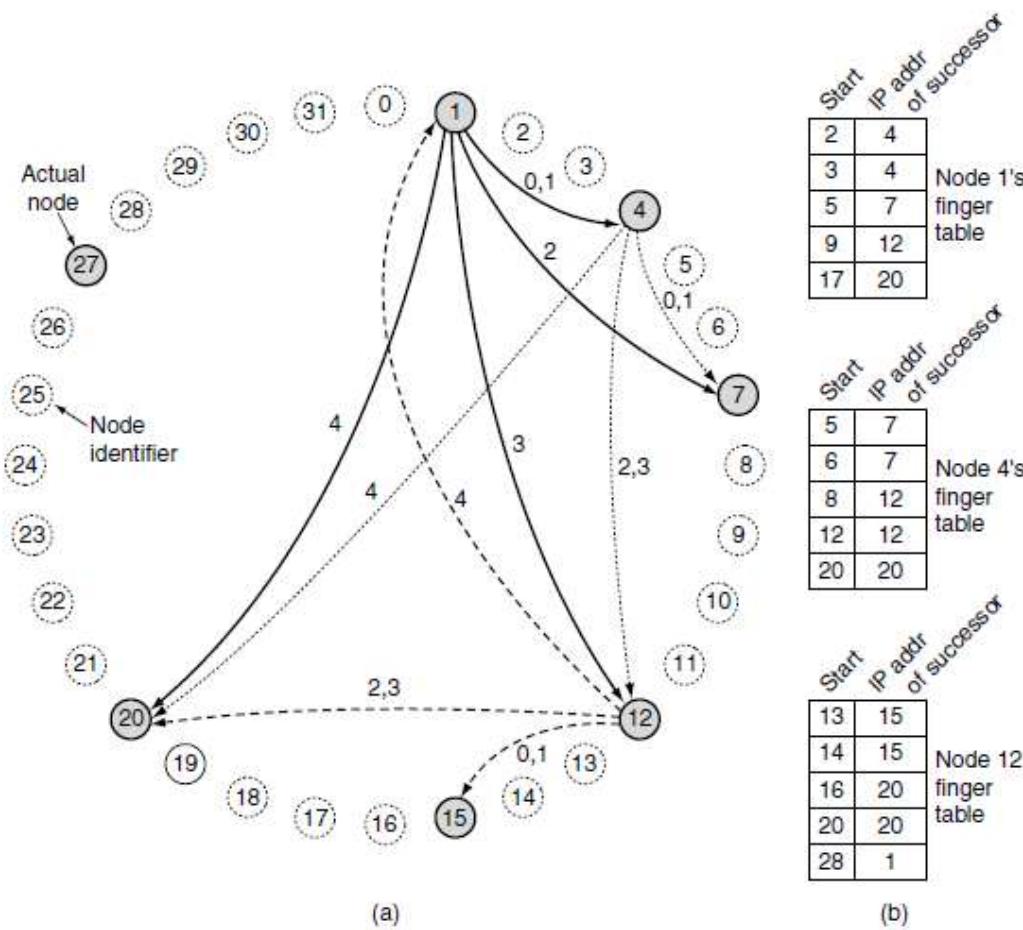
1. How does a peer find other peers
2. How is content replicated by peers to provide high-speed downloads
3. How do peers encourage each other to upload content to others

Peer-to-Peer Networks (2)



BitTorrent.

Peer-to-Peer Networks (3)



(a) A set of 32 node identifiers arranged in a circle. The shaded ones correspond to actual machines. The arcs show the fingers from nodes 1, 4 and 12. The labels on the arcs are the table indices. **(b)** Examples of the finger tables.

Recommended Exercises

In 4th Edition:

- 1, 6, 7, 8, 13, 20,
- 21, 23, 27, 28, 29

In 5th Edition:

- 4, 6, 7, 13, 20, 23, 24, 27-29