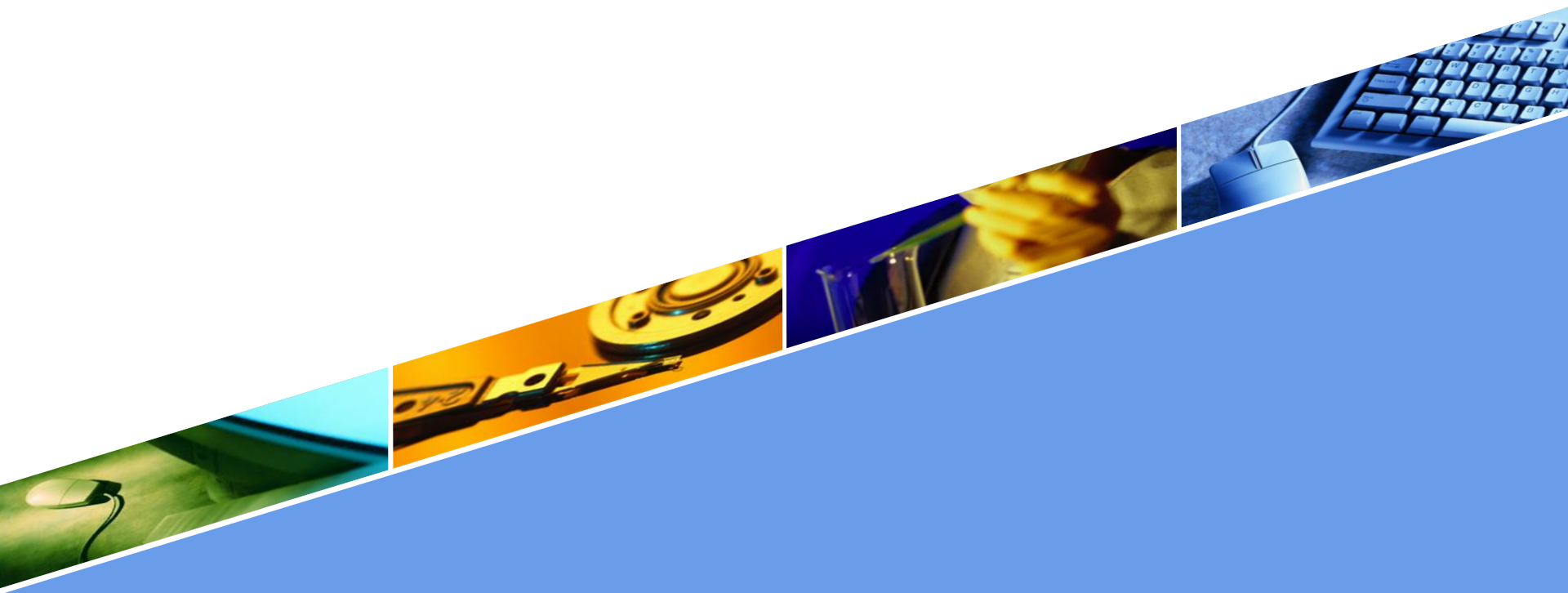
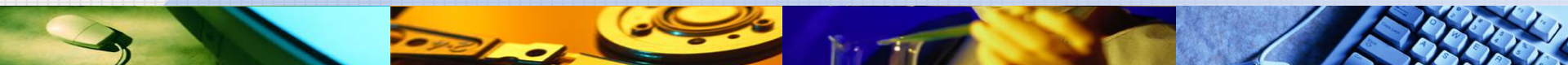


# 应用层概览



# 主题一



1 协议层次回顾

2 应用层简介

3 域名系统DNS

4 电子邮件Email

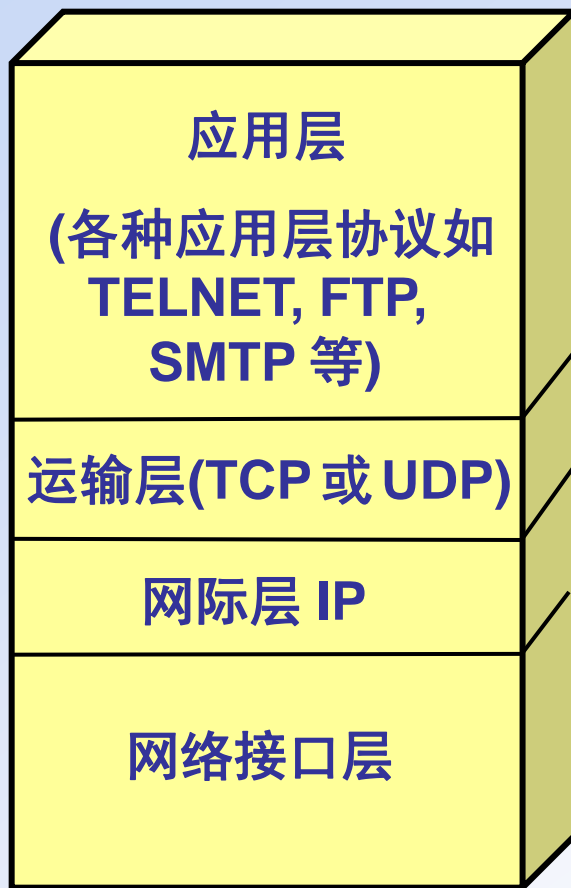
5 万维网WWW

# 网络协议：分层体系结构

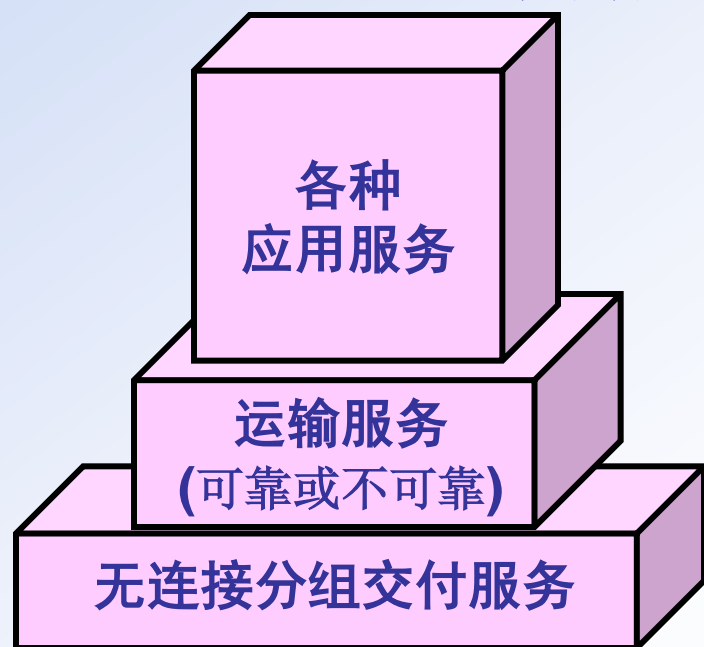
OSI 的体系结构



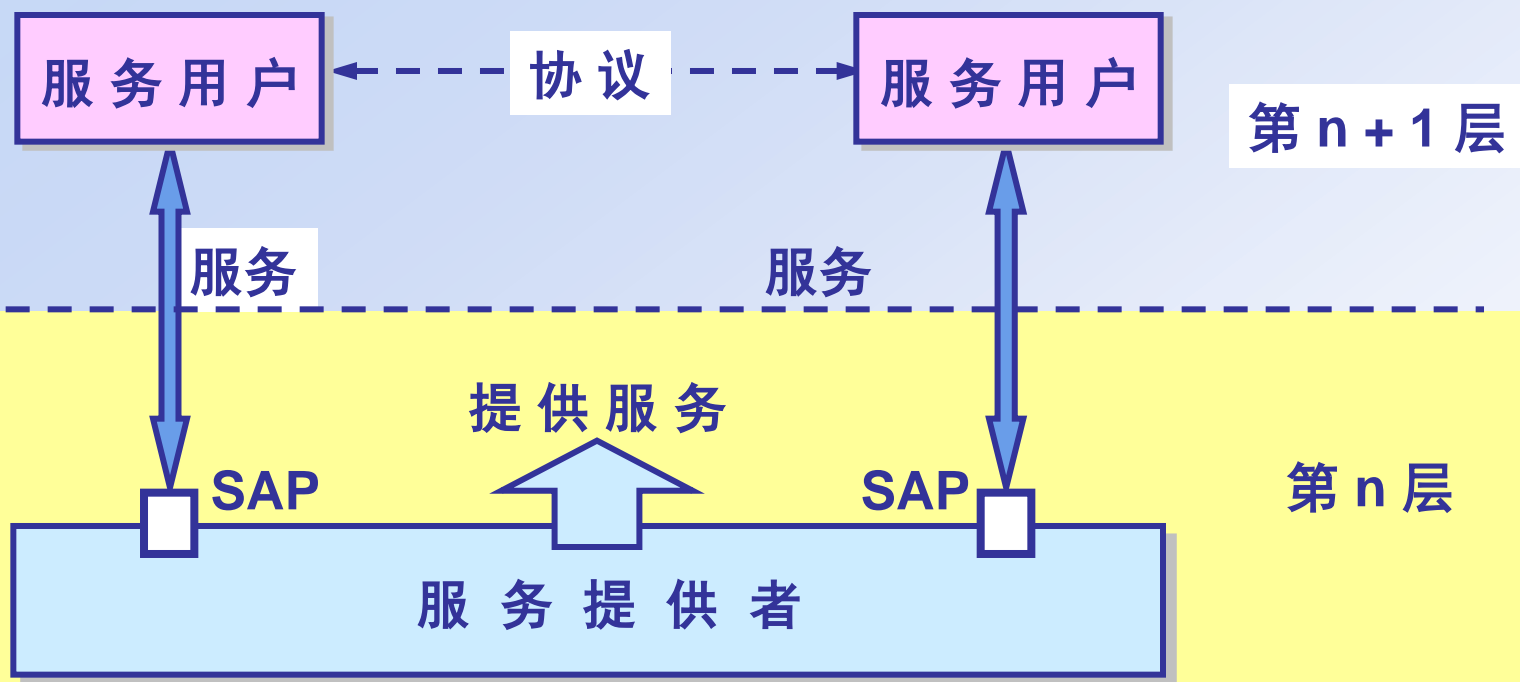
TCP/IP 的体系结构



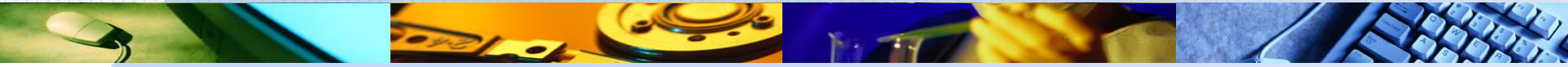
TCP/IP 的三个服务层次



# 模型：要素—实体、协议和服务



## 主题二



1 协议层次回顾

2 应用层简介

3 域名系统DNS

4 电子邮件Email

5 万维网WWW



# 重要性：互联网上每分钟发生什么

## 大数据场景的经典引用：离不开互联网应用 - 应用层协议



- ① 谷歌的搜索引擎被访问694445次
- ② Flickr上上传了6600多张照片
- ③ YouTube上上传了600多个视频
- ④ Facebook上有695000条状态更新
- ⑤ 70个新的域名会被注册
- ⑥ 168000000多条邮件会被发送
- ⑦ .....

# 一些有趣的数据



## ❖ 移动或智能终端、互联网设备

- **2013人均7个，2020人均70**

## ❖ 数据

- **目前：企业7EB，个人6EB**

## ❖ 技术和GDP

- **3G使用增加1倍，GDP增加1.4%**

## ❖ 大数据价值

- **管理方面：价值2500亿美元**
- **制造业：可降低50%成本**
- **是下一个创新、竞争和生产力前沿，是生产资料**

## ❖ IT发展趋势

- **Garner、麦肯锡、IBM十大战略等**
- **BigImpact：大数据与发展倡议 - 发展、挑战与机遇**

# 应用层协议的功用



## ■ 应用层协议

- 目标：每个应用层协议都是为了解决某一类应用问题
- 实施：问题的解决又往往是通过位于不同主机中的多个应用进程之间的通信和协同工作来完成的
- 应用层的具体内容就是规定应用进程在通信时所遵循的协议。

## ■ 体系架构

- 应用层的许多协议都是基于客户服务器方式



# 应用层协议基本概念

应用层是体系结构中的最高层。应用层确定进程之间通信的性质以满足用户的需要（这反映在用户所产生的服务请求）。

支持应用层协议有HTTP、SMTP和FTP等。

特点：面向人的需求

## 应用层

- ① 不是应用软件，不是应用进程，不解决用户的业务需求，为业务需求的实现提供通信
- ② 是应用进程实现网络通信需要实现的协议内容
- ③ 是面向应用需求的通信协议

## 小结

- ① 为实际业务应用提供通用的、标准化的通信平台，以沟通运输层及以下协议——对一类应用的通信“解释”，以便于通信“数据”能被顺利传输
- ② 人类的需求类型实际是相对有限的，所以应用层协议并不多

说说人之间的沟通方式

# 为什么是那些协议

## ❖ 从人的需求探究——沟通的需求内容和方式

### ■ 使用者角度

- 最基本：数据传输
- 大单元信息，文件往来：文件传输—FTP
- 周到的“信件”：邮件传输—SMTP、POP3
- “报刊杂志”：网页浏览—HTTP
- “视觉享受”：视频传输
- “聊天交心”：对等沟通—QQ等
- 快速传送：迅雷、BT等—在WWW资源的基础上
- 快速查询：Google、Baidu等搜索引擎—在WWW资源的基础上

### ■ 维护管理者角度

- 远程工作：远程登录维护—Telnet
- 网络管理：SNMP
- 地址翻译：DNS等

# 协议实现的主流方式



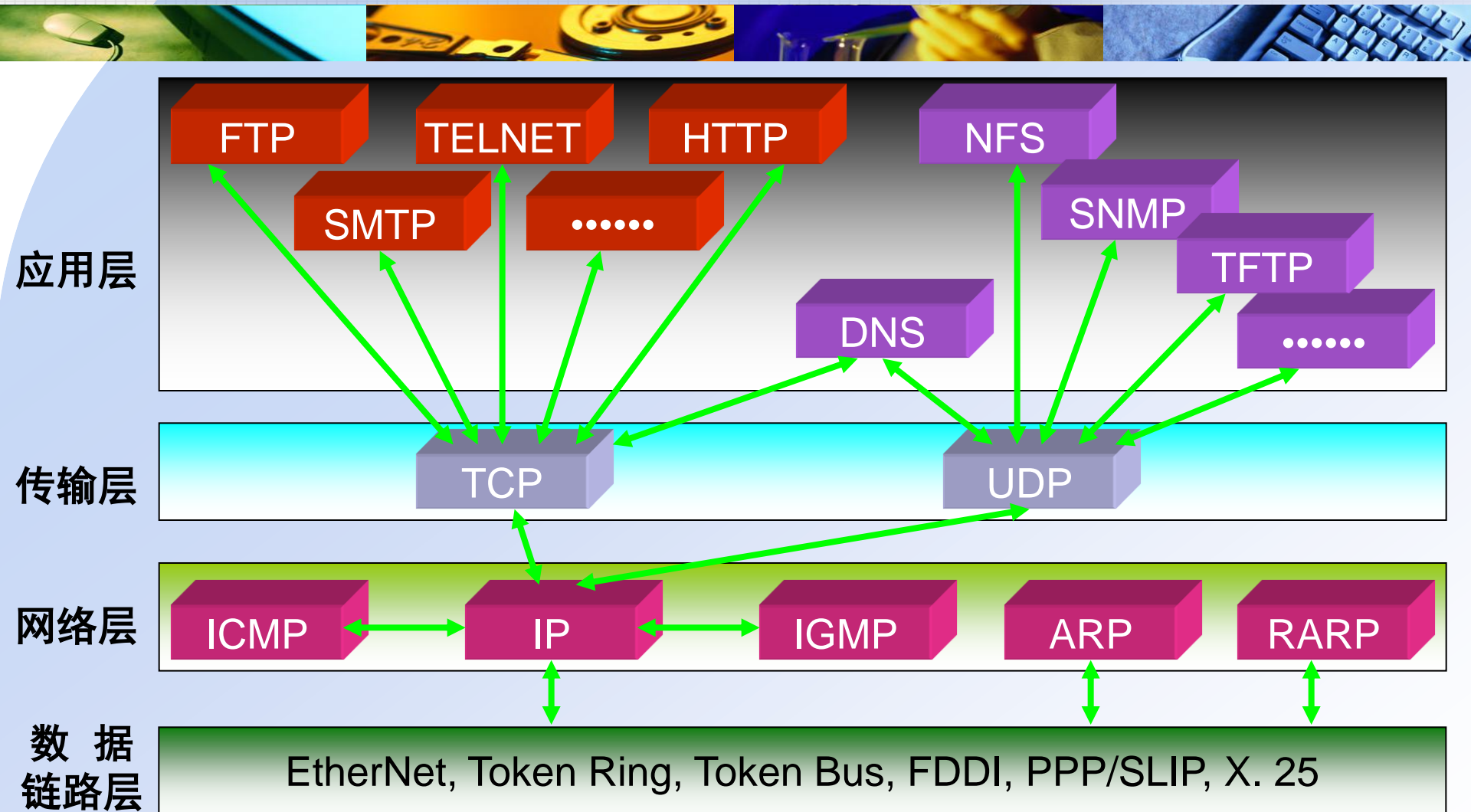
## ❖ 客户—服务器方式

- **客户：提出需求，完成期望的行为**
- **服务器：管理和调度资源，满足客户对资源利用的需求**
- **交互：在协议的约束下，避免客户申请和服务器调度间的冲突**

## ❖ P2P方式

- **对称方式**
- **对等实体**
- **生存能力大大提高**
- **不仅“连接”是网络，而且资源分布和通信也是“网络”**

# 协议层次



# 上下层实体

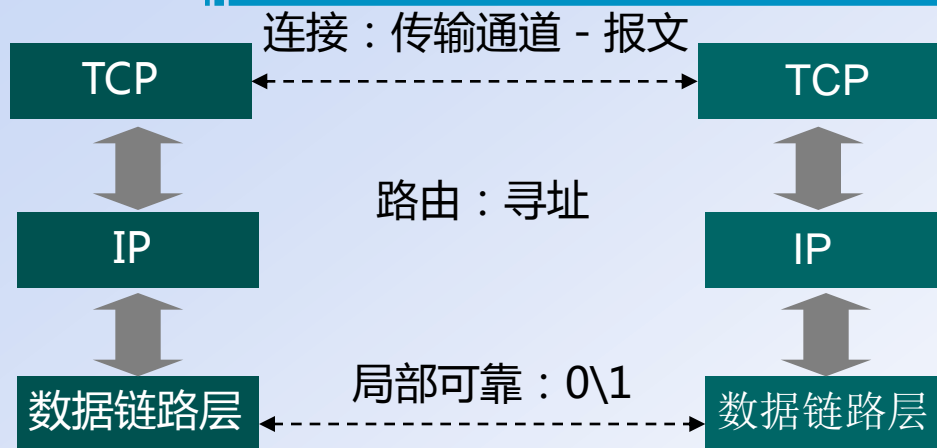
## 用户需求

### 应用程序

用户需求不一：

- ✓ 通信内容：文字、文件、
- ✓ 视频、音频
- ✓ 实时性：实时、非实时
- ✓ 方式：即传即看、接收后打开
- ✓ 同步对等：聊天
- ✓ 共享人数：一对多、多对多等
- ✓ 操作方式：简单、复杂
- ✓ 人的行为习惯：邮件、报刊

### 人看得懂吗？



## 应用层协议

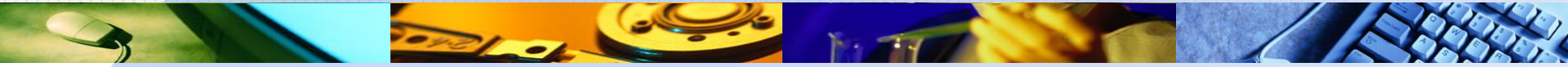
- ✓ 域名系统、HTTP协议、邮件协议.....

## 端口

- ✓ 指向应用进程的通道



# 主题三



1 协议层次回顾

2 应用层简介

3 域名系统DNS

4 电子邮件Email

5 万维网WWW

# 域名系统 DNS



- ❖ DNS概述
- ❖ 域名结构
- ❖ 域名服务器
- ❖ 域名解析过程

# 背景



## ❖ IP地址和地址名字

- **IP地址32bit——数字标识**
  - 便于运算——应用和技术底层
- **地址名称——现实世界术语标识**
  - 便于记忆——应用在面向用户层面
- **在用户层面和技术层面需要转换**

## ❖ 两种转换方式

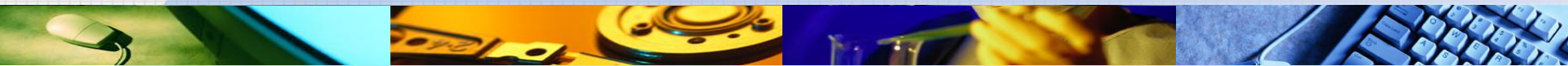
- **当网络规模比较小时，每台主机只需查找指定的文件（UNIX的hosts），就找出主机名字与IP地址的对应关系。**
- **当网络规模很大时，上述方法就出现开销大，效率低，管理难问题，因此研制了域名系统DNS来解决这种问题。**

# 域名系统概述



- ❖ 域名系统 DNS (Domain Name System) 实现名字到 IP 地址的解析。
- ❖ 应用层软件经常直接使用域名系统，但计算机的用户只是间接而不是直接使用域名系统。
- ❖ 因特网采用层次结构的命名树作为主机的名字，并使用分布式的域名系统 DNS。
- ❖ DNS 是一个联机分布式数据库系统，并采用客户/服务器模式。
- ❖ 名字到 IP 地址的解析是由若干个域名服务器程序完成的。域名服务器程序在专设的结点上运行，运行该程序的机器称为域名服务器。

# 域名解析过程简介

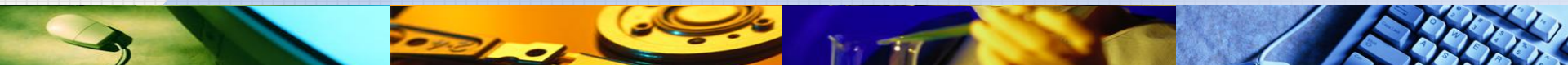


## ❖ 工作过程：

- ❖ 1) 当应用程序需要解析域名时（从符号名到IP地址），它成为域名系统的一个客户。
- ❖ 2) 它以UDP包格式向本地域名服务器发出解析请求，并将待解析的域名放在请求报文中。
- ❖ 3) 本地域名服务器找到域名后，将对应的IP地址放在回答报文中返回，应用进程获得目的主机的IP地址后即可进行通信。
- ❖ 4) 若本地域名服务器无法完成域名解析，它临时变成其上级域名服务器的客户，并向其它域名服务器发出查询请求，递归解析，直到该域名解析完成。



# 解析原理



## ❖ 原理

- **C-S模式**
- **UDP协议**
- **优先本地域名服务器**
- **逐层反映直至获得答案**

## ❖ 问题

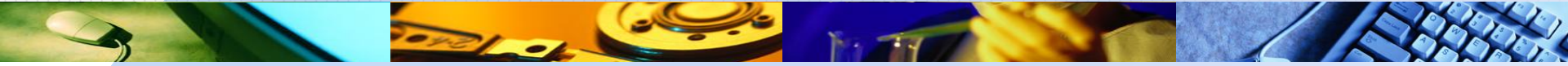
- **UDP是无连接的**
- **可能解析不成功**
- **解析有时间的差异 - 本地和远程**

# 域名系统 DNS



- ❖ DNS概述
- ❖ 域名结构
- ❖ 域名服务器
- ❖ 域名解析过程

# 域名结构

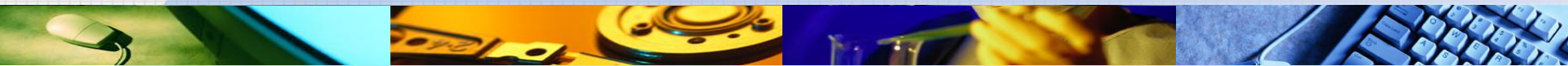


- 因特网采用了层次树状结构的命名方法。
- 任何一个连接在因特网上的主机或路由器，都有一个**唯一**的层次结构的**名字**，即**域名**。
- 域名的结构由若干个分量组成，各分量之间用点隔开：  
... . 三级域名 . 二级域名 . 顶级域名
- 各分量分别代表不同级别的域名。

# 域名只是个逻辑概念

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

# 域名结构—顶级域名TLD (Top Level Domain)



(1) 国家顶级域名：如：.cn 表示中国，.us 表示美国，.uk 表示英国，等等。

(2) 通用顶级域名：

最先确定的7个：

.com 表示公司企业

.net 表示网络服务机构

.org 表示非赢利性组织

.edu 表示教育机构（美国专用）

.gov 表示政府部门（美国专用）

.mil 表示军事部门（美国专用）

.int （国际组织）

陆续增加了13个：

.aero 用于航空运输企业

.biz 用于公司和企业

.coop 用于合作团体

.info 适用于各种情况

.museum 用于博物馆

.name 用于个人

.pro 用于会计、律师等自由职业者

(3) 基础结构域名(infrastructure domain):

arpa, 用于反向域名解析，因此又称为反向域名。



# 因特网的域名空间

树根

顶级域名

... coop info biz aero com net org edu gov mil int cn uk ...

二级域名

cctv

...

ibm

hp

mot

...

hk

js

sh

bj

org

net

gov

edu

com

ac

三级域名

mail

...

tsinghua

pku

fudan

sztu

seu

四级域名

mail

csnet1

ep

...

# 域名系统 DNS



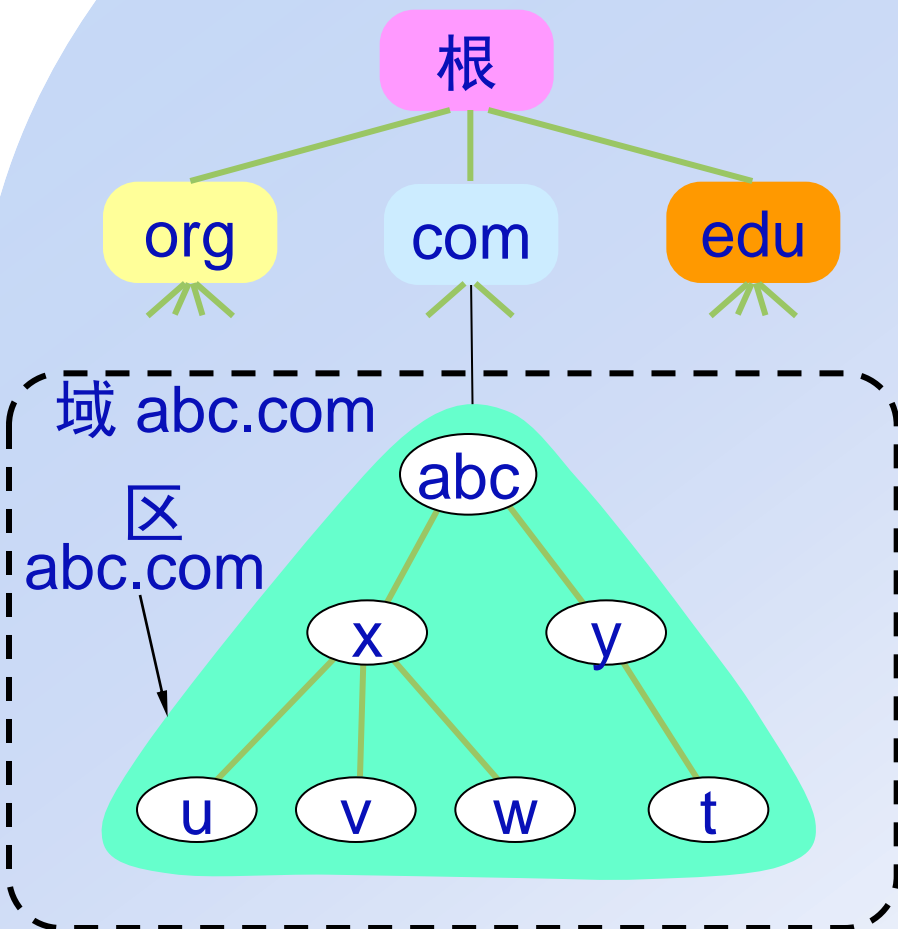
- ❖ DNS概述
- ❖ 域名结构
- ❖ **域名服务器**
- ❖ 域名解析过程

# 权限域名服务器

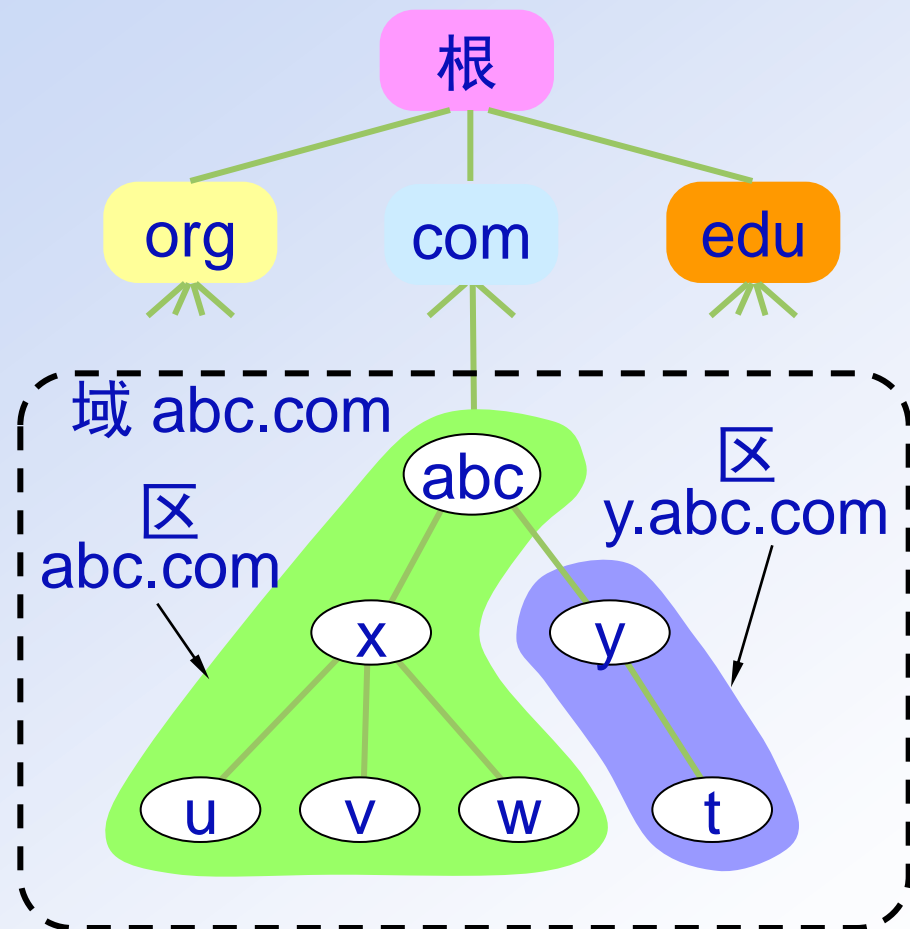


- ❖ 一个服务器所负责管辖的（或有权限的）范围叫做**区(zone)**。
- ❖ 各单位根据具体情况来划分自己管辖范围的区。但在一个区中的所有节点必须是能够连通的。
- ❖ 每一个区设置相应的**权限域名服务器**，用来保存该区中的所有主机的域名到IP地址的映射。
- ❖ DNS 服务器的**管辖范围**不是以“域”为单位，而是以“区”为单位。

# 区的不同划分方法举例

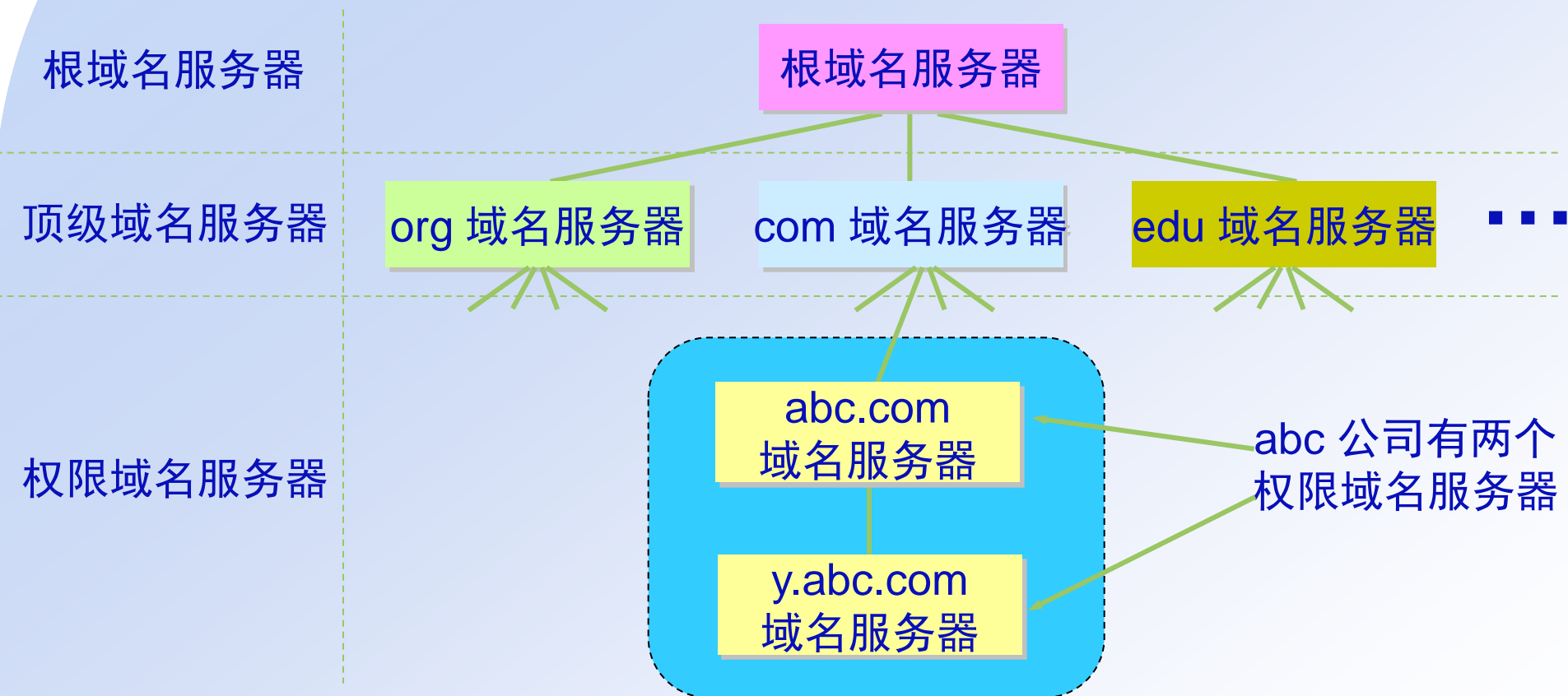
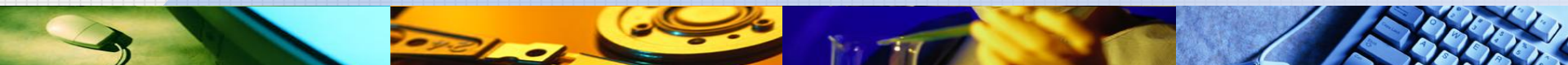


(a) 区 = 域



(b) 区 < 域

# 树状结构的 DNS 域名服务器





# 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

# 根域名服务器—最高层次的域名服务器



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

# 根域名服务器共有 13 套装置（不是 13 个机器）



❖ 这些根域名服务器相应的域名分别是

a. rootservers.net

b. rootservers.net

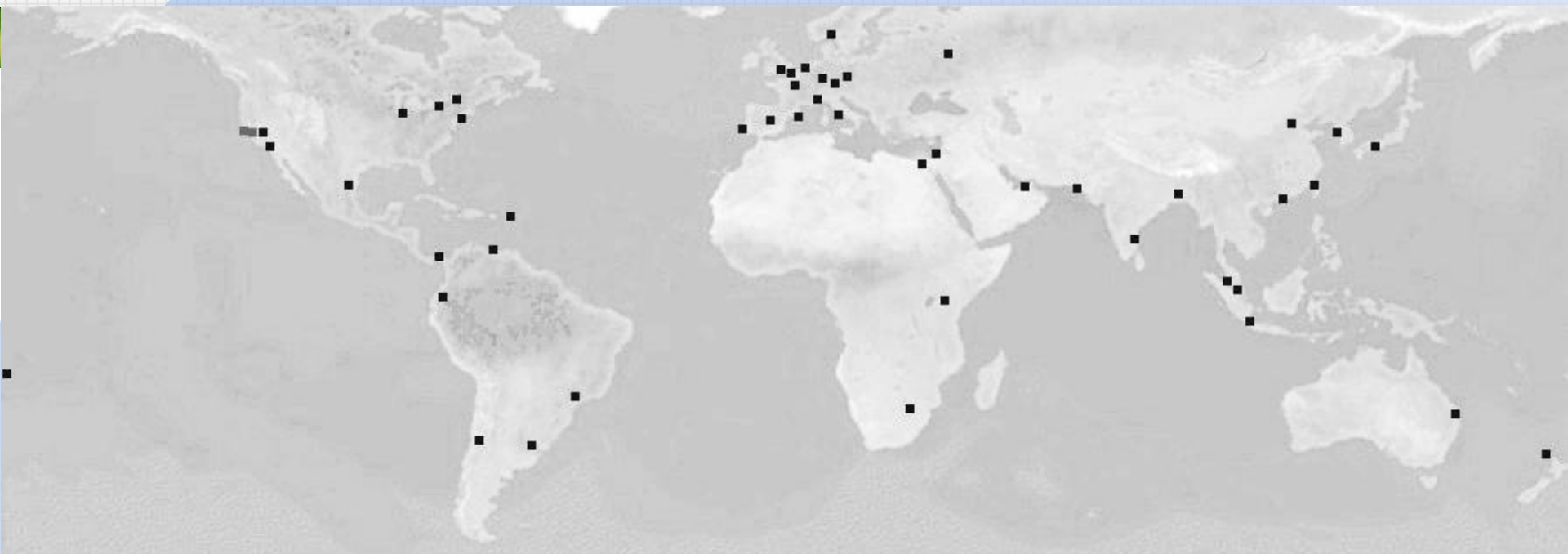
...

m. rootservers.net

❖ 到 2012年全世界已经安装了312个根域名服务器机器，分布在世界各地。

❖ 这样做的目的是为了更方便用户，使世界上大部分 DNS 域名服务器都能**就近**找到一个根域名服务器。

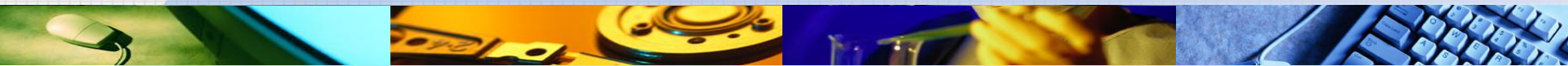
## 举例：根域名服务器 f 的地点分布图（2012年5月）



根域名服务器 f 共有 49 个机器

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

# 顶级域名服务器（即 TLD 服务器）



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



# 权限域名服务器



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



# 本地域名服务器



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

# 提高域名服务器的可靠性



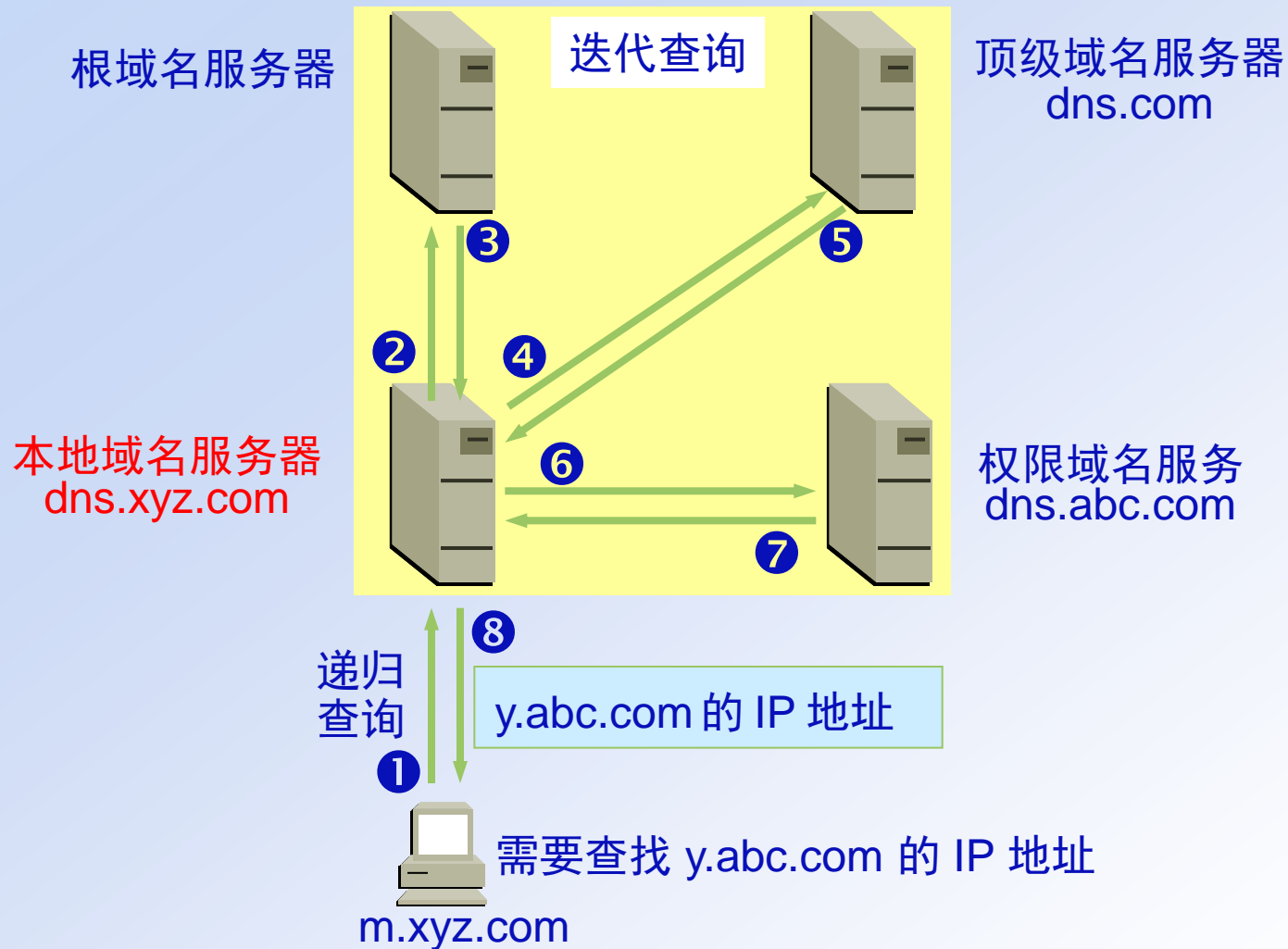
- ❖ DNS 域名服务器都把数据复制到几个域名服务器来保存，其中的一个是**主域名服务器**，其他的是**辅助域名服务器**。
- ❖ 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断。
- ❖ 主域名服务器定期把数据复制到辅助域名服务器中，而**更改数据只能在主域名服务器中进行**。这样就保证了数据的一致性。

# 域名系统 DNS

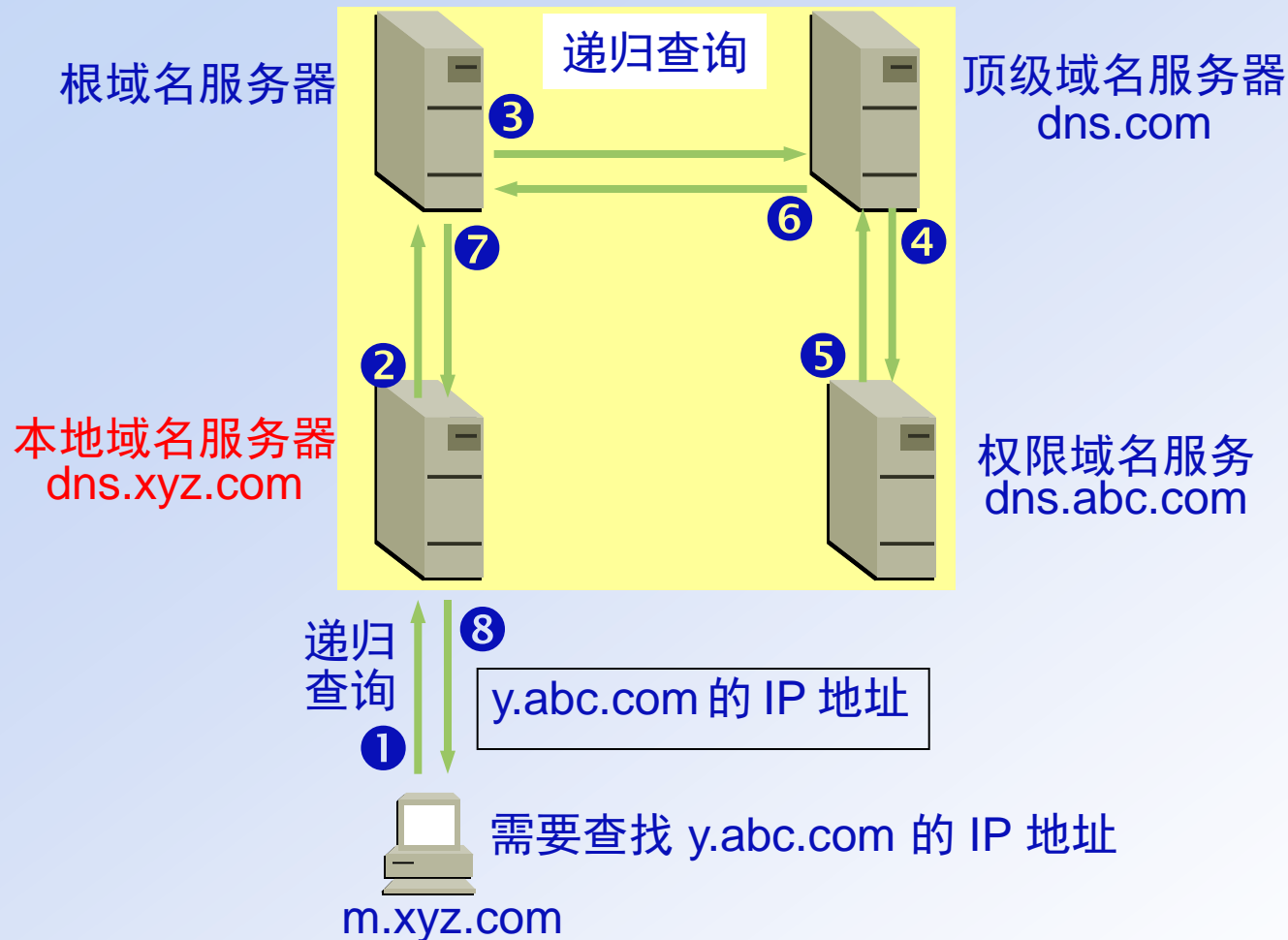


- ❖ DNS概述
- ❖ 域名结构
- ❖ 域名服务器
- ❖ 域名解析过程

# 域名的解析过程—本地域名服务器采用迭代查询

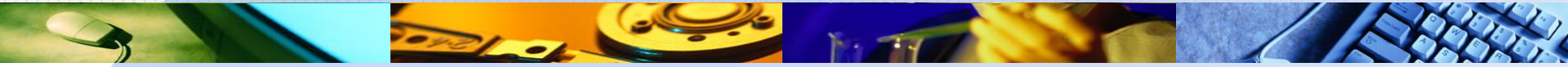


# 域名的解析过程—本地域名服务器采用递归查询 (比较少用)





# 名字的高速缓存



- ❖ 每个域名服务器都维护一个高速缓存，存放最近用过的名字以及从何处获得名字映射信息的记录。
- ❖ 当客户请求域名服务器转换名字时，查询此高速缓存。
- ❖ 可优化查询的开销，大大减轻根域名服务器的负荷，使因特网上的 DNS 查询请求和回答报文的数量大为减少。
- ❖ 为保持高速缓存中的内容正确，域名服务器应为每项内容设置计时器，并处理超过合理时间的项（例如，每个项目只存放两天）。
- ❖ 当权限域名服务器回答一个查询请求时，在响应中都指明绑定有效存在的时间值。



# 主题四



1 协议层次回顾

2 应用层简介

3 域名系统DNS

4 电子邮件Email

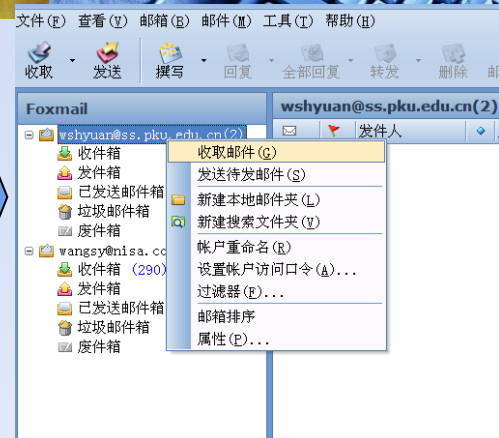
5 万维网WWW

# 电子邮件Email



- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件

# 实际邮件操作示例



看到什么?

✓有客户端程序

✓能够感受到服务  
器程序：有时候邮  
件发送不出去

邮件有格式：

- ✓收发件人
- ✓主题
- ✓正文
- ✓附件
- ✓.....



感受到“连接”

- ✓建立发送“连接”——实  
际是会话
- ✓并非不管不顾的发送
- ✓收邮件亦然

# 人的需求：从通信双方的视角

收件人邮政编码

收件人地址、姓名.....  
.....谁转送给谁.....

寄件人地址、姓名.....

寄件人邮政编码

封装内容：有时候从信封  
颜色、强调大体可知主题

## 解读：规程和格式

收件人

- ① 邮编
- ② 地址
- ③ 姓名

寄件者

- ① 邮编
- ② 地址
- ③ 姓名

内容提示

- ① 信封上的文字提示
- ② 信封颜色

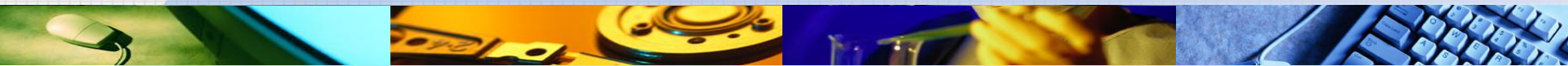
中间转送或多人传阅

- ① 转送
- ② 多人阅读

格式：

- ① 文字清晰
- ② 收件人和寄件人的部署位置
- ③ 邮政编码符合规定

# 电子邮件概述



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

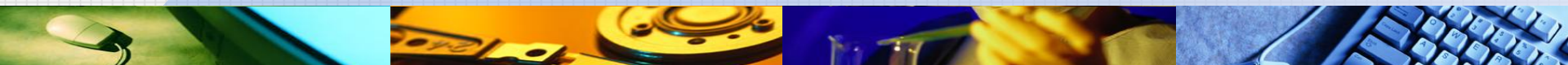


# 电子邮件的一些标准

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

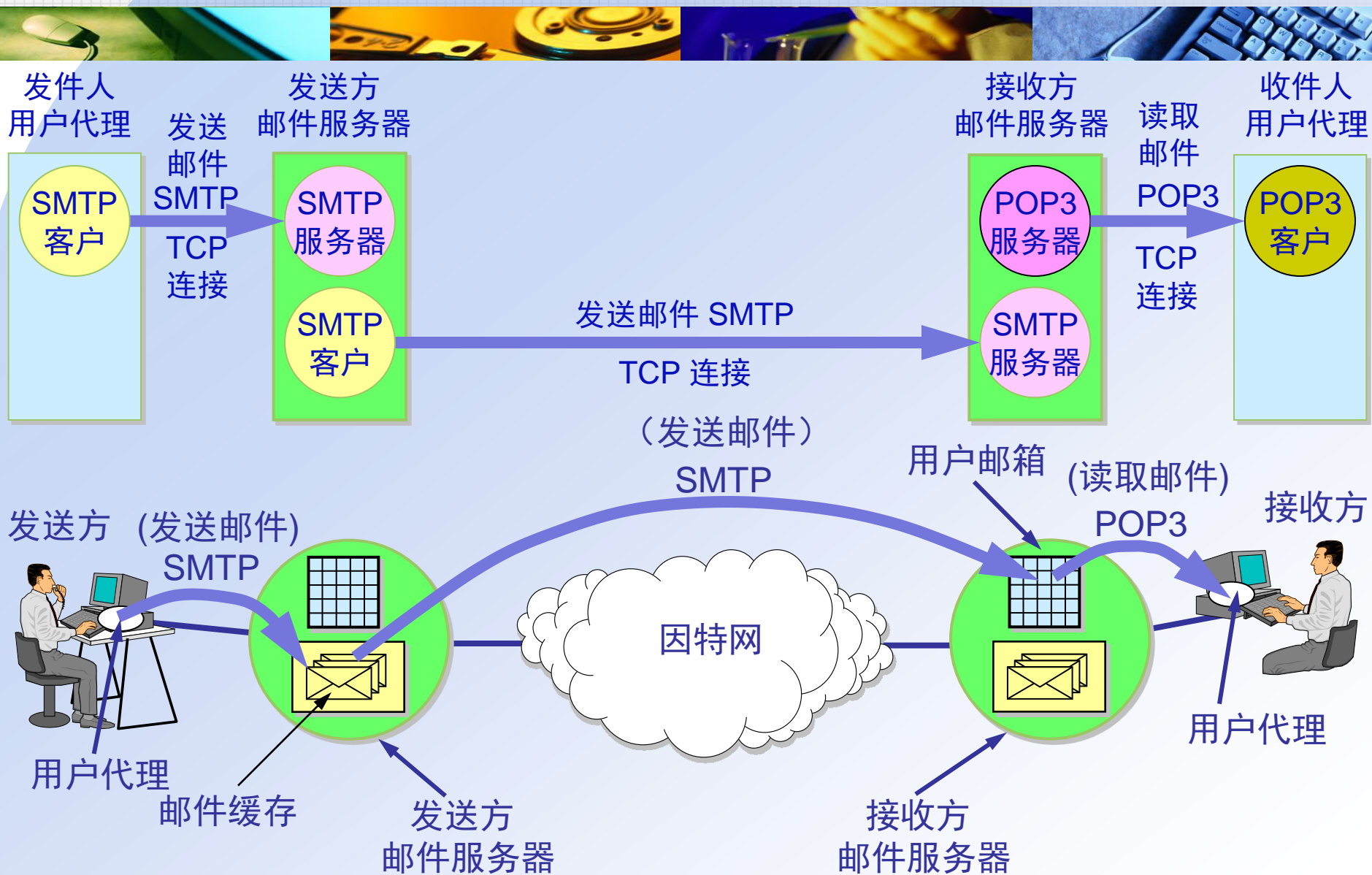


# 电子邮件Email

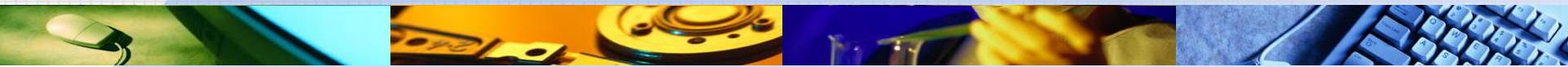


- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件

# 电子邮件的最主要的组成构件

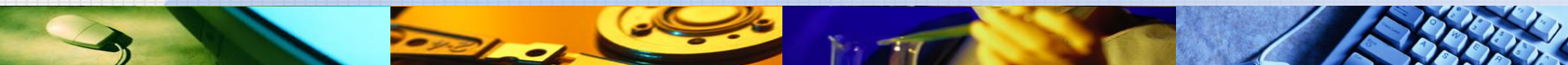


# 用户代理 UA (User Agent) 和邮件服务器



- ❖ **用户代理 UA** 就是用户与电子邮件系统的接口，是电子邮件客户端软件。
  - **用户代理的功能：撰写、显示、处理和通信等。**
  - **用户代理的例子：Outlook，Foxmail等。**
- ❖ **邮件服务器**的功能是发送和接收邮件，同时还要向发信人报告邮件传送的情况（已交付、被拒绝、丢失等）。
  - **邮件服务器按照客户-服务器方式工作。**
  - **邮件服务器需要使用发送和读取两个不同的协议。**
  - **一个邮件服务器既可以作为客户，也可以作为服务器。**

# 发送和接收电子邮件的几个重要步骤



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

## 发送和接收电子邮件的几个重要步骤（续）



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



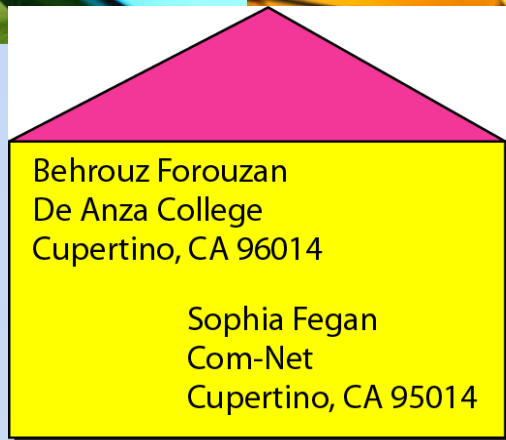
# 电子邮件Email



- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件



## *Format of an e-mail*



Behrouz Forouzan  
De Anza College  
Cupertino, CA 96014

Sophia Fegan  
Com-Net  
Cupertino, CA 95014

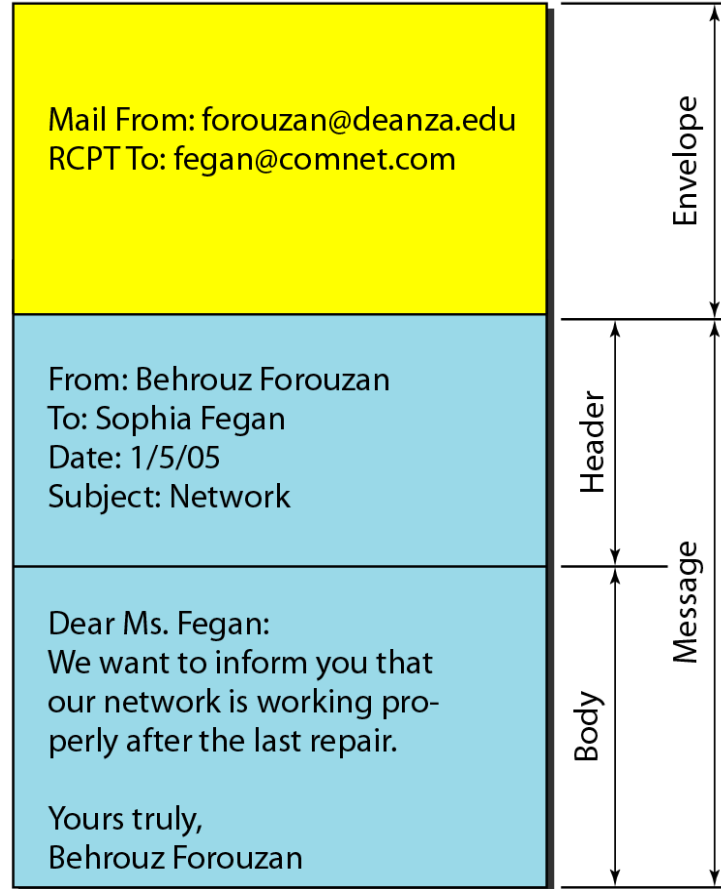
Sophia Fegan  
Com-Net  
Cupertino, CA 95014  
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:  
We want to inform you that  
our network is working properly  
after the last repair.

Yours truly,  
Behrouz Forouzan

a. Postal mail



Mail From: forouzan@deanza.edu  
RCPT To: fegan@comnet.com

From: Behrouz Forouzan  
To: Sophia Fegan  
Date: 1/5/05  
Subject: Network

Dear Ms. Fegan:  
We want to inform you that  
our network is working properly  
after the last repair.

Yours truly,  
Behrouz Forouzan

Envelope

Header

Body

Message

b. Electronic mail

# 电子邮件的信息格式



- ❖ 一个电子邮件分为信封和内容两大部分。
- ❖ 电子邮件的传输程序根据邮件信封上的信息来传送邮件。用户在从自己的邮箱中读取邮件时才能见到邮件的内容。
- ❖ RFC 822 只规定了邮件内容中的首部(header)格式，而对邮件的主体(body)部分则让用户自由撰写。
- ❖ 在邮件的信封上，最重要的就是收件人的地址。
- ❖ 用户写好首部后，邮件系统将自动地将信封所需的信息提取出来并写在信封上。所以用户不需要填写电子邮件信封上的信息。
- ❖ 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：To 和 Subject。

# 邮件内容的首部基本内容



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

# 电子邮件地址的格式

- ❖ TCP/IP 体系的电子邮件系统规定电子邮件地址的格式如下：

收件人邮箱名@邮箱所在主机的域名

- ❖ 符号“@”读作“at”，表示“在”的意思。
- ❖ 例如，电子邮件地址 xiexiren@tsinghua.org.cn

这个用户名在该域名的范围内是唯一的。

邮箱所在的主机的域名在全世界必须是唯一的

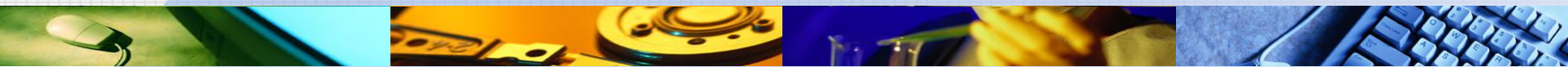
## 通用因特网邮件扩充 MIME的提出

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





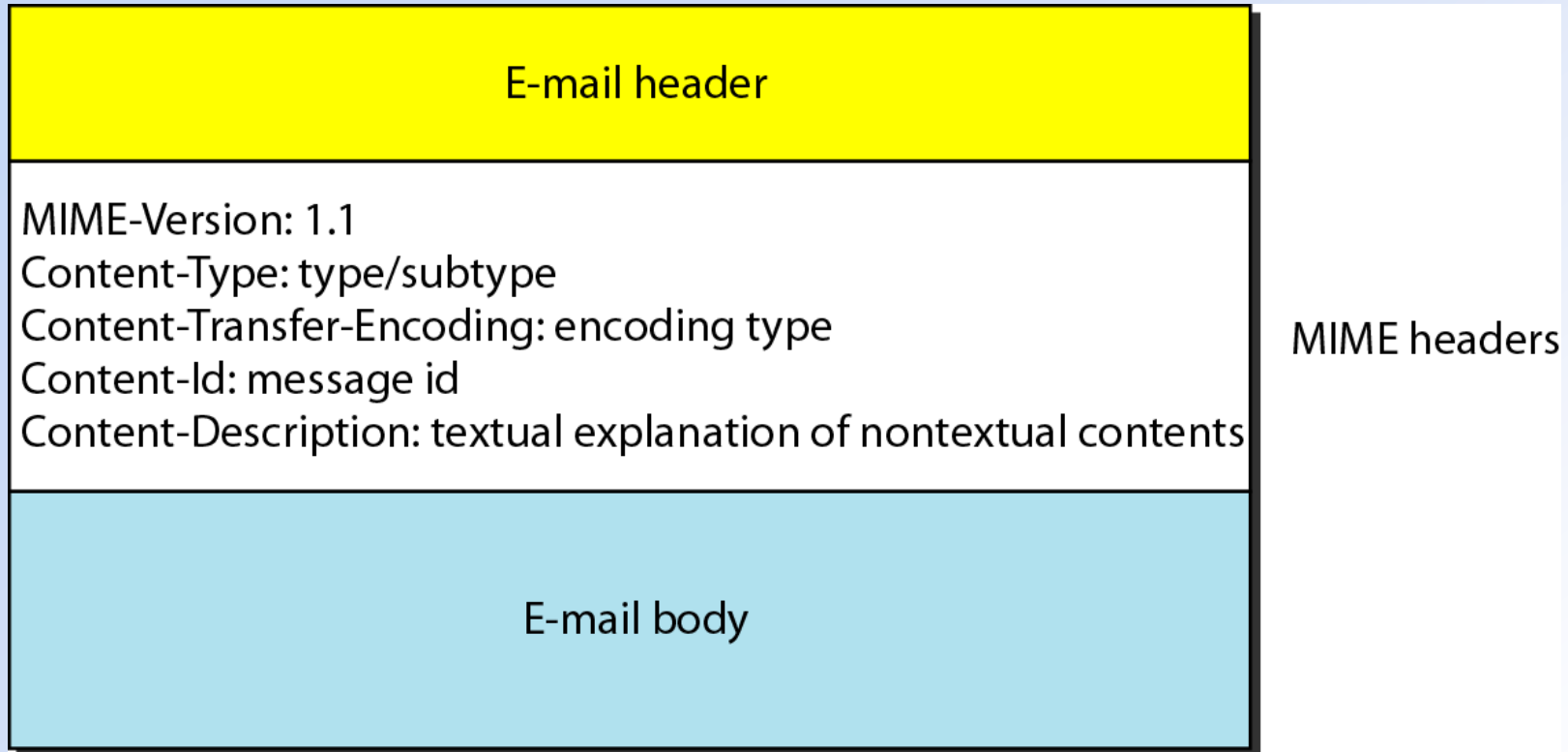
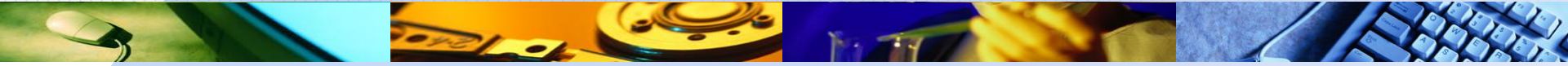
# MIME



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



# MIME 首部



# MIME 增加 5 个新的邮件首部



- ❖ MIME-Version: 标志 MIME 的版本。现在的版本号是 1.0。若无此行，则为英文文本。
- ❖ Content-Type/SubType: 说明邮件的性质。
- ❖ Content-Transfer-Encoding: 在传送时邮件的主体是如何编码的。
- ❖ Content-Id: 邮件的唯一标识符。
- ❖ Content-Description: 这是可读字符串，说明此邮件是什么。和邮件的主题差不多。

# 内容类型和子类型 (Content-Type/SubType)

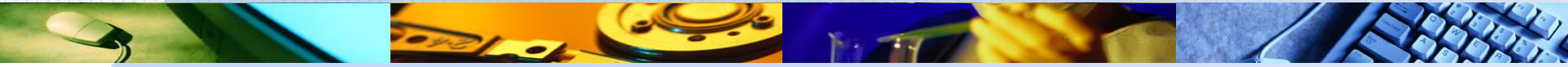


- ❖ MIME 标准规定 Content-Type 说明必须含有两个标识符，即内容类型 (type) 和子类型 (subtype)，中间用 “/” 分开。
- ❖ MIME 标准定义了 7 个基本内容类型和 15 种子类型。

# Content-Type/SubType

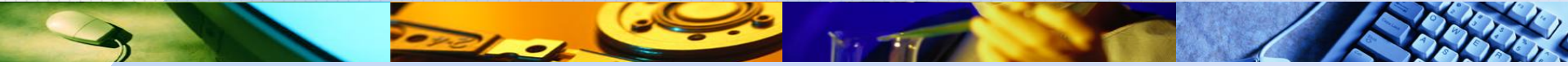
<i>Type</i>	<i>Subtype</i>	<i>Description</i>
Text	Plain	Unformatted
	HTML	HTML format (see Chapter 27)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed subtypes, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single-channel encoding of voice at 8 kHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (8-bit bytes)

# 内容传送编码(Content-Transfer-Encoding)



- ❖ 最简单的编码就是 7 位 ASCII 码，而每行不能超过 1000 个字符。MIME 对这种由 ASCII 码构成的邮件主体不进行任何转换。
- ❖ 另一种编码称为 quoted-printable，这种编码方法适用于当所传送的数据中只有少量的非 ASCII 码。
- ❖ 对于任意的二进制文件，可用 base64 编码。

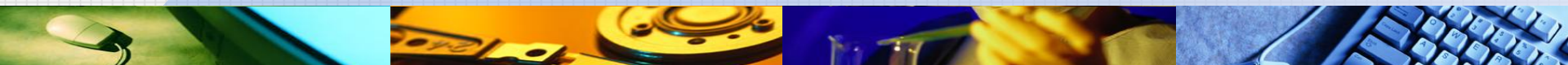
# Content-Transfer-Encoding



<i>Type</i>	<i>Description</i>
7-bit	NVT ASCII characters and short lines
8-bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base-64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

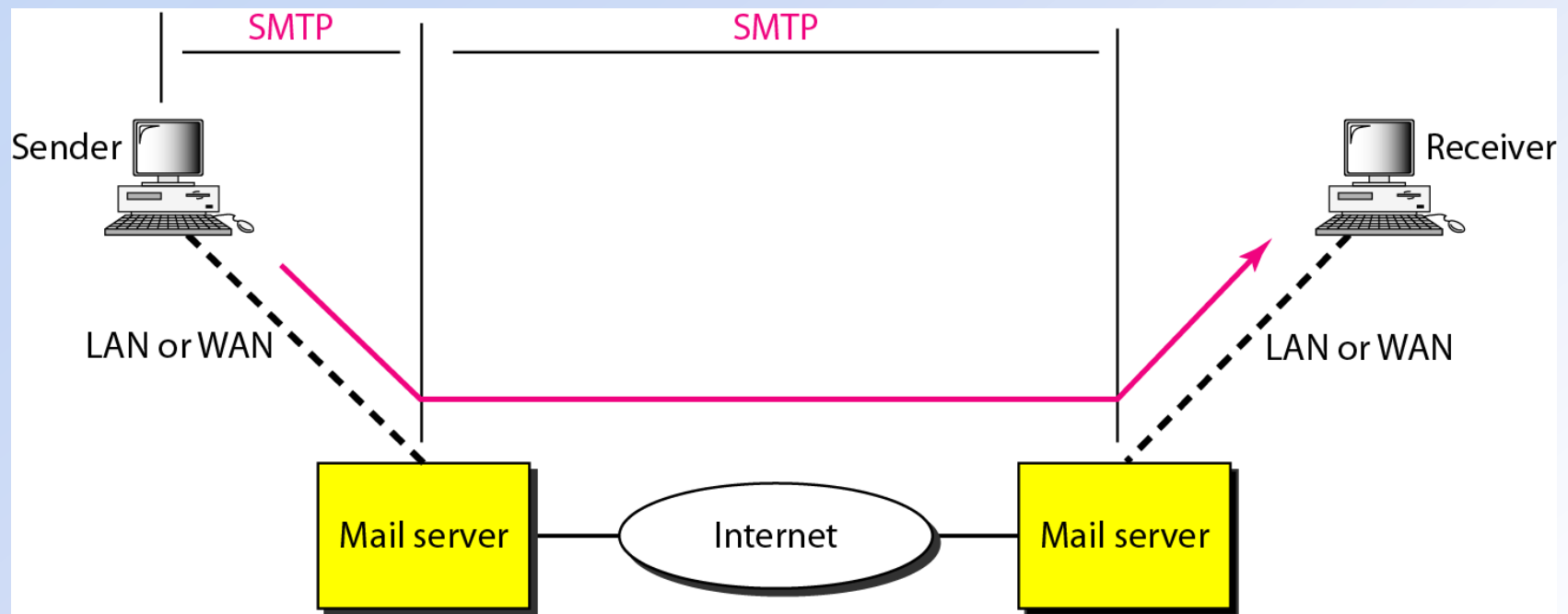


# 电子邮件Email



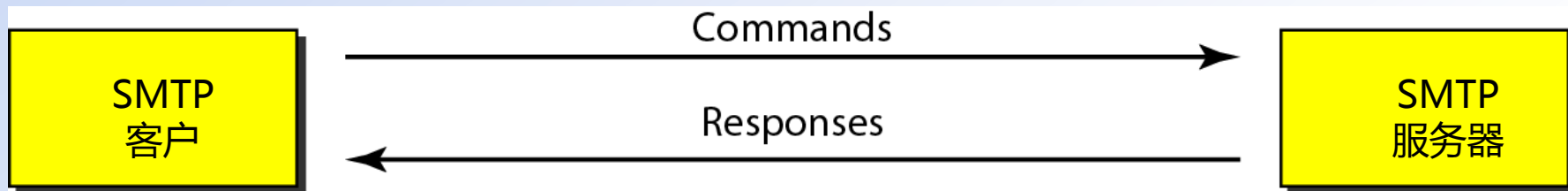
- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件

# 简单邮件传送协议 SMTP 作用范围



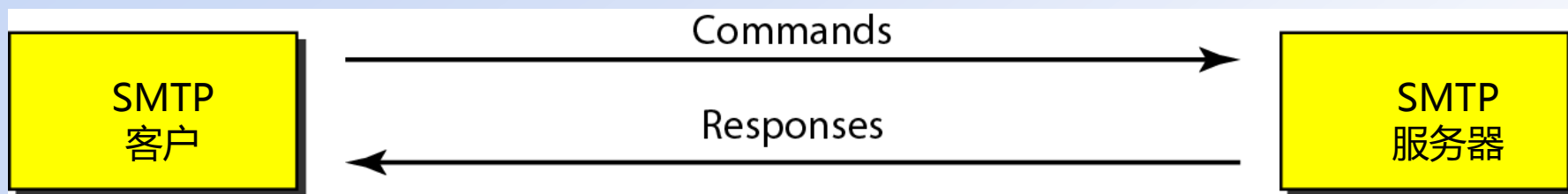
# SMTP 简介

- ❖ SMTP 所规定的就是在两个相互通信的 SMTP 进程之间应如何交换信息。
- ❖ 由于 SMTP 使用客户服务器方式，因此负责发送邮件的 SMTP 进程就是 SMTP 客户，而负责接收邮件的 SMTP 进程就是 SMTP 服务器。
- ❖ SMTP 规定了 14 条命令和 21 种应答信息。
- ❖ SMTP 的命令和响应是在 TCP 之上传输的。SMTP 的 TCP 服务 端口号是 25（Well-known）。



# SMTP协议命令和应答

❖ SMTP 规定了 14 条命令和 21 种应答信息。



# SMTP 命令

**Keyword:** argument(s)

<i>Keyword</i>	<i>Argument(s)</i>
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

# SMTP 应答

由一个 3 位数字的代码开始，后面附上（也可不附上）很简单的文字说明

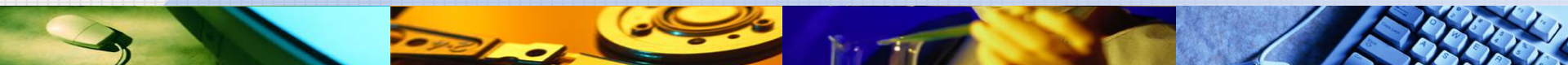
<i>Code</i>	<i>Description</i>
<b>Positive Completion Reply</b>	
<b>211</b>	System status or help reply
<b>214</b>	Help message
<b>220</b>	Service ready
<b>221</b>	Service closing transmission channel
<b>250</b>	Request command completed
<b>251</b>	User not local; the message will be forwarded
<b>Positive Intermediate Reply</b>	
<b>354</b>	Start mail input
<b>Transient Negative Completion Reply</b>	
<b>421</b>	Service not available
<b>450</b>	Mailbox not available
<b>451</b>	Command aborted: local error
<b>452</b>	Command aborted: insufficient storage



# SMTP 应答 (续)

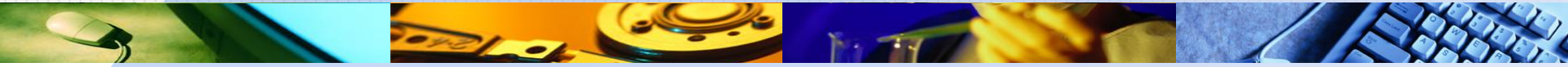
<i>Code</i>	<i>Description</i>
<b>Permanent Negative Completion Reply</b>	
<b>500</b>	Syntax error; unrecognized command
<b>501</b>	Syntax error in parameters or arguments
<b>502</b>	Command not implemented
<b>503</b>	Bad sequence of commands
<b>504</b>	Command temporarily not implemented
<b>550</b>	Command is not executed; mailbox unavailable
<b>551</b>	User not local
<b>552</b>	Requested action aborted; exceeded storage location
<b>553</b>	Requested action not taken; mailbox name not allowed
<b>554</b>	Transaction failed

# SMTP 通信的三个阶段



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

# SMTP 发送邮件---通信过程示例



使用TELNET 登陆到SMTP服务器25 端口，直接使用命令发送一封电子邮件。

本示例中, forouzanb@adelphia.net 发送邮件给他自己.



*TELNET trying to connect to the  
Adelphia mail server*

*\$ telnet mail.adelphia.net 25  
Trying 68.168.78.100 . . .  
Connected to mail.adelphia.net (68.168.78.100).*

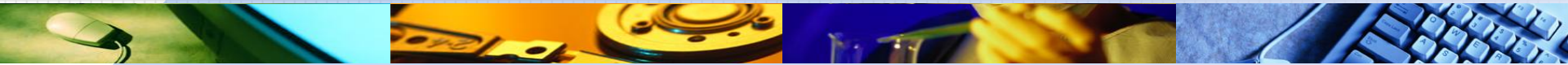
*After connection, we can type the SMTP  
commands and then receive the responses*

Comments

```
===== Connection Establishment =====  
220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .  
HELO mail.adelphia.net  
250 mta13.adelphia.net
```

Command

Response



===== Mail Transfer =====

**MAIL FROM: forouzanb@adelphia.net**

**250 Sender <forouzanb@adelphia.net> Ok**

**RCPT TO: forouzanb@adelphia.net**

**250 Recipient <forouzanb@adelphia.net> Ok**

**DATA**

**354 Ok Send data ending with <CRLF>.<CRLF>**

**From: Forouzan**

**TO: Forouzan**

**This is a test message  
to show SMTP in action.**

**•**

===== Connection Termination =====

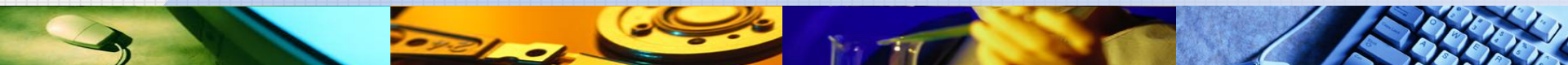
**250 Message received: adelphia.net@mail.adelphia.net**

**QUIT**

**221 mta13.adelphia.net SMTP server closing connection**

**Connection closed by foreign host.**

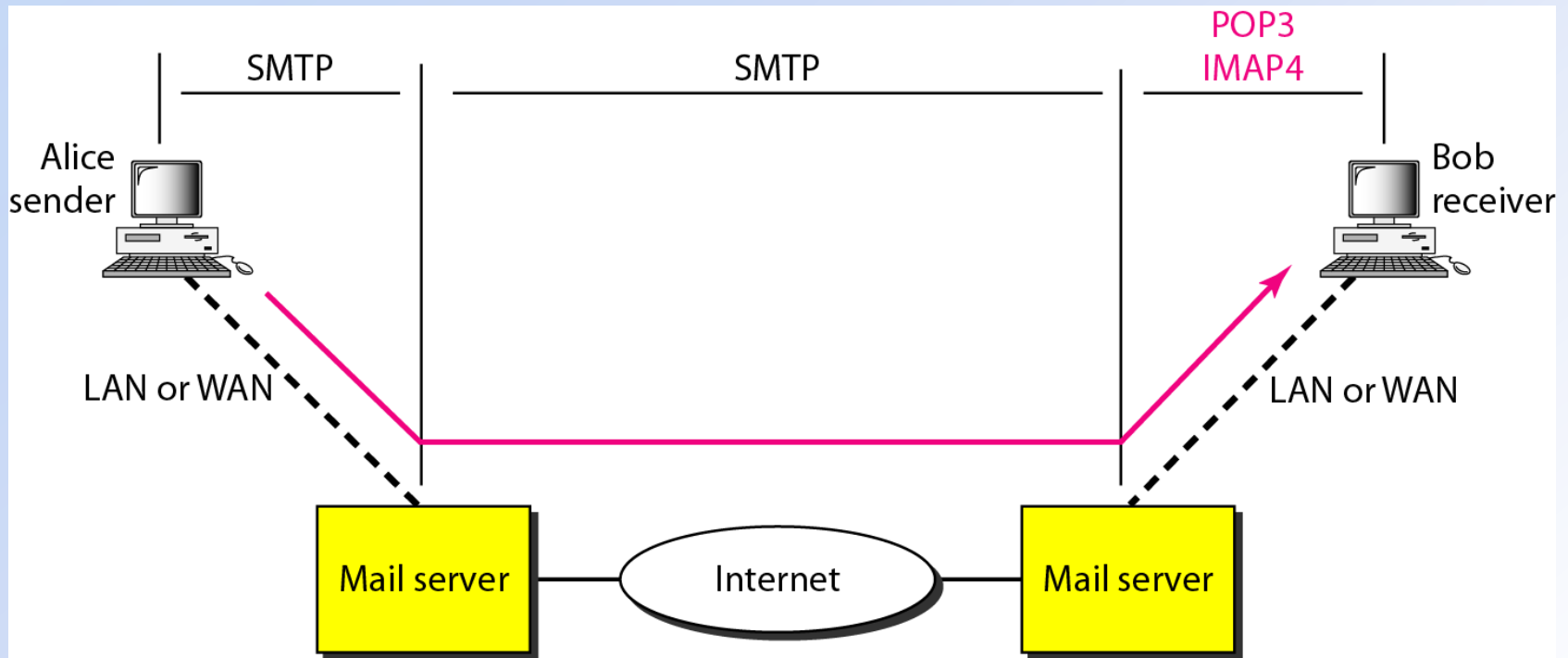
# 电子邮件Email



- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件

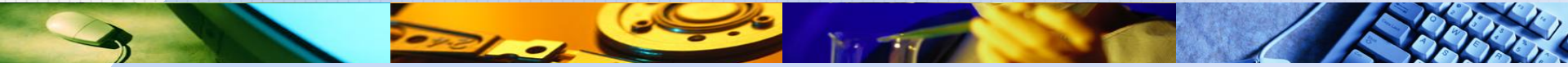


# 邮件读取协议POP3 和 IMAP



IMAP4与POP3相似，但IMAP4功能更强大、更复杂。

# POP3



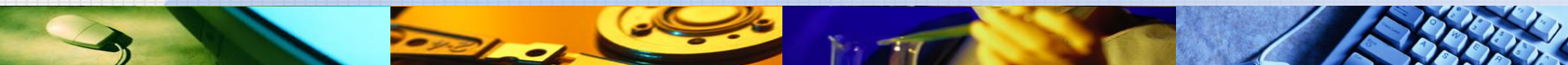
- ❖ 邮局协议 POP 是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 POP3。
- ❖ POP 也使用客户服务器的工作方式。
- ❖ 在接收邮件的用户 PC 机中必须运行 POP 客户程序，而在用户所连接的 ISP 的邮件服务器中则运行 POP 服务器程序。

# IMAP 协议(Internet Message Access Protocol)



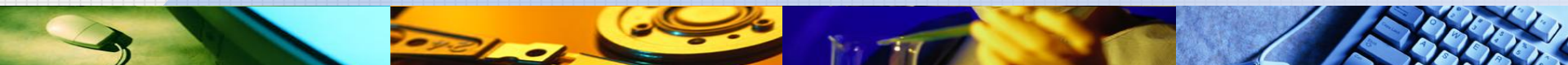
- ❖ IMAP 也是按客户服务器方式工作，现在较新的是版本 4，即 IMAP4。
- ❖ 用户在自己的 PC 机上就可以操纵 ISP 的邮件服务器的邮箱，就像在本地操纵一样。
- ❖ 因此 IMAP 是一个联机协议。当用户 PC 机上的 IMAP 客户程序打开 IMAP 服务器的邮箱时，用户就可看到邮件的首部。若用户需要打开某个邮件，则该邮件才传到用户的计算机上。

# IMAP 的特点



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

# POP3和IMAP4基于TCP服务收取邮件



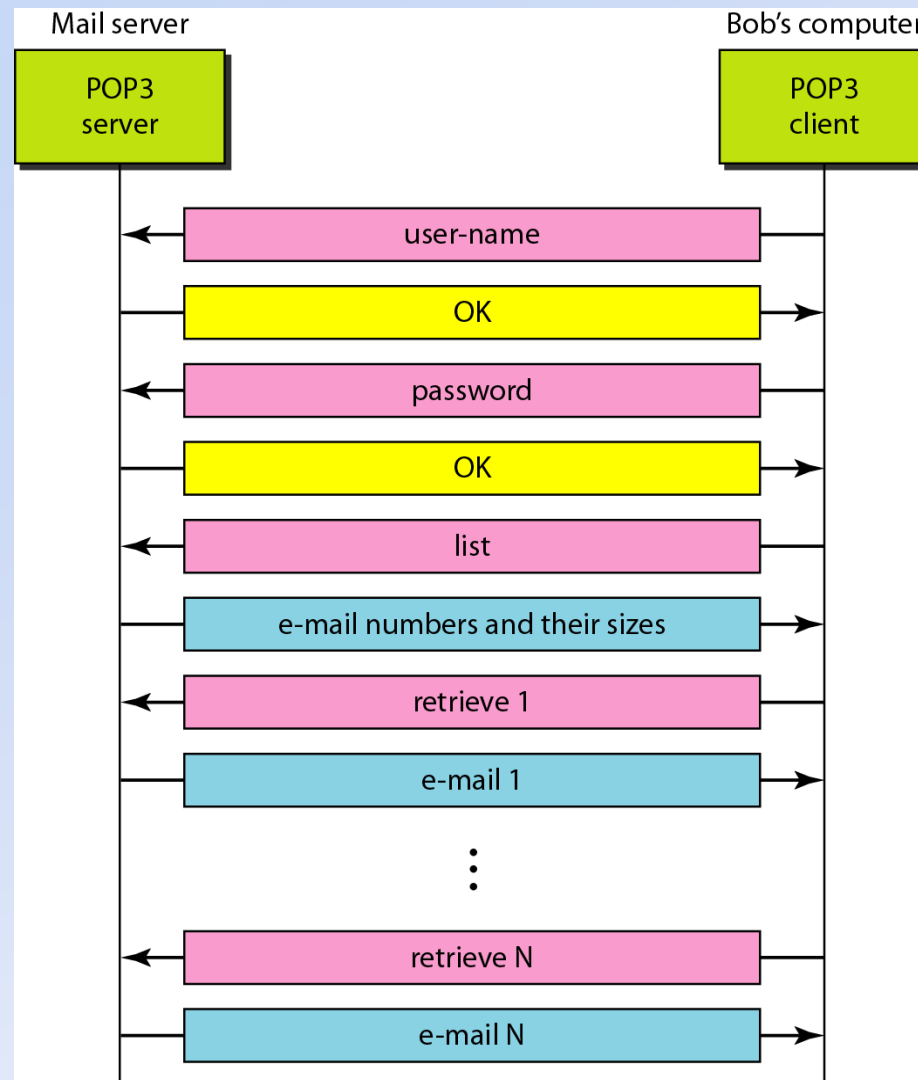
---

POP3 uses the services of TCP on well-known port 110.  
IMAP4 uses the services of TCP on well-known port 143.

---

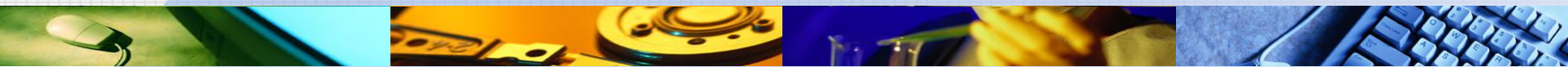
用户端代理开启TCP连接，并使用TCP连接向服务器发送命令访问邮箱，读取邮件。

# POP3收取文件过程示例（命令和响应的交互）





# 电子邮件Email



- ❖ 电子邮件简介
- ❖ 电子邮件系统结构及邮件过程
- ❖ 电子邮件信息格式
- ❖ 邮件传送协议SMTP
- ❖ 邮件读取协议POP3 和 IMAP
- ❖ 基于万维网的电子邮件

# 基于万维网的电子邮件

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



# 主题五



1 协议层次回顾

2 应用层简介

3 域名系统DNS

4 电子邮件Email

5 万维网WWW

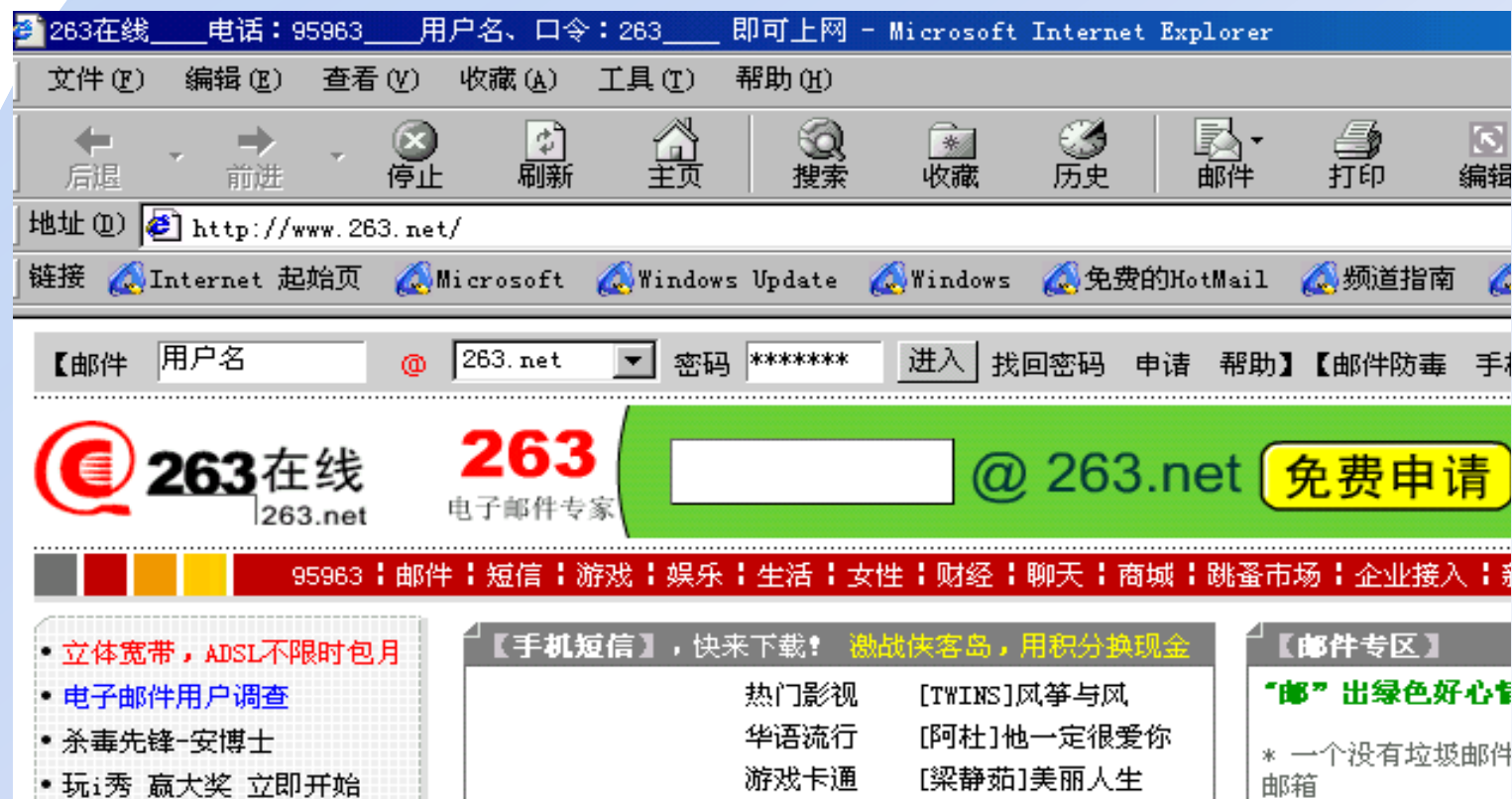
# WWW万维网



## ❖ 概述

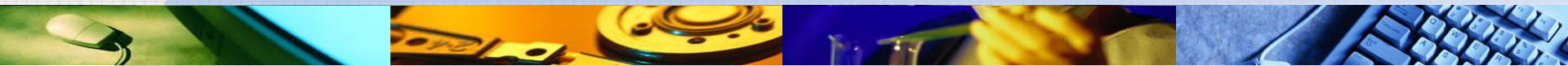
- ❖ 统一资源定位符
- ❖ 超文本传输协议
- ❖ 万维网的文档
- ❖ 万维网的信息检索系统
- ❖ 博客、微波和轻博

# 网上冲浪



263主页

# 网上冲浪：要点



## ❖ 操作要点

- **IE浏览器**
- **访问网站的域名或IP地址输入**
- **内容丰富多彩：文字、音频、视频、动画等**
- **“链接”使得转换内容或页面方便无比**
- **网站 - Web服务器**
- **B-S模式，但是是“瘦客户端”**

## ❖ 效果

- **“秀才不出门，便知天下事”**

## ❖ 疑问

- **很奇妙，这是怎么实现的？**

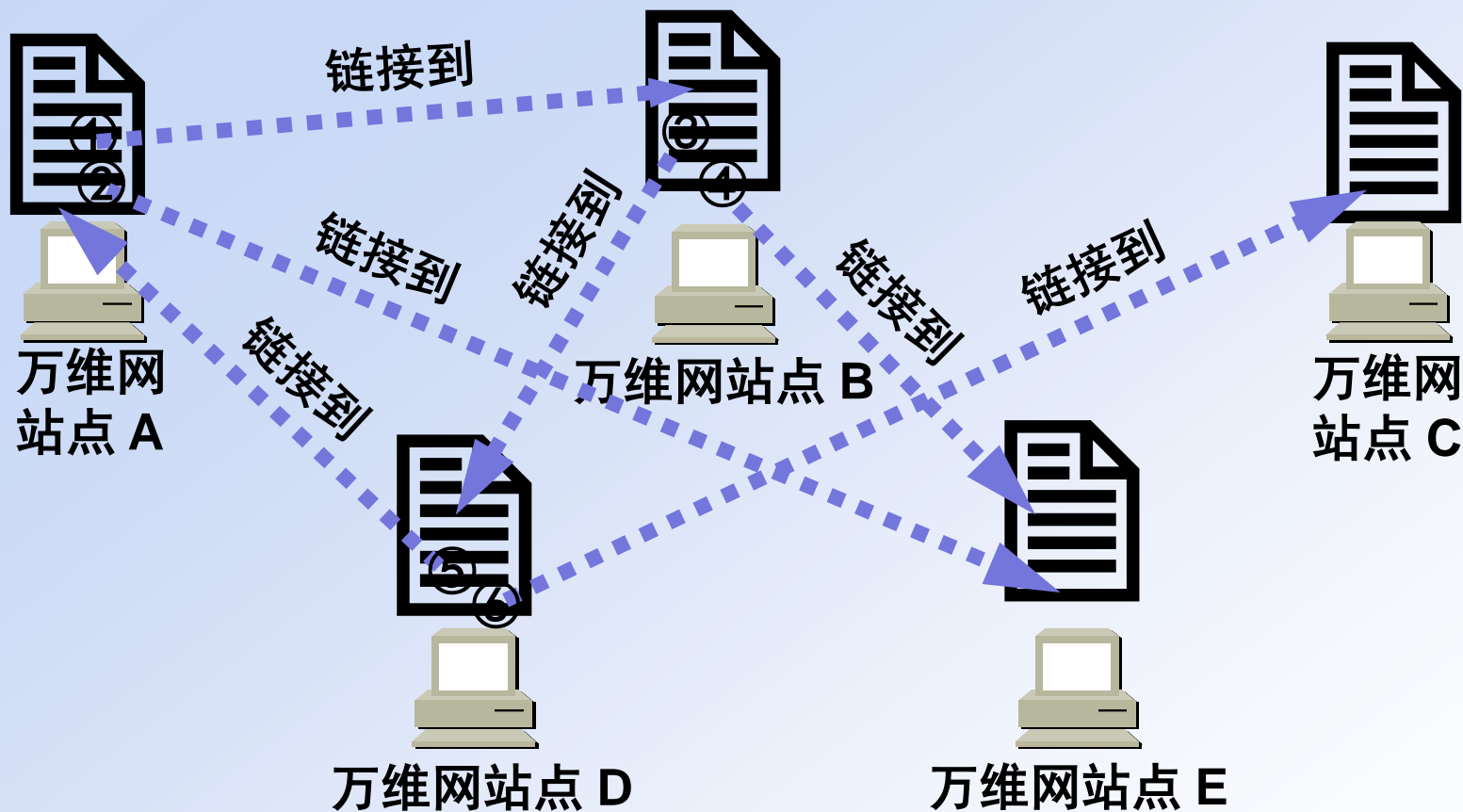


# WWW 概述

- 万维网 WWW (World Wide Web)并非某种特殊的计算机网络。
- 万维网是一个大规模的、联机式的信息储藏所。
- 万维网用链接的方法能非常方便地从因特网上的一个站点访问另一个站点，从而主动地按需获取丰富的信息。
- 这种访问方式称为 **“链接”**。

信息类型	Telnet	FTP	Email	Web	其他
比例	1.6%	8.4%	10.7%	78.3%	1%

# 分布式服务



# 超媒体与超文本



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

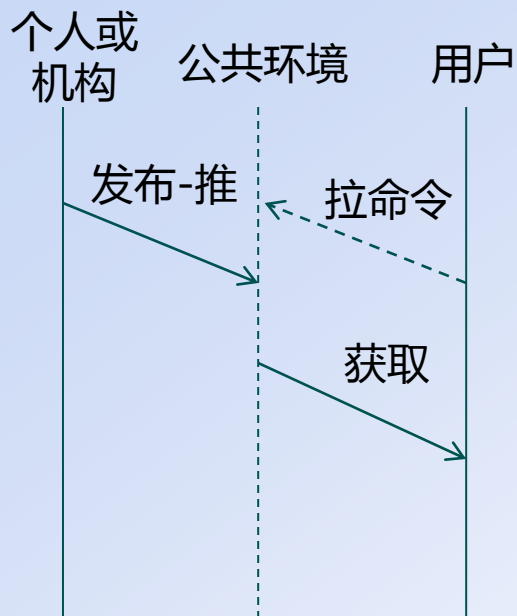
# 工作方式



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

# WWW服务及应用需求

类似于浏  
览报刊



## 特点分析

发布者：

- ① 一对多
- ② 多样信息
- ③ 吸引用户
- ④ 编排和布局很重要

接收者：

- ① 主动选择
- ② 忽略或丢弃多

同步：

- ① 二者推和送不要求同步
- ② 推和拉均主动，但因不同步，相互间是随机的

操作程序

- ① 先推
- ② 后拉

格式：布局 and 编排  
至关重要

- ① 栏目
- ② 头版
- ③ 重要新闻
- ④ 重用度高的
- ⑤ 滚动新闻
- ⑥ 图片
- ⑦ 视频
- ⑧ 关联机构

字典 + 语义



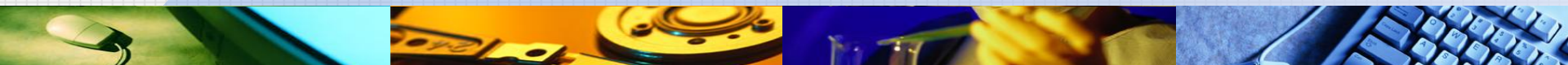
# WWW网需要解决的问题



263主页



# WWW网需要解决的问题



## (1) 定位：怎样标志分布在整个因特网上的万维网文档？

使用统一资源定位符 URL (Uniform Resource Locator)来标志万维网上的各种文档。

使每一个文档在整个因特网的范围内具有惟一的标识符 URL。

---

## (2) 协议：用何协议实现万维网上各种超链的链接？

在万维网客户程序与万维网服务器程序之间进行交互所使用的协议，是超文本传送协议 HTTP (HyperText Transfer Protocol)。便于随机选择，释放资源。

HTTP 是一个应用层协议，它使用 TCP 连接进行可靠的传送。

---

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

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

---

## (4) 便利：怎样使用户能够很方便地找到所需的信息？

为了在万维网上方便地查找信息，用户可使用各种的搜索工具（即搜索引擎）。

# WWW万维网



- ❖ 概述
- ❖ 统一资源定位符
- ❖ 超文本传输协议
- ❖ 万维网的文档
- ❖ 万维网的信息检索系统
- ❖ 博客、微波和轻博

# 统一资源定位符：URL



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

# URL的一般形式

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

**<URL的访问方式>://<主机>:<端口>/<路径>**

**ftp —— 文件传送协议 FTP**

**http —— 超文本传送协议 HTTP**

**news —— USENET 新闻**

# URL的一般形式

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

**<URL的访问方式>://<主机>:<端口>/<路径>**

**<主机> 是存放资源的主机  
在因特网中的域名**

# URL的一般形式

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

<URL的访问方式>://<主机>:<端口>/<路径>

有时可省略



# 使用 HTTP 的 URL

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

这表示使用 HTTP 协议

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

冒号和两个斜线是规定的格式

# 使用 HTTP 的 URL

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

这里写主机的域名

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

**HTTP 的默认端口号是 80，通常可省略**

# 使用 HTTP 的 URL

- 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

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

# 使用FTP 的 URL

- 使用 FTP 的 URL 举例

**ftp**://rtfm.mit.edu/pub/abc.txt

这表示使用 **FTP** 协议

- 使用 FTP 的 URL 举例

ftp**:**//rtfm.mit.edu/pub/abc.txt

冒号和两个斜线是规定的格式

# 使用FTP 的 URL

- 使用 FTP 的 URL 举例

ftp://rtfm.mit.edu/pub/abc.txt

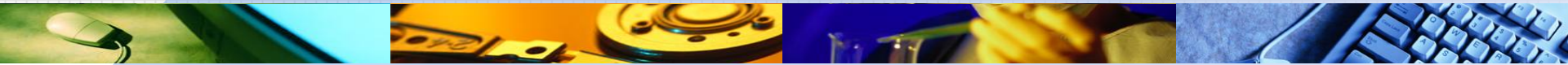
这是网站的域名

- 使用 FTP 的 URL 举例

ftp://rtfm.mit.edu/pub/abc.txt

这是路径和文件名

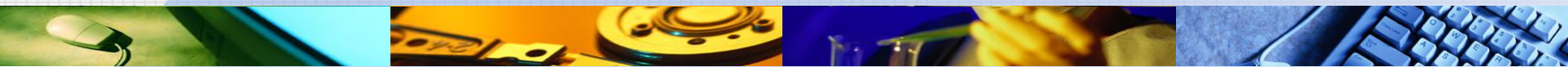
# WWW万维网



- ❖ 概述
- ❖ 统一资源定位符
- ❖ **超文本传输协议**
- ❖ 万维网的文档
- ❖ 万维网的信息检索系统
- ❖ 博客、微波和轻博

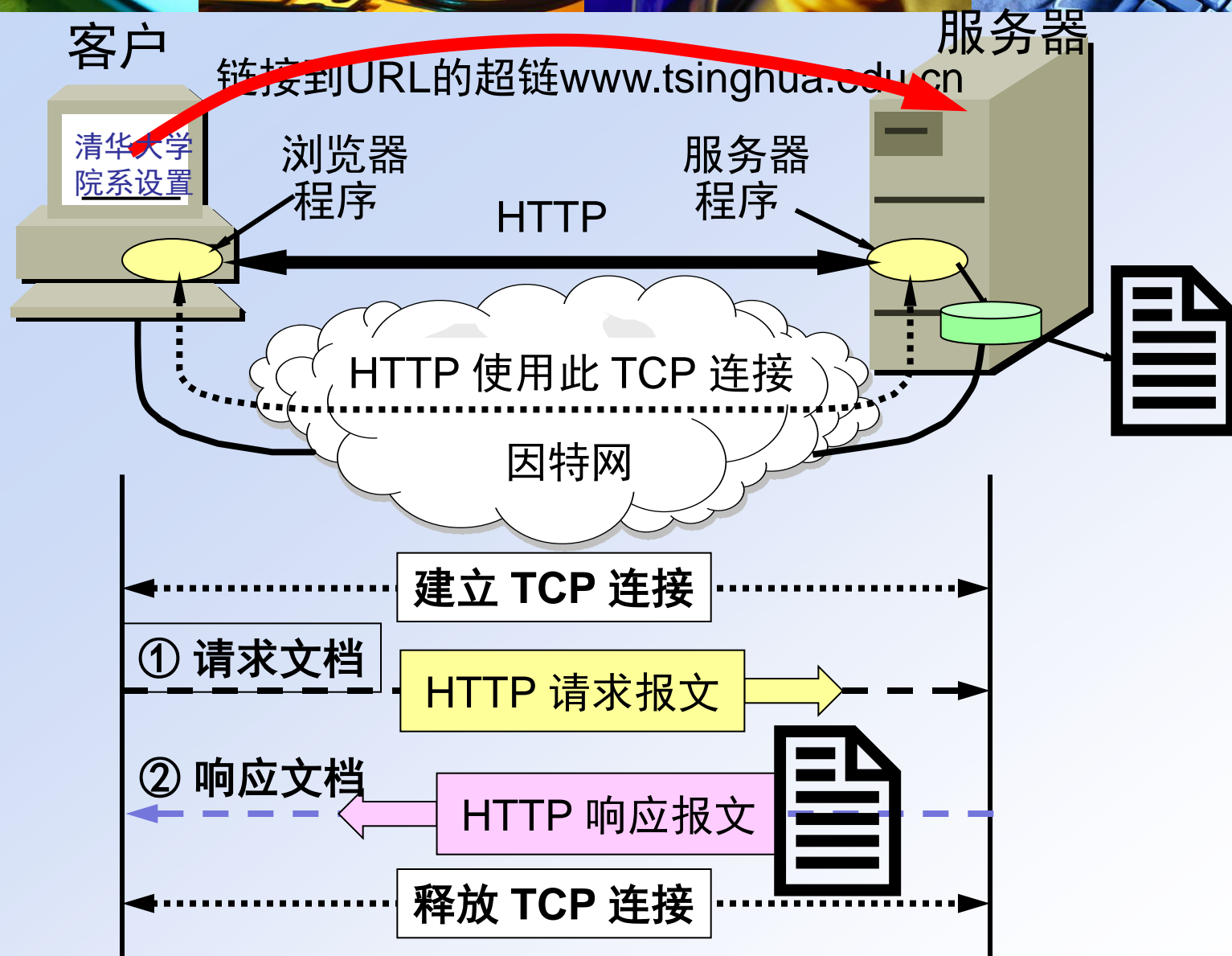


# 超文本传输协议：HTTP



- 为了使超文本的链接能够高效率地完成，需要用 HTTP 协议来传送一切必须的信息。
- HTTP 是面向事务的(transaction-oriented)应用层协议。
- HTTP是万维网上能够可靠地交换文件（包括文本、声音、图像等各种多媒体文件）的重要基础。
- HTTP是无状态的协议
- HTTP是无连接协议，使用了面向连接的TCP传输
- HTTP 是客户/服务器协议，浏览器就是一个客户端，服务器上等待请求的进程是HTTP daemon——HTTPD
- HTTPD响应请求，传输文件给客户

# HTTP—工作过程

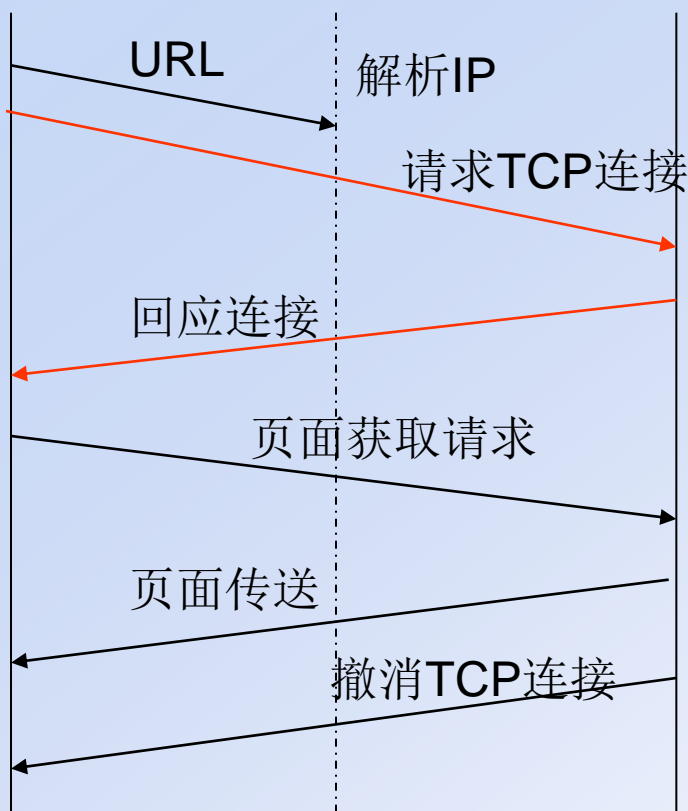


# HTTP—工作过程：C/S模式

浏览器（启动）

DNS

服务器

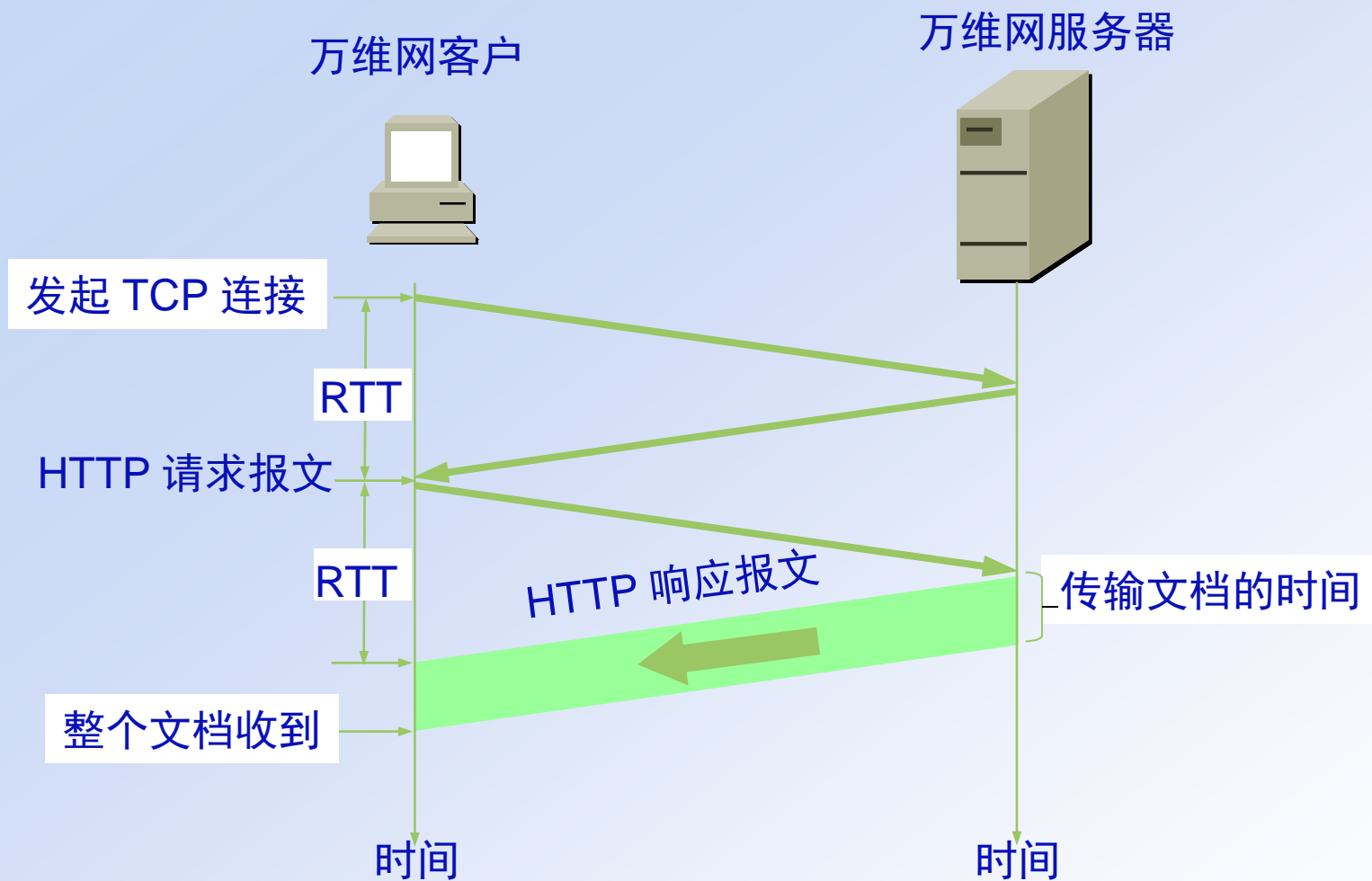


- 1) 多种协议协同工作
- 2) 运输层协议提供可靠通道
- 3) 运输层协议的具体运作过程其实对应用层而言是“透明”的
- 4) DNS解析运输层采用的是UDP协议——树型遍历

# 用户点击鼠标后所发生的事件

- (1) 浏览器分析超链指向页面的 URL。
- (2) 浏览器向 DNS 请求解析 `www.tsinghua.edu.cn` 的 IP 地址。
- (3) 域名系统 DNS 解析出清华大学服务器的 IP 地址。
- (4) 浏览器与服务器建立 TCP 连接
- (5) 浏览器发出取文件命令：  
    `GET /chn/yxsx/index.htm`。
- (6) 服务器给出响应，把文件 `index.htm` 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示“清华大学院系设置”文件 `index.htm` 中的所有文本。

# 请求一个万维网文档所需的时间





# 持续连接 (persistent connection)



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



# 持续连接的两种工作方式



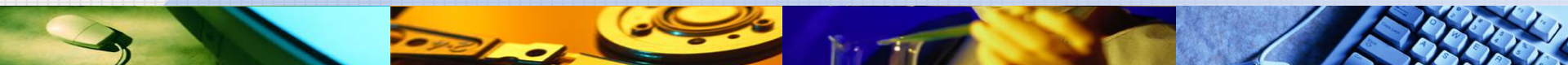
❖ 非流水线方式：客户在收到前一个响应后才能发出下一个请求。

- 这比非持续连接的两倍 RTT 的开销节省了建立 TCP 连接所需的一个 RTT 时间。
- 但服务器在发送完一个对象后，其 TCP 连接就处于空闲状态，浪费了服务器资源。

❖ 流水线方式：客户在收到 HTTP 的响应报文之前就能够接着发送新的请求报文。

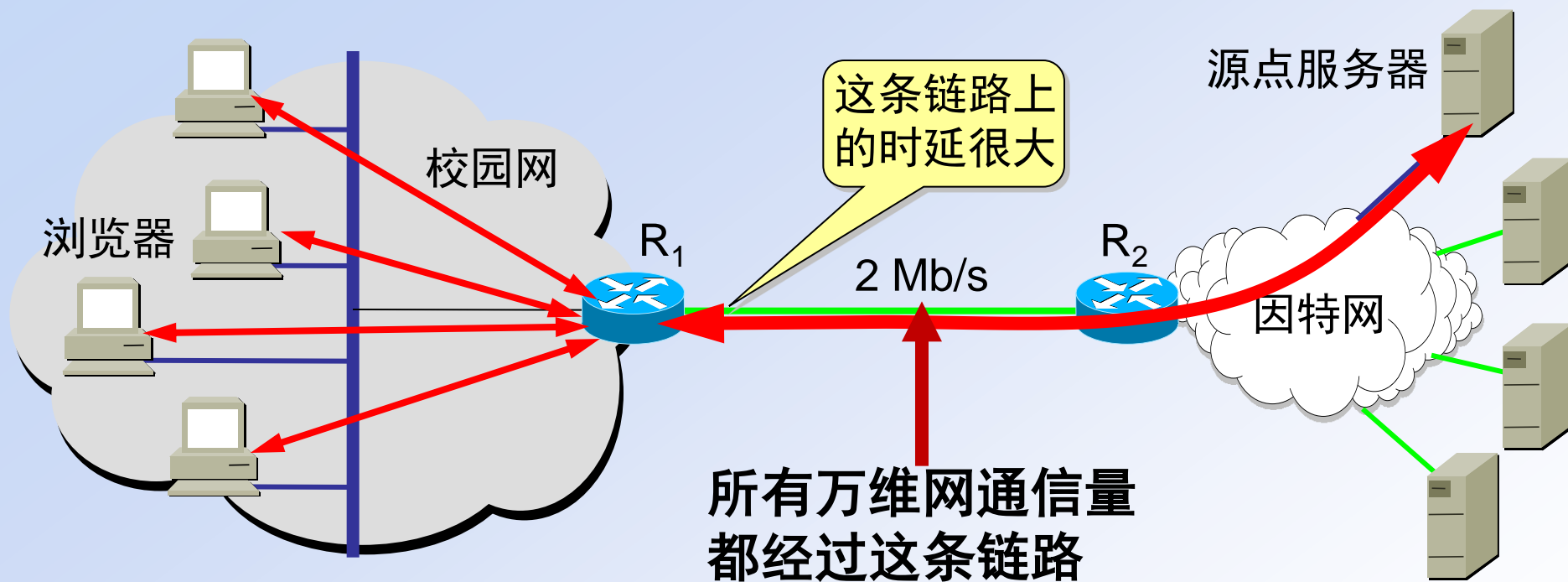
- 一个接一个的请求报文到达服务器后，服务器就可连续发回响应报文。
- 使用流水线方式时，客户访问所有的对象只需花费一个 RTT 时间。
- 使 TCP 连接中的空闲时间减少，提高了下载文档效率。

# HTTP—高速缓存



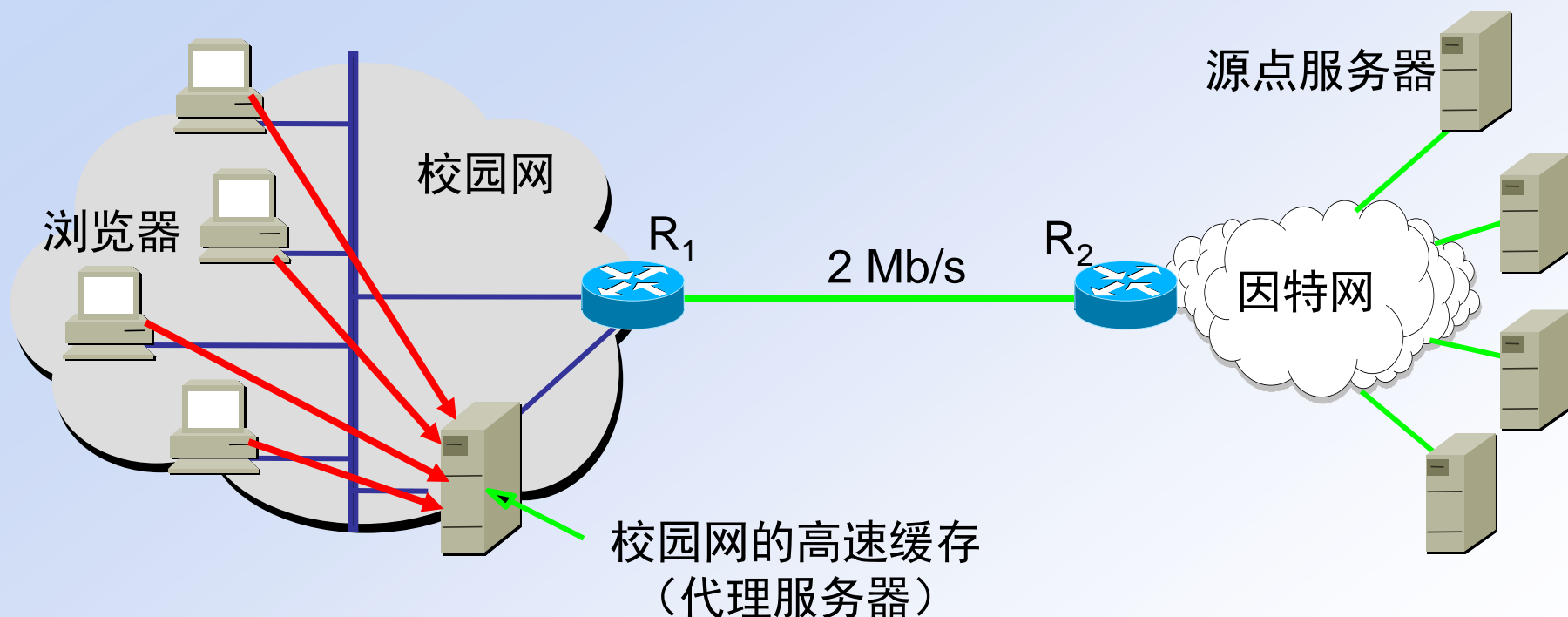
- 万维网高速缓存能代表浏览器发出 HTTP 请求，因此又称为代理服务器(proxy server)。
- 万维网高速缓存将最近的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求相同的新请求到达时，万维网高速缓存就把暂存的响应发送出去，而不需要按 URL 的地址再去因特网访问该资源。

# HTTP—无缓存



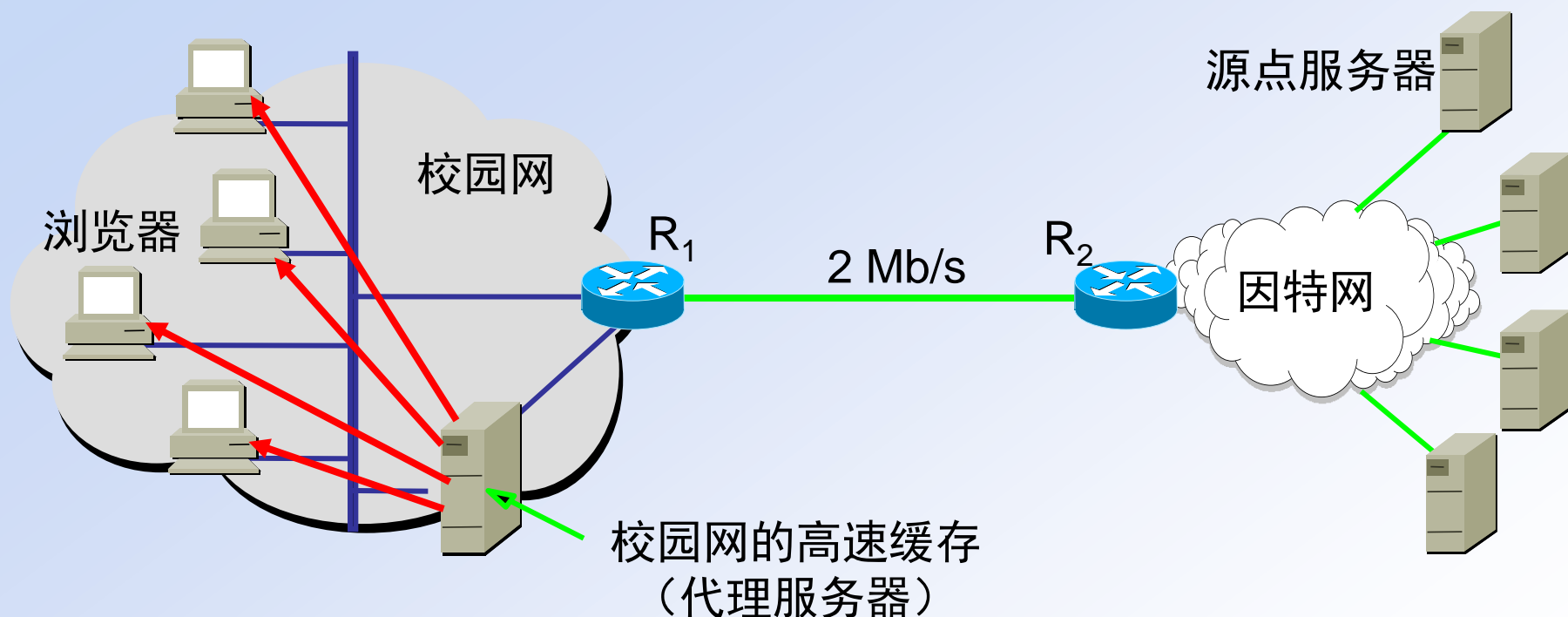
# HTTP—有缓存

(1) 浏览器访问因特网的服务器时，要先与校园网的高速缓存建立 TCP 连接，并向高速缓存发出 HTTP 请求报文。



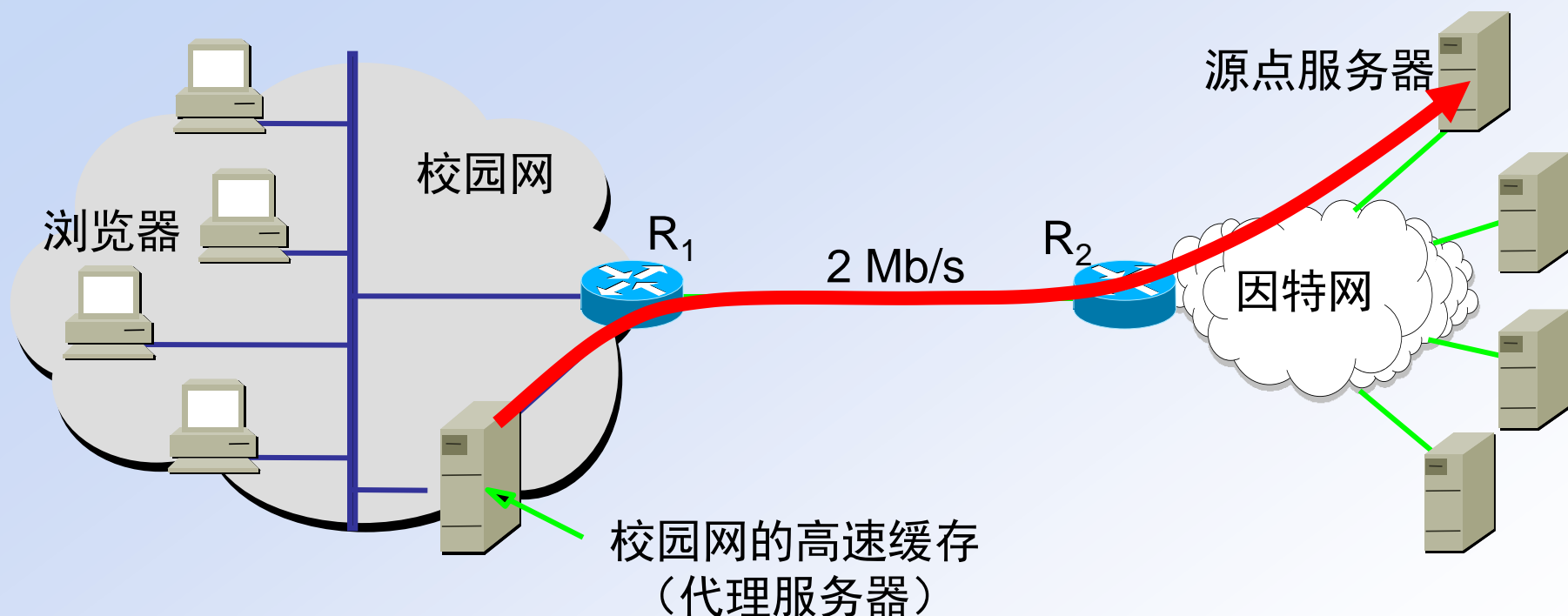
# HTTP—有缓存

(2) 若高速缓存已经存放了所请求的对象，则将此对象放入 HTTP 响应报文中返回给浏览器。



# HTTP—有缓存

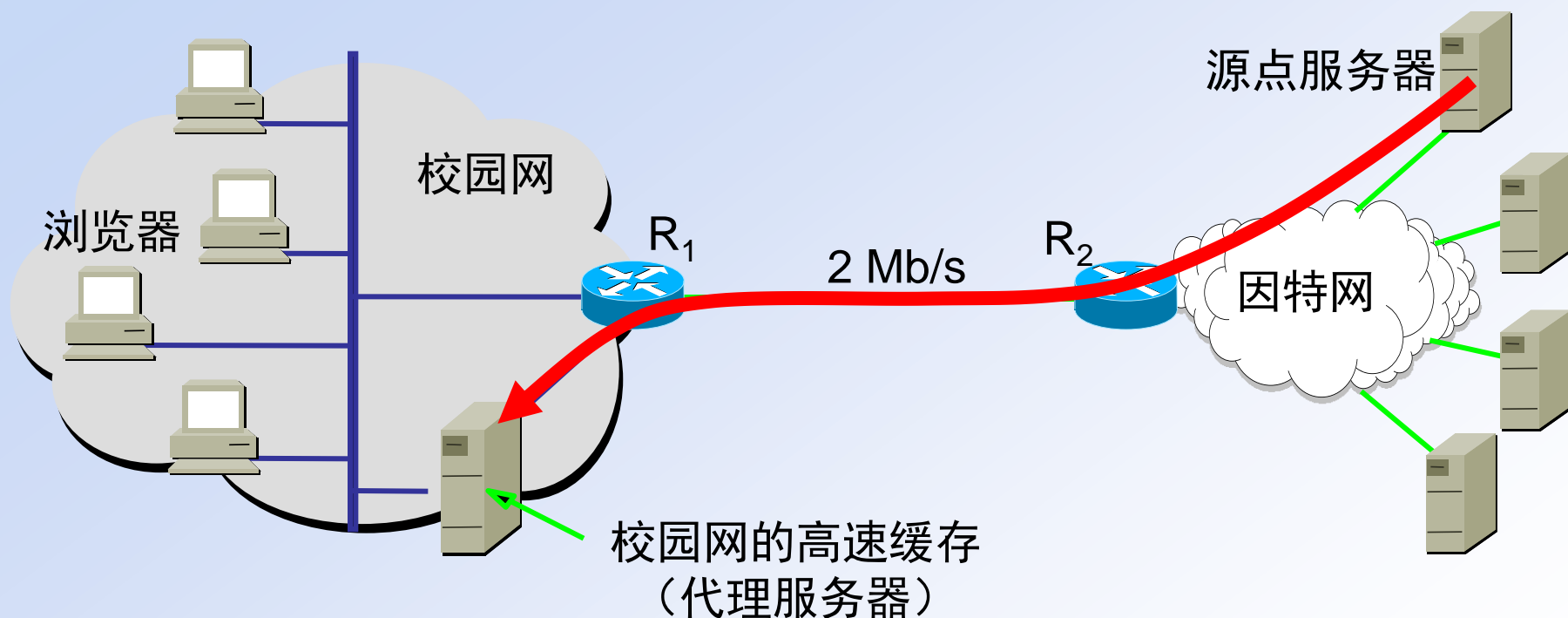
(3) 否则，高速缓存就代表发出请求的用户浏览器，与因特网上的源点服务器建立 TCP 连接，并发送 HTTP 请求报文。





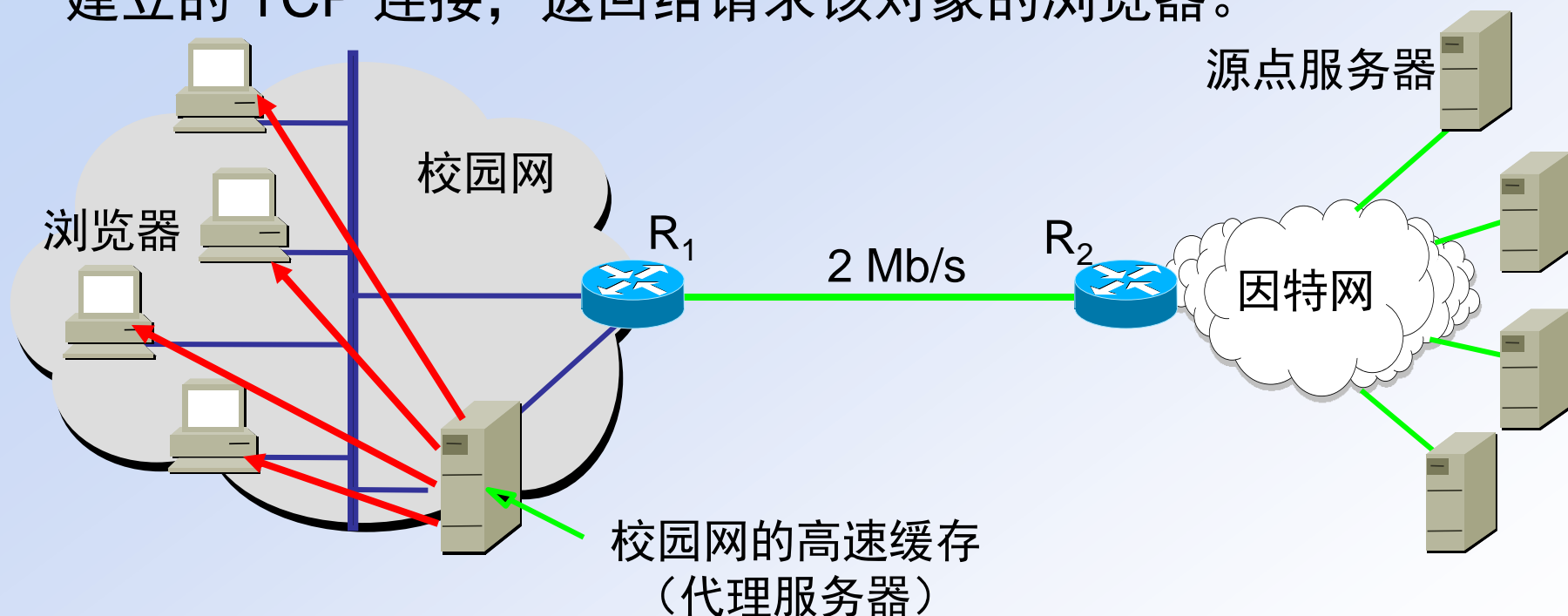
# HTTP—有缓存

(4) 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存。

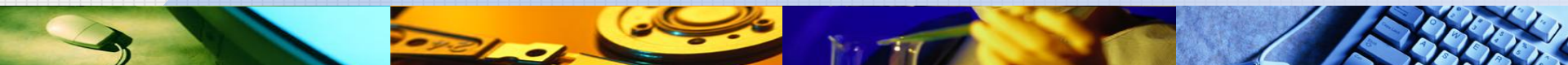


# HTTP—有缓存

(5) 高速缓存收到此对象后，先复制在其本地存储器中（为今后使用），然后再将该对象放在 HTTP 响应报文中，通过已建立的 TCP 连接，返回给请求该对象的浏览器。



# HTTP—报文结构

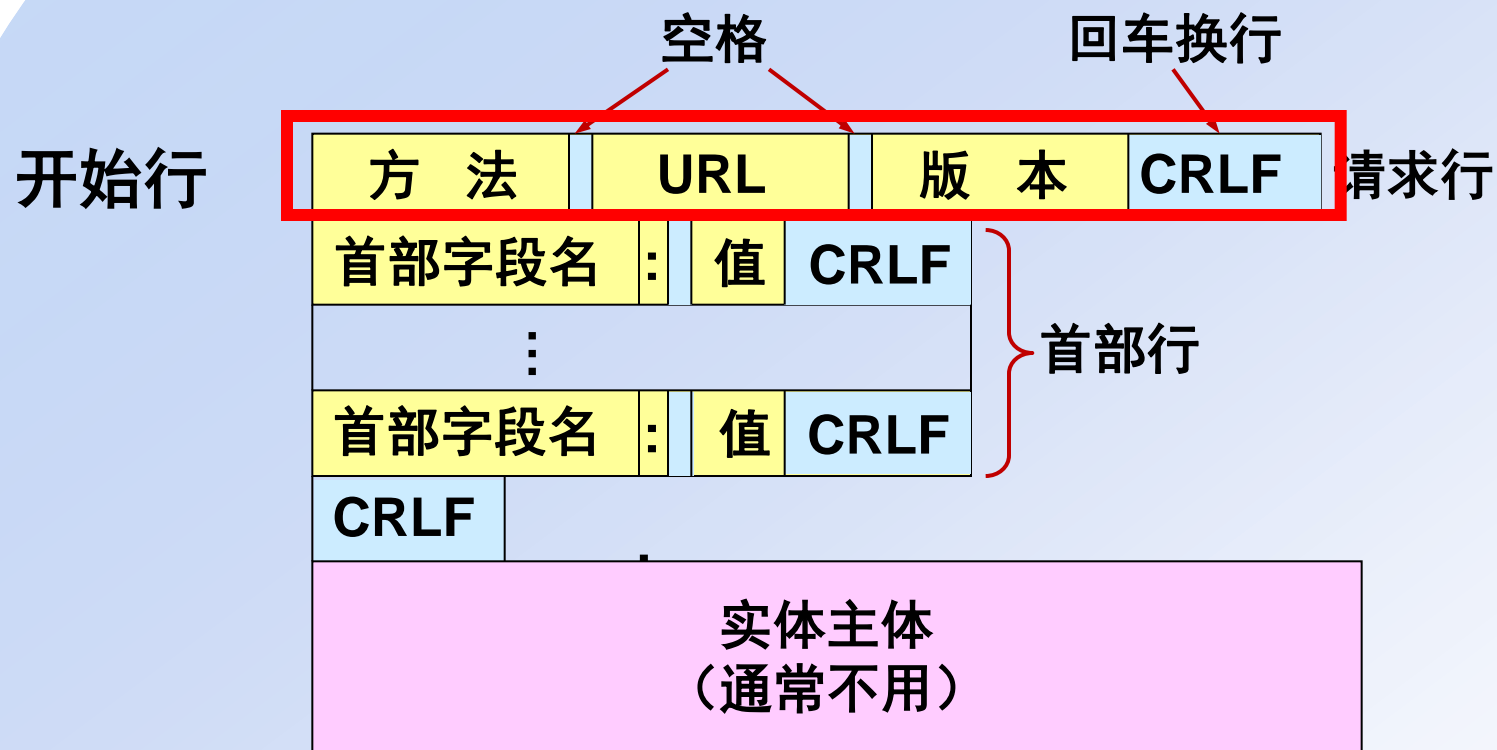


HTTP 有两类报文：

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

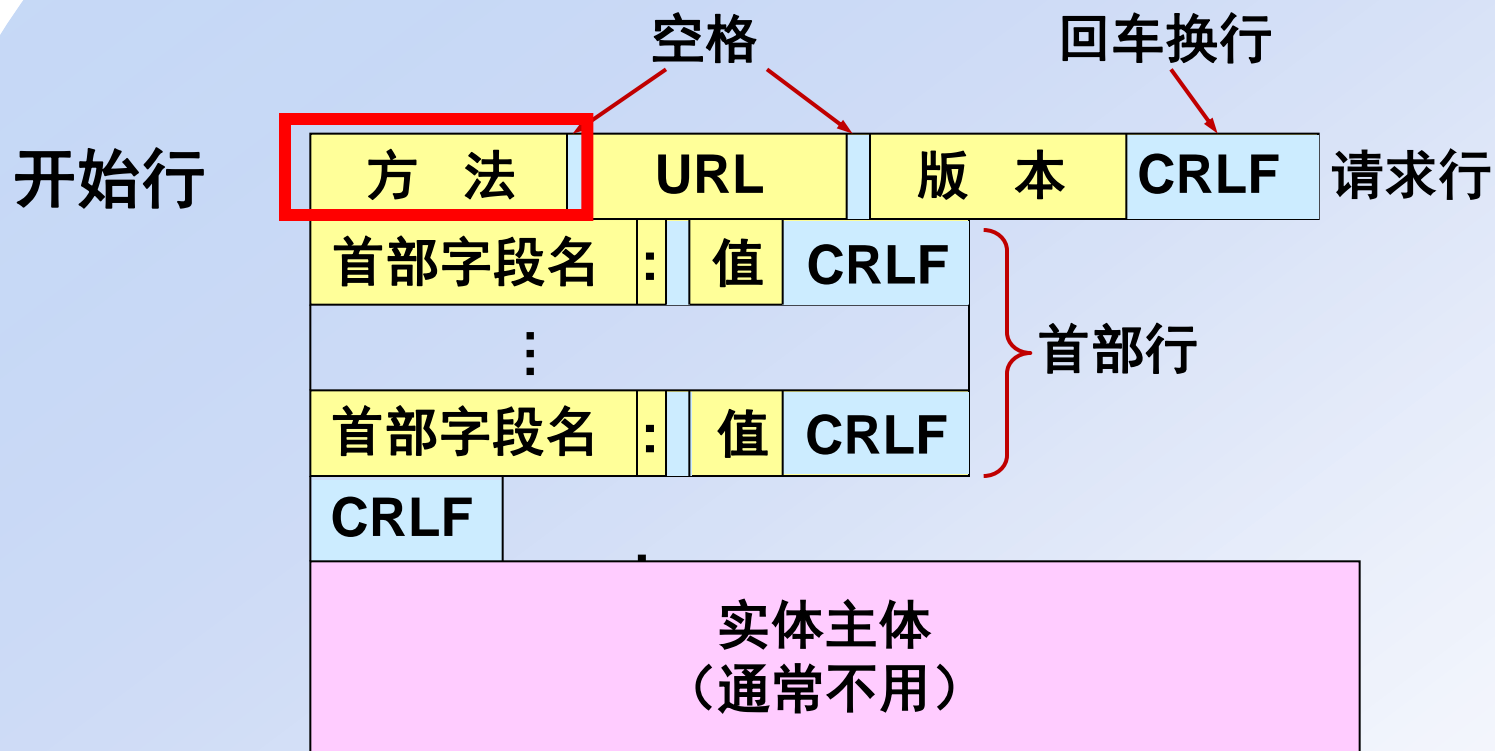
由于 HTTP 是面向正文的，在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的。

# HTTP—报文结构：请求报文



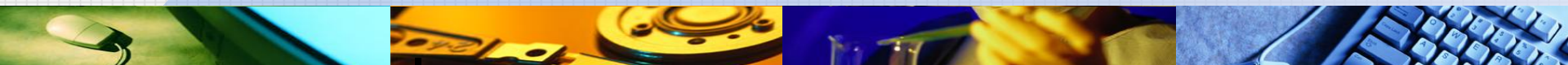
报文由三个部分组成，即开始行、首部行和实体主体。在请求报文中，开始行就是请求行。

# HTTP—报文结构：请求报文



“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的。

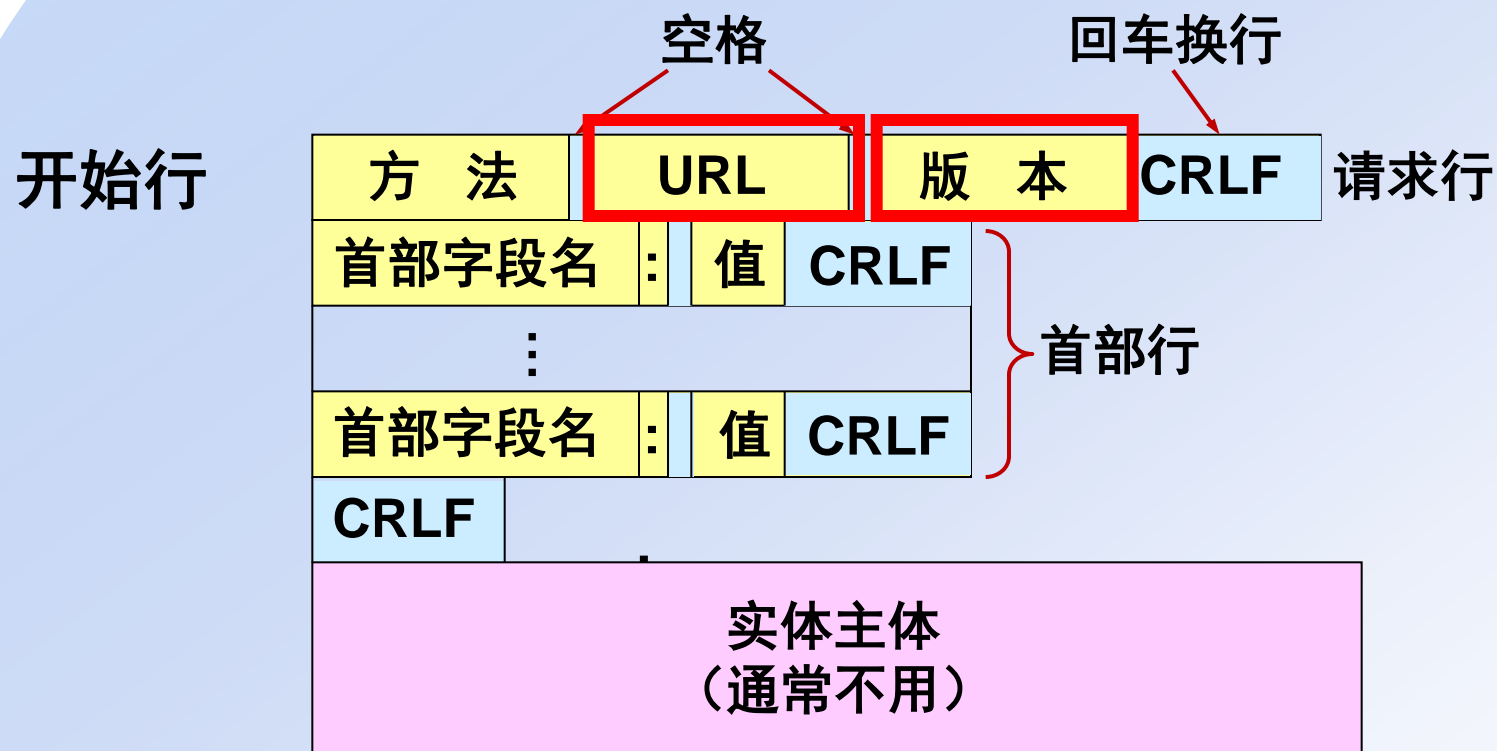
# HTTP—请求报文中的方法



方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由URL所标志的信息
HEAD	请求读取由URL所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的URL下存储一个文档
DELETE	删除指明的URL所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器



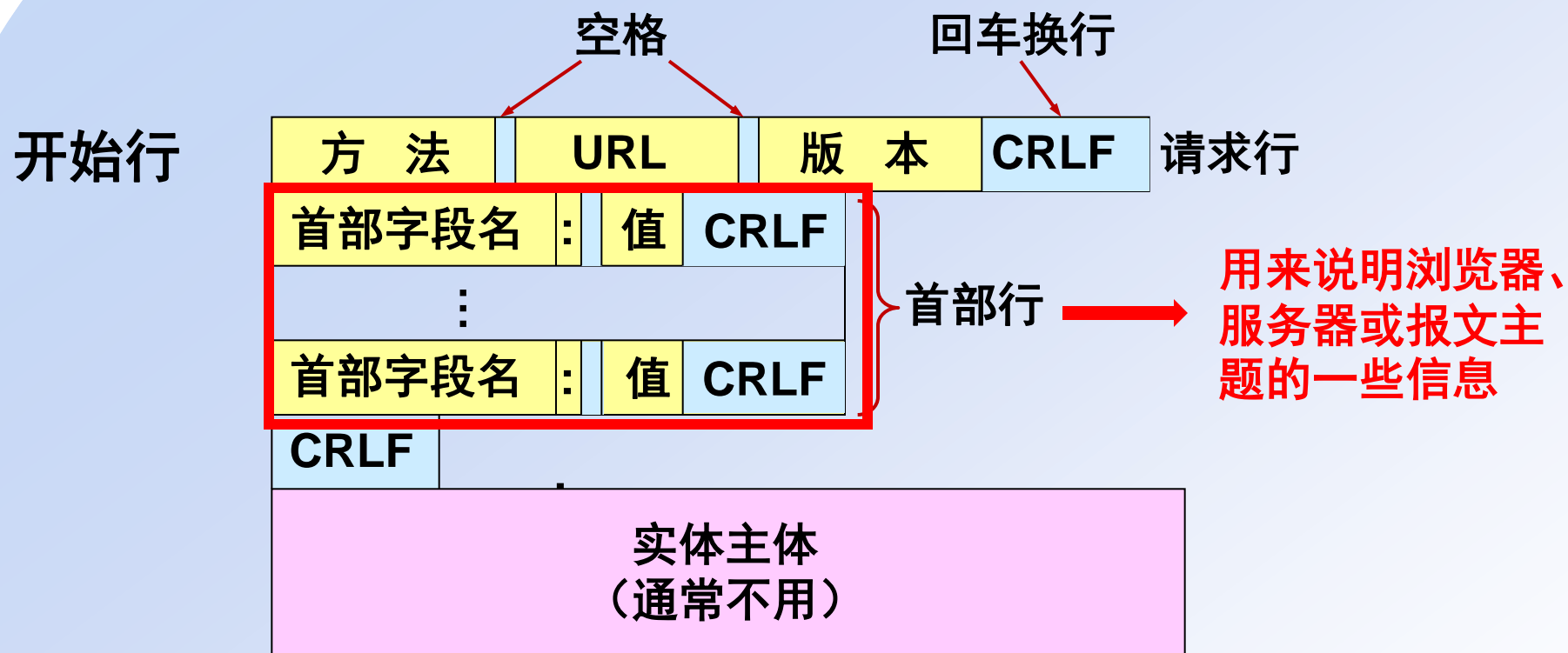
# HTTP—报文结构：请求报文



“URL”是所请求的资源的 URL。

“版本” 是 HTTP 的版本。

# HTTP—报文结构：请求报文



“URL”是所请求的资源的 URL。

“版本”是 HTTP 的版本。

# HTTP请求报文的例子



GET /chn/yxsx/index.htm HTTP/1.1

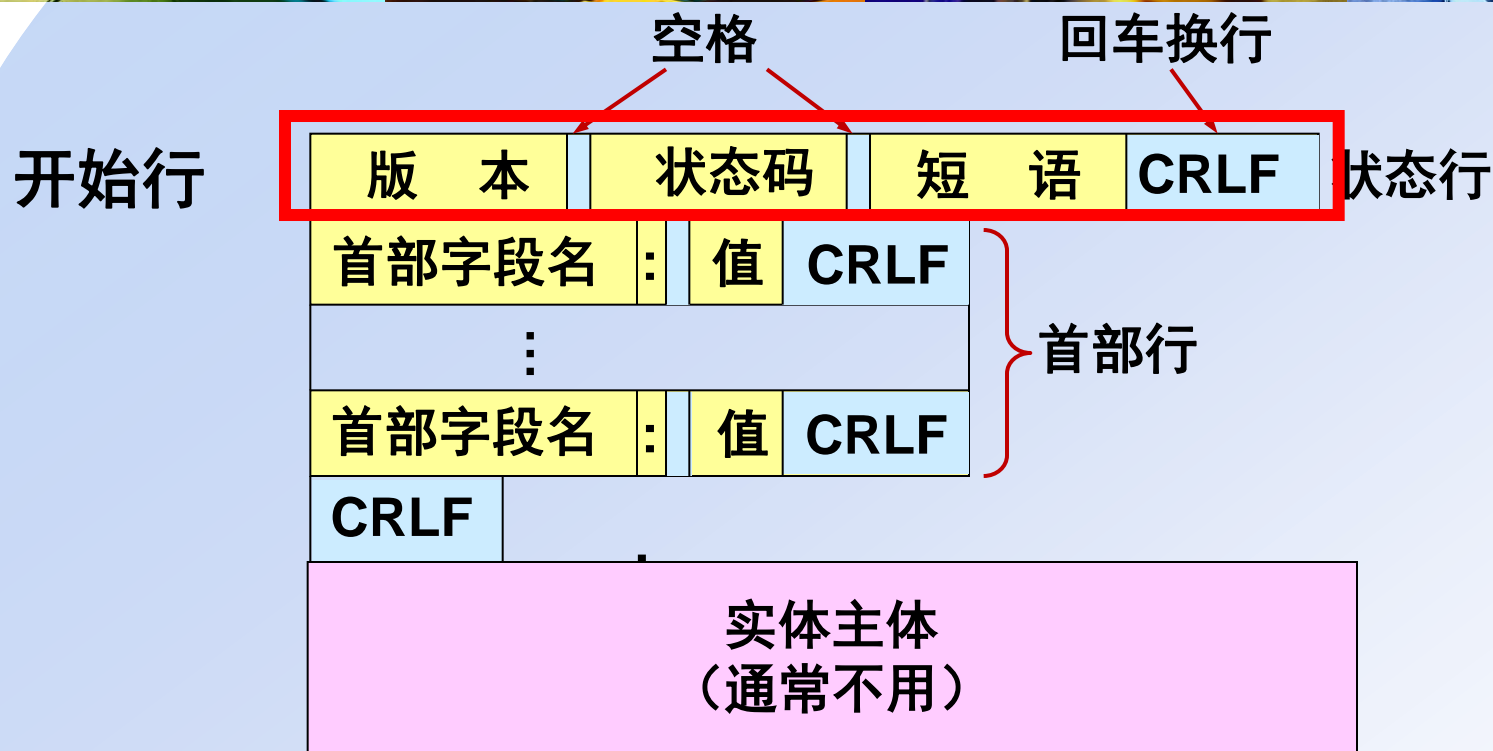
Host: [www.tsinghua.edu.cn](http://www.tsinghua.edu.cn)

Connection: close

User-Agent: Mozilla/5.0

Accept-Language: cn

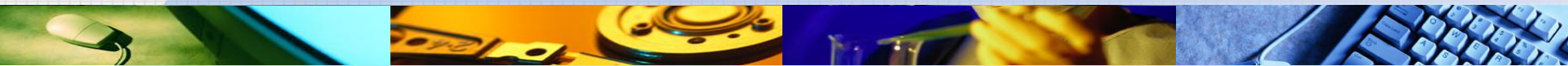
# HTTP—报文结构：响应报文



响应报文的开始行是**状态行**。

状态行包括三项内容，即 **HTTP** 的版本，**状态码**，以及解释状态码的**简单短语**。

# HTTP—响应报文中的状态码



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

# HTTP响应报文的例子



HTTP/1.1 202 Accepted

HTTP/1.1 400 Bad Request

HTTP/1.1 404 Not Found

HTTP/1.1 301 Moved Permanently

Location: <http://www.xyz.edu/ee/index.html>



# 在服务器上存放用户的信息



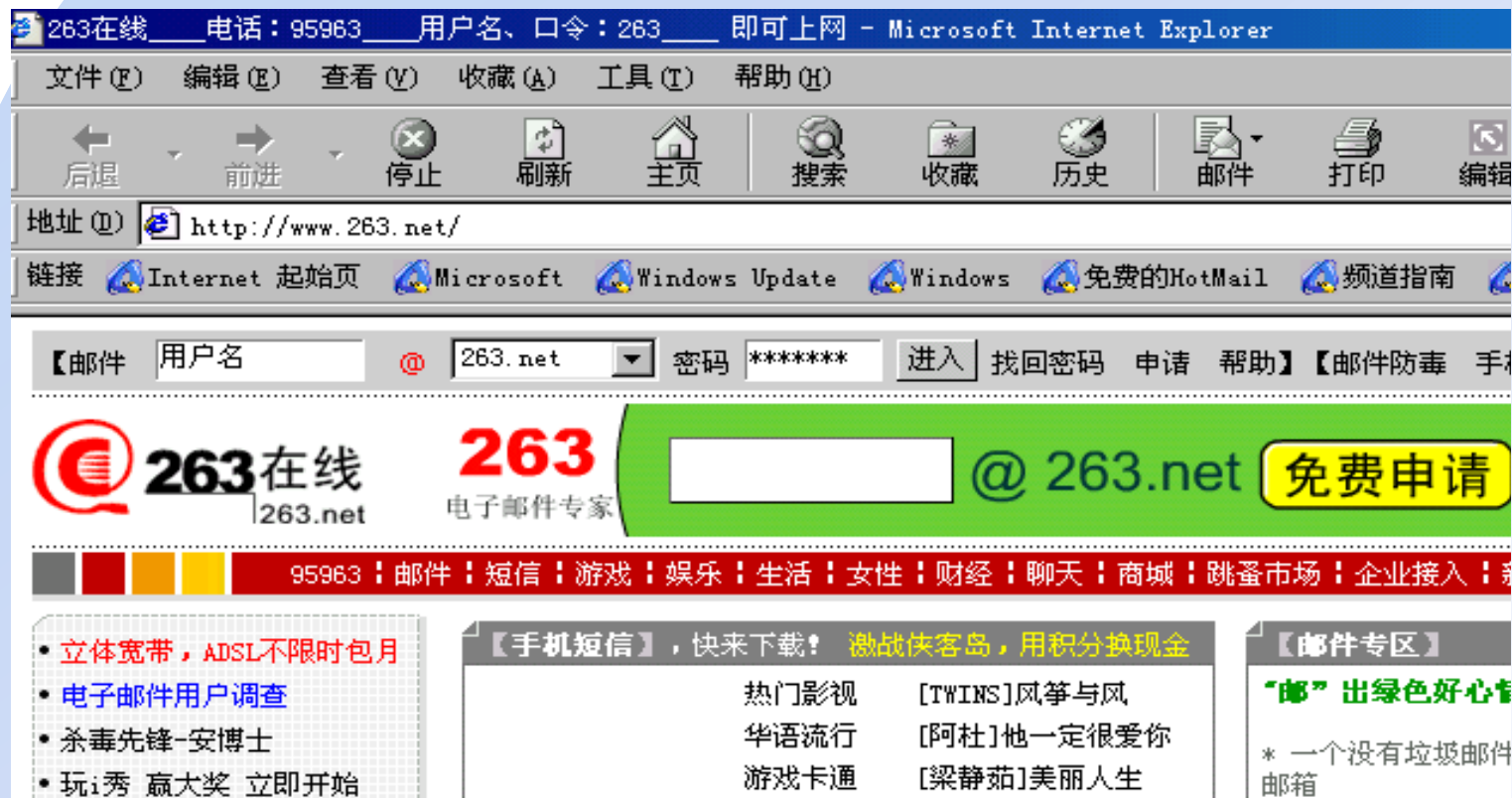
- ❖ 万维网站点使用 Cookie 来跟踪用户。
- ❖ Cookie 表示在 HTTP 服务器和客户之间传递的状态信息。
- ❖ 使用 Cookie 的网站服务器为用户产生一个唯一的识别码。利用此识别码，网站就能够跟踪该用户在该网站的活动。

# WWW万维网



- ❖ 概述
- ❖ 统一资源定位符
- ❖ 超文本传输协议
- ❖ **万维网的文档**
- ❖ 万维网的信息检索系统
- ❖ 博客、微波和轻博

# 超文本标记语言HTML：有格式的信息展示



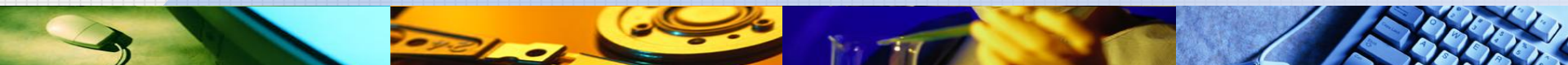
263主页

# 超文本标记语言HTML



- Ted Nelsion 1960 年代提出 HTML(Hypertext Markup Language)
- 超文本标记语言 HTML 中的 Markup 的意思就是“**设置标记**”。
- HTML 定义了许多**用于排版的命令（标签）**。
- HTML 把各种**标签嵌入到万维网的页面中**。这样就构成了所谓的 HTML 文档。**HTML 文档**是一种可以用任何文本编辑器创建的**ASCII 码文件**。

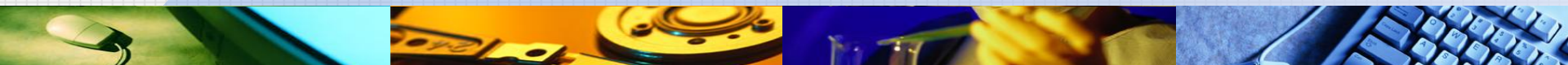
# HTML文档



- 仅当 HTML 文档是以.html 或 .htm 为后缀时，浏览器才对此文档的各种标签进行解释。
- 如 HTML 文档改换以 .txt 为其后缀，则 HTML 解释程序就不对标签进行解释，而浏览器只能看见原来的文本文件。
- 当浏览器从服务器读取 HTML 文档后，就按照 HTML 文档中的各种标签，根据浏览器所使用的显示器的尺寸和分辨率大小，重新进行排版并恢复出所读取的页面。



# HTML—标签与格式



- **元素(element)**是 HTML 文档结构的基本组成部分。一个 HTML 文档本身就是一个元素。每个 HTML 文档由两个主要元素组成：首部(head)和主体(body)。
- **首部**包含文档的标题(title)，以及系统用来标识文档的一些其他信息。标题相当于文件名。
- 文档的主体是 HTML 文档的最主要的部分。
- **主体部分**往往又由若干个更小的元素组成，如**段落(paragraph)**、**表格(table)**、和**列表(list)**等。



# HTML—标签与格式



- HTML 用一对标签（即一个开始标签和一个结束标签）或几对标签来标识一个元素。
- 开始标签由一个小于字符“<”、一个标签名、和一个大于字符“>”组成。
- 结束标签和开始标签的区别只是在小于字符的后面要加上一个斜线字符“/”。

# HTML—标签与格式



<HTML>

HTML 文档开始

<HEAD>

<TITLE>一个 HTML 的例子</TITLE>

</HEAD>

<BODY>

<H1>HTML 很容易掌握</H1>

<P>这是第一个段落。虽然很短，但它仍是一个段落。</P>

<P>这是第二个段落。</P>

</BODY>

</HTML>

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

首部开始

```
  <TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <H1>HTML 很容易掌握</H1>
```

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>一个 HTML 的例子</TITLE>
```

标题

```
</HEAD>
```

```
<BODY>
```

```
  <H1>HTML 很容易掌握</H1>
```

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

首部结束

```
<BODY>
```

```
  <H1>HTML 很容易掌握</H1>
```

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

主体开始

```
  <H1>HTML 很容易掌握</H1>
```

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```



# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <H1>HTML 很容易掌握</H1>
```

1 级标题

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
<TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>HTML 很容易掌握</H1>
```

```
<P>这是第一个段落。虽然很  
短，但它仍是一个段落。</P>
```

```
<P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

第一个段落

# HTML—标签与格式



```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>一个 HTML 的例子</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <H1>HTML 很容易掌握</H1>
```

```
  <P>这是第一个段落。虽然很  
    短，但它仍是一个段落。</P>
```

```
  <P>这是第二个段落。</P>
```

```
</BODY>
```

```
</HTML>
```

第二个段落

# HTML—标签与格式



<HTML>

<HEAD>

<TITLE>一个 HTML 的例子</TITLE>

</HEAD>

<BODY>

<H1>HTML 很容易掌握</H1>

<P>这是第一个段落。虽然很短，但它仍是一个段落。</P>

<P>这是第二个段落。</P>

</BODY>

主体结束

</HTML>

# HTML—标签与格式



<HTML>

<HEAD>

<TITLE>一个 HTML 的例子</TITLE>

</HEAD>

<BODY>

<H1>HTML 很容易掌握</H1>

<P>这是第一个段落。虽然很短，但它仍是一个段落。</P>

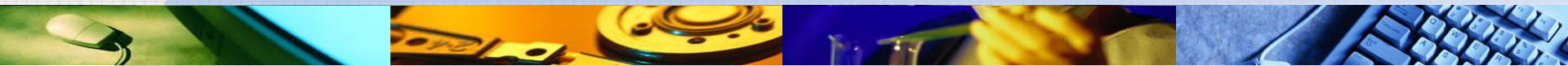
<P>这是第二个段落。</P>

</BODY>

</HTML>

**HTML 文档结束**

# HTML—标签与格式：插入图像



开始标签

结束标签

**<IMG SRC=portrait.gif HEIGHT=100 WIDTH=65>**

插入图像

高度是 **100** 像素

宽度是 **65** 像素

插入的图像文件名是 **portrait.gif**



# HTML一标签与格式：超链接



链接到其他网点上的页面

- 定义一个超链的标签是<A>。字符A表示锚(Anchor)。
- 在HTML文档中定义一个超链的语法是：

`<A HREF="... ">X</A>`

超链的起点

这个地方填写超链终点的 URL

- 远程链接：超链的终点是其他网点上的页面。
- 本地链接：超链指向本计算机中的某个文件。

# 远程超链接例子



`<A HREF="http://www.tsinghua.edu.cn">清华大学</A>`

超链的终点是清华大学的主页

超链的起点是某个文档中的这四个字

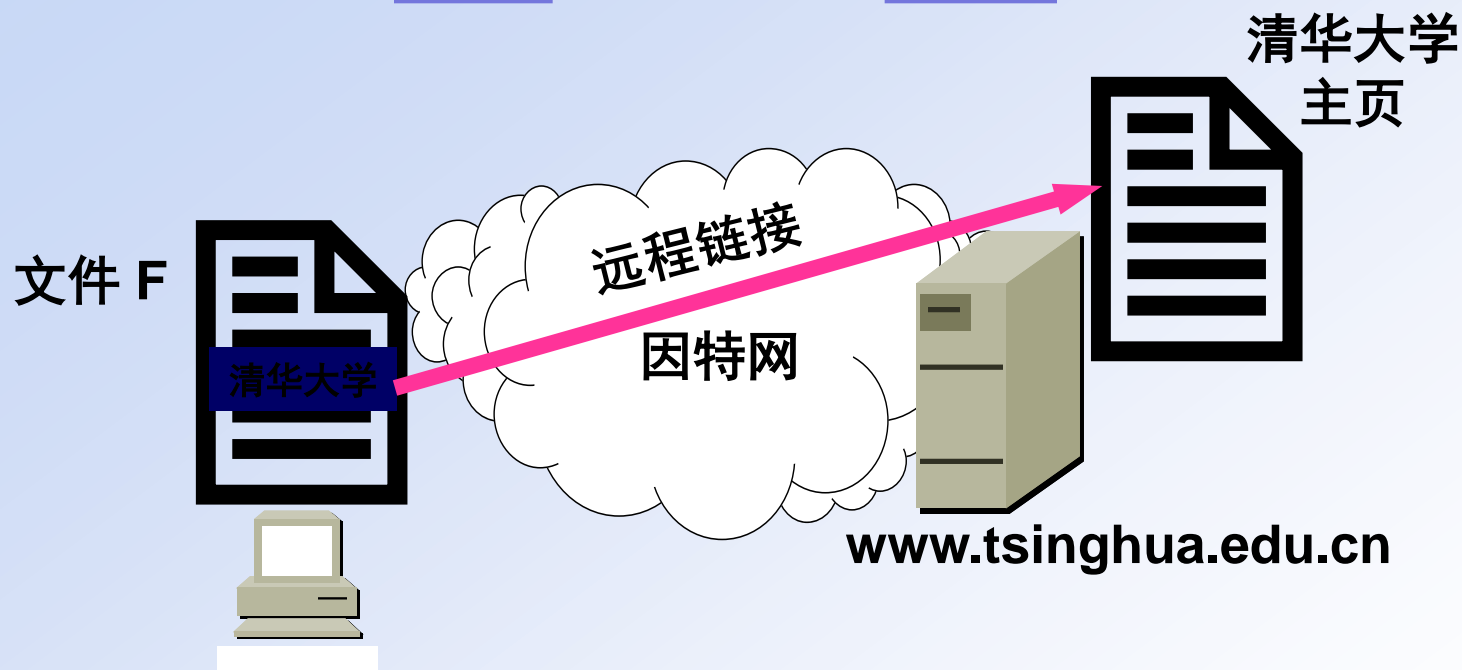
# 远程超链接例子

远程链接：

<A HREF="<http://www.tsinghua.edu.cn>">清华大学 </A>

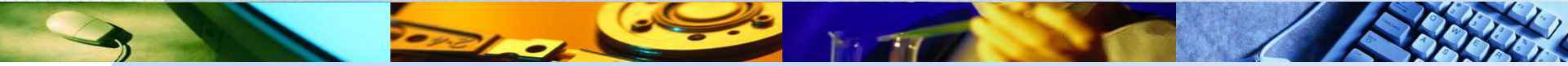
终点

起点



- 远程链接：超链的终点是其他网点上的页面。
- 本地链接：超链指向本计算机中的某个文件。

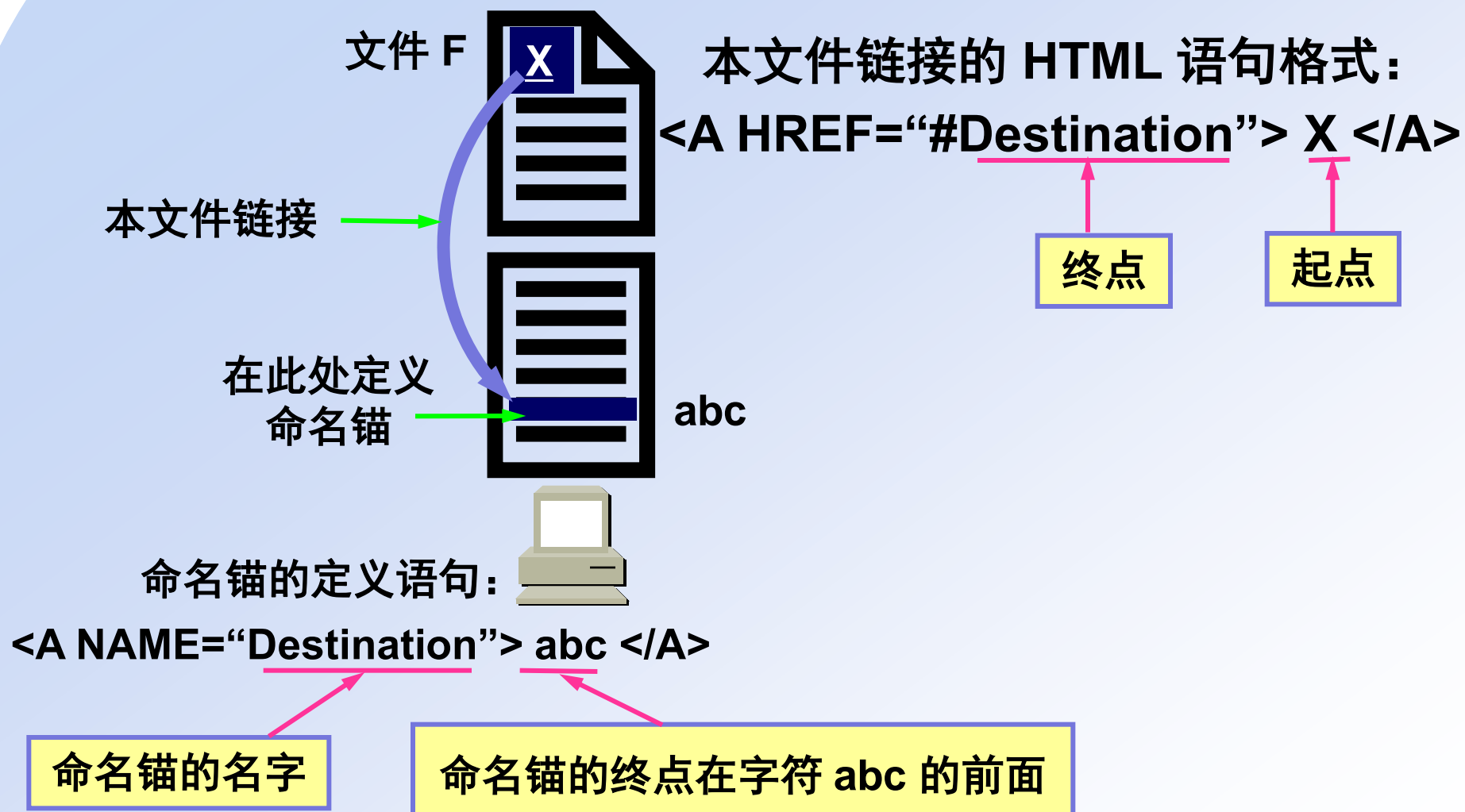
# 本地超链接



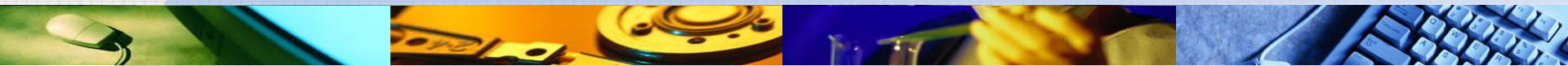
本地链接可进行许多的简化：

- 协议(http://)被省略——表明与当前页面的协议相同。
- 主机域名被省略——表明是当前的主机域名。
- 目录路径被省略——表明是当前目录（对于远程链接，表明是主机的默认根目录）。
- 文件名被省略——表明是当前文件（对于远程链接，表明是对方服务器上默认的文件名，通常是一个名为 index.html 的文件）。

# 本地超链接的例子



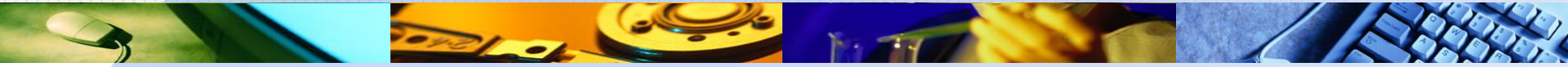
# 动态万维网文档



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

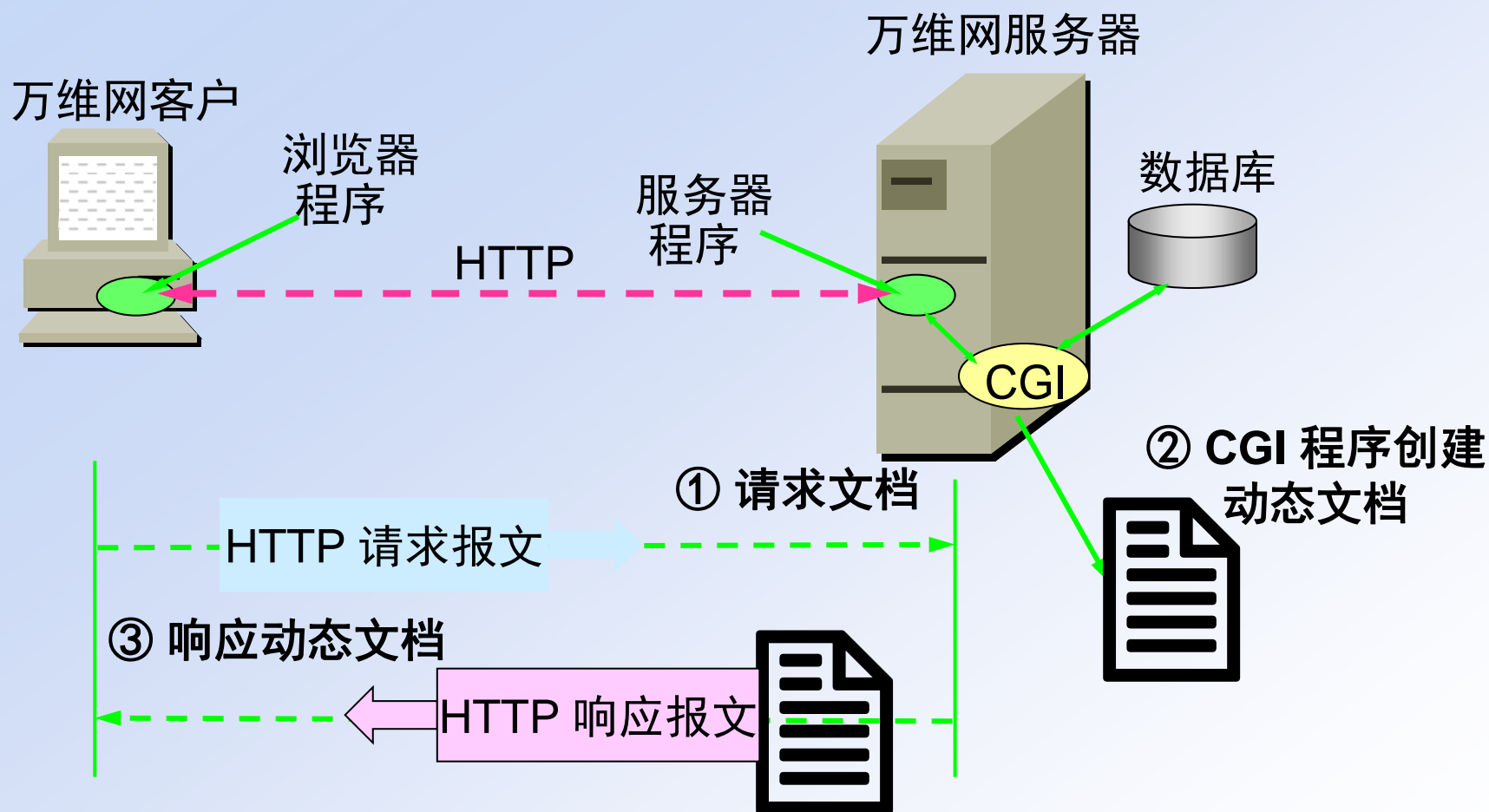


# 万维网服务器功能的扩充

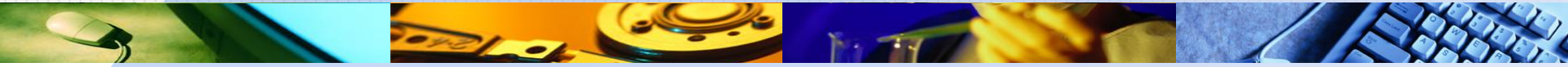


- (1) 应增加另一个应用程序，用来处理浏览器发来的数据，并创建动态文档。
- (2) 应增加一个机制，用来使万维网服务器把浏览器发来的数据传送给这个应用程序，然后万维网服务器能够解释这个应用程序的输出，并向浏览器返回 HTML 文档。

# 扩充了功能的万维网服务器



# 通用网关接口 CGI



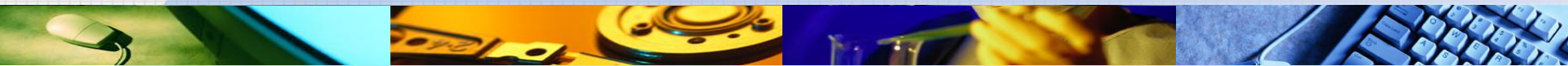
- ❖ CGI (Common Gateway Interface)
- ❖ CGI 是一种标准，它定义了动态文档应如何创建，输入数据应如何提供给应用程序，以及输出结果应如何使用。
- ❖ 万维网服务器与 CGI 的通信遵循 CGI 标准。
- ❖ “通用”：CGI 标准所定义的规则对其他任何语言都是通用的。
- ❖ “网关”：CGI 程序的作用像网关。
- ❖ “接口”：有一些已定义好的变量和调用等可供其他 CGI 程序使用。

# CGI 程序



- ❖ CGI 程序的正式名字是 CGI 脚本(script)。
- ❖ “脚本”指的是一个程序，它被另一个程序（解释程序）而不是计算机的处理机来解释或执行。
- ❖ 脚本运行起来要比一般的编译程序要慢，因为它的每一条指令先要被另一个程序来处理（这就要一些附加的指令），而不是直接被指令处理器来处理。

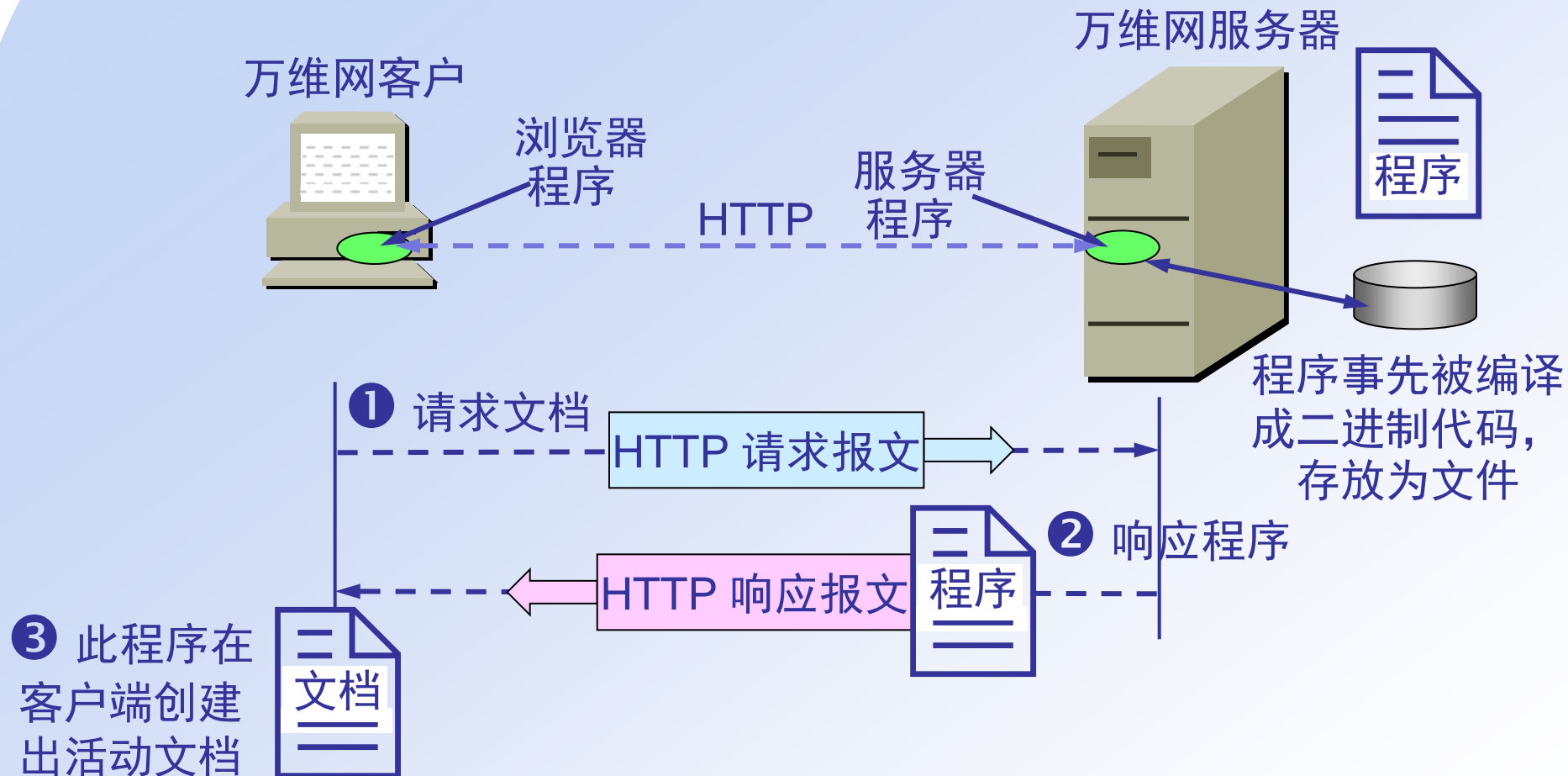
# 活动万维网文档



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



# 活动文档在客户端创建





# 用 Java 技术创建活动文档



- ❖ 由美国 Sun 公司开发的 Java 语言是一项用于创建和运行活动文档的技术。
- ❖ 在 Java 技术中使用 “小应用程序” (applet) 来描述活动文档程序。
- ❖ 用户从万维网服务器下载嵌入了 Java 小应用程序的 HTML 文档后，可在浏览器的屏幕上点击某个图像，就可看到动画效果，或在下拉式菜单中点击某个项目，就可看到计算结果。
- ❖ Java 技术是活动文档技术的一部分。

# WWW万维网



- ❖ 概述
- ❖ 统一资源定位符
- ❖ 超文本传输协议
- ❖ 万维网的文档
- ❖ **万维网的信息检索系统**
- ❖ 博客、微波和轻博

# 万维网的信息检索系统



在万维网中用来进行搜索的程序叫做搜索引擎。

❖ 全文检索搜索

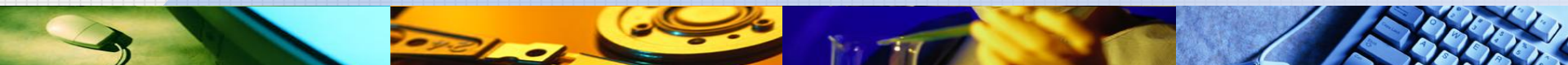
❖ 分类目录搜索

# 全文检索搜索



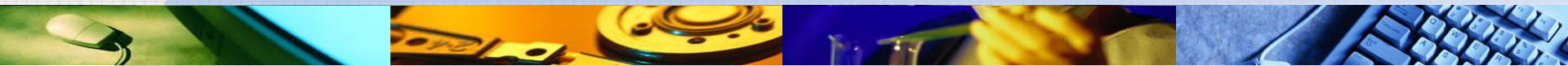
- ❖ 全文检索搜索引擎是一种纯技术型的检索工具。它的工作原理是通过搜索软件到因特网上的各网站收集信息，找到一个网站后可以从这个网站再链接到另一个网站。然后按照一定的规则建立一个很大的在线数据库供用户查询。
- ❖ 用户在查询时只要输入关键词，就从已经建立的索引数据库上进行查询（并不是实时地在因特网上检索到的信息）。

# 分类目录搜索



- ❖ 分类目录搜索引擎并不采集网站的任何信息，而是利用各网站向搜索引擎提交的网站信息时填写的关键词和网站描述等信息，经过人工审核编辑后，如果认为符合网站登录的条件，则输入到分类目录的数据库中，供网上用户查询。
- ❖ 分类目录搜索也叫做分类网站搜索。

# 一些著名的搜索引擎



## ❖ 最著名的全文检索搜索引擎:

- Google (谷歌) ([www.google.com](http://www.google.com))
- 百度 ([www.baidu.com](http://www.baidu.com))

## ❖ 最著名的分类目录搜索引擎:

- 雅虎 ([www.yahoo.com](http://www.yahoo.com))
- 雅虎中国 ([cn.yahoo.com](http://cn.yahoo.com))
- 新浪 ([www.sina.com](http://www.sina.com))
- 搜狐 ([www.sohu.com](http://www.sohu.com))
- 网易 ([www.163.com](http://www.163.com))



# 垂直搜索引擎



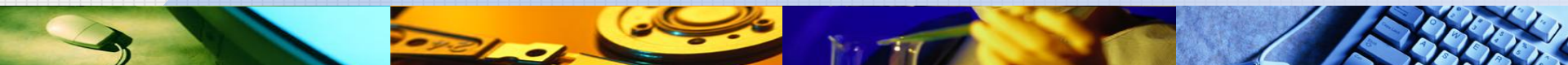
- ❖ 针对某一特定领域、特定人群或某一特定需求提供搜索服务。垂直搜索也是提供关键字来进行搜索的，但被放到了一个行业知识的上下文中，返回的结果更倾向于信息、消息、条目等。

# WWW万维网



- ❖ 概述
- ❖ 统一资源定位符
- ❖ 超文本传输协议
- ❖ 万维网的文档
- ❖ 万维网的信息检索系统
- ❖ 博客、微博和轻博

# 博客



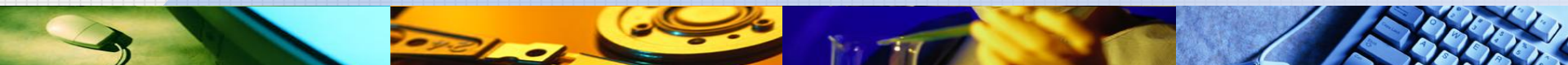
- ❖ 博客是万维网日志(web log)的简称。也有人把blog 进行音译，译为“部落格”，或“部落阁”。还有人用“博文”来表示“博客文章”。
- ❖ Weblog 这个新词是 Jorn Barger 于 1997 年创造的。
- ❖ 简写的 blog（这是今天最常用的术语）则是Peter Merholz 于 1999 年创造的。
- ❖ 有人把 blog 既当作名词，也当作动词，表示编辑博客或写博客。

## 博客（续）



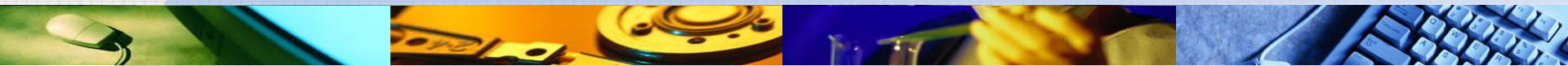
- ❖ 博客已经极大地扩充了因特网的应用和影响。
- ❖ 在博客出现以前，网民是因特网上内容的消费者。
- ❖ 但博客改变了这种情况，网民不仅是因特网上内容的消费者，而且还是因特网上内容的生产者。

# 微博



- ❖ 微博就是微型博客 (microblog)，又称为微博客。
- ❖ 微博不同于一般的博客。微博只记录片段、碎语，三言两语，现场记录，发发感慨，晒晒心情，永远只针对一个问题进行回答。
- ❖ 用户可以通过网页、WAP 网、手机短信彩信、手机客户端等多种方式更新自己的微博。
- ❖ 每条微博字数限制为 140 字，提供插入单张图片、视频地址、音乐功能。

# 轻博



- ❖ 轻博就是轻博客(light blogging)。
- ❖ 轻博客是一种介于博客和微博之间的网络服务，同样为用户提供生成内容表达自己的平台。
- ❖ 轻博可以发送博文，没有字数限制。
- ❖ 轻博发表后，其界面会好看些。
- ❖ 在轻博中，推荐与发现的内容比较丰富。



# 课堂讨论



## ❖ 讨论说明如下问题:

- 从学院账号([acl@ss.pku.edu.cn](mailto:acl@ss.pku.edu.cn))给新浪账号 ( [cal@sina.com.cn](mailto:cal@sina.com.cn))发送邮件
- 给出收发邮件服务器，用协议命名服务器
- 从应用层到网络层涉及哪些协议?

# 邮件发送问题分析：SMTP和POP3推和拉数据的协议

