



计算机网络

第 6 章 应用层

授课教师：洪锋

<http://osn.ouc.edu.cn/~hong>



第 6 章 应用层

6.4 万维网 WWW

6.4.1 概述

4.2.2 IP地址

4.3 划分子网和构造超网

6.1 DNS

6.4.2 统一资源定位符 URL

6.4.3 超文本传送协议

HTTP

6.4.5 万维网的信息检索系统

补充: Google 原理

6.5 电子邮件

6.2 文件传送协议

6.3 远程终端协议 TELNET

补充: 对等计算



应用层协议的特点

- 每个应用层协议是为了解决某一类应用问题。
 - 应用问题的解决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。
- 应用层的许多协议都是基于客户服务器方式。
 - 客户(client)和服务端(server)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。
 - 客户是服务请求方，服务器是服务提供方。

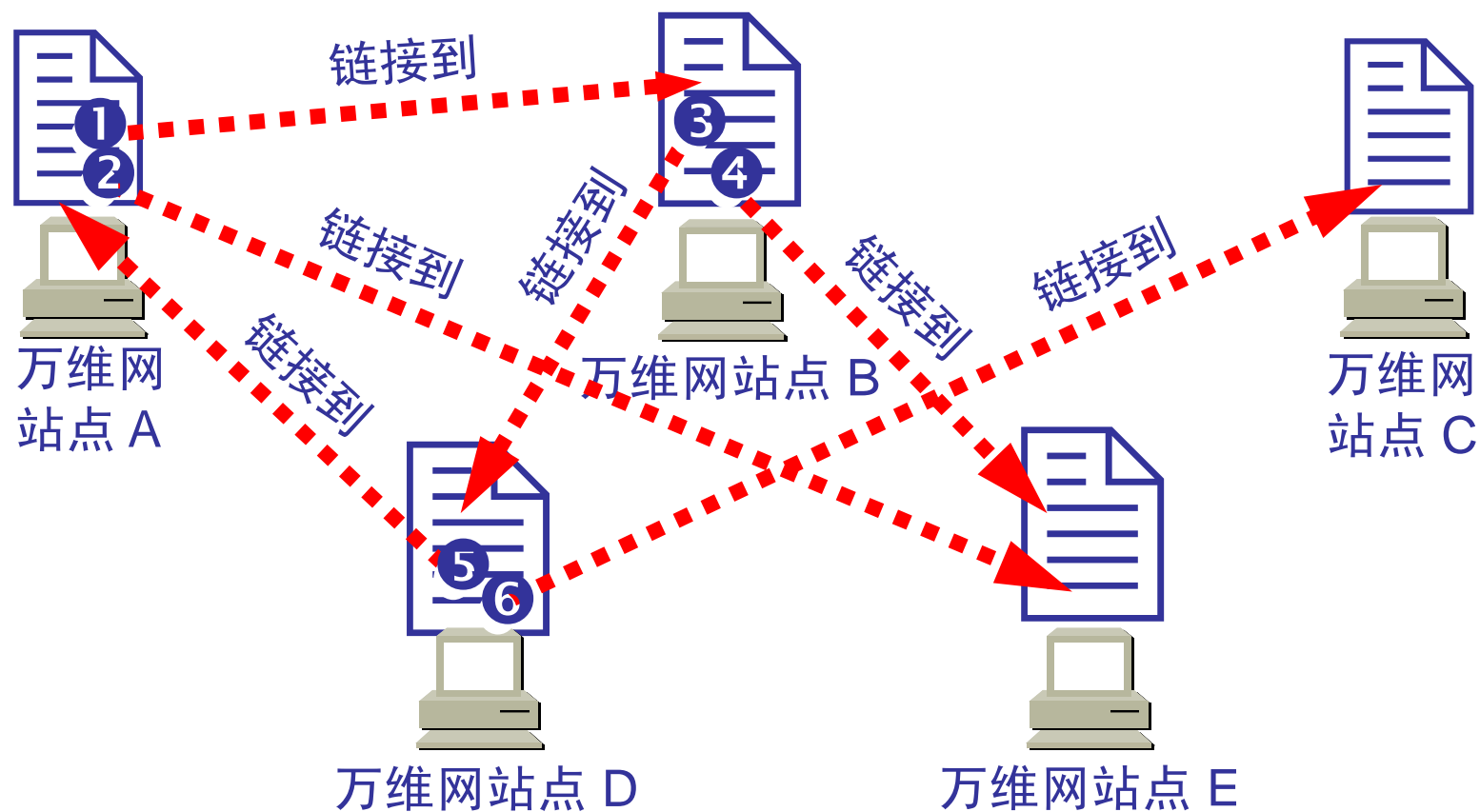


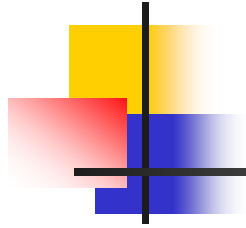
6.4 万维网 WWW

6.4.1 万维网概述

- **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**”。

万维网提供分布式服务





超媒体与超文本

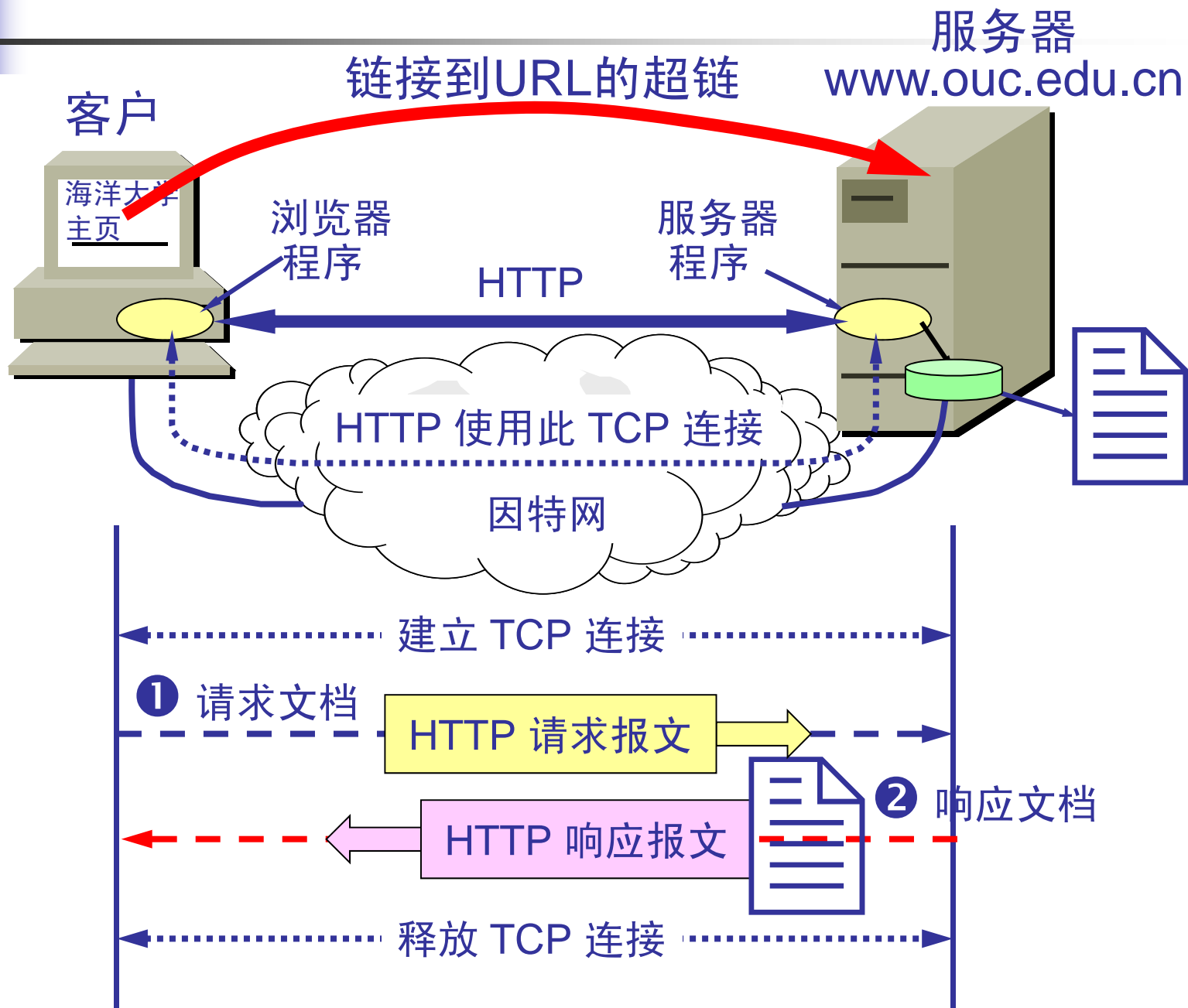
- 万维网是**分布式超媒体**(hypermedia)系统，它是**超文本**(hypertext)系统的扩充。
- 一个超文本由多个信息源链接成。
 - 利用一个链接可使用户找到另一个文档。这些文档可以位于世界上任何一个接在因特网上的超文本系统中。超文本是万维网的基础。
- **超媒体与超文本的区别是文档内容不同。**
 - 超文本文档仅包含文本信息，而超媒体文档还包含其他表示方式的信息，如图形、图像、声音、动画，甚至活动视频图像。



万维网的工作方式

- 万维网以客户服务器方式工作。
 - 浏览器就是在用户计算机上的万维网客户程序。
 - 万维网文档所驻留的计算机则运行服务器程序，因此这个计算机也称为万维网服务器。
- 客户程序向服务器程序发出请求，服务器程序向客户程序送回客户所要的万维网文档。
- 在一个客户程序主窗口上显示出的万维网文档称为页面(page)。

万维网的工作过程





万维网必须解决的问题

(1) 怎样找到目标主机？

- IP地址：IP协议保证找到Internet上任意地址对应的主机
- DNS：实现字符串域名到IP地址的映射

(2) 怎样标志分布在整个因特网上的万维网文档？

- 使用**统一资源定位符** URL (Uniform Resource Locator)来标志万维网上的各种文档。
- 使每一个文档在整个因特网的范围内具有唯一标识符 URL

(2) 用何协议实现万维网上各种超链的链接？

- 在万维网客户程序与万维网服务器程序之间进行交互所使用的协议，是**超文本传送协议** HTTP (HyperText Transfer Protocol)。
- HTTP 是一个应用层协议，使用 TCP 连接进行可靠的传送。



万维网必须解决的问题

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

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

(4) 怎样使用户能够很方便地找到所需的信息？

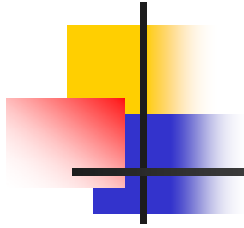
- 为了在万维网上方便地查找信息，用户可使用各种的搜索工具（即搜索引擎）。



4.2.2 分类的 IP 地址

1. IP 地址及其表示方法

- 我们把整个因特网看成为一个单一的、抽象的网络。IP 地址就是给每个连接在因特网上的主机（或路由器）分配一个在全世界范围是唯一的 32 位的标识符。
- IP 地址现在由因特网名字与号码指派公司 ICANN (Internet Corporation for Assigned Names and Numbers) 进行分配。



IP 地址的编址方法

- **分类的 IP 地址**。这是最基本的编址方法，在 1981 年就通过了相应的标准协议。
- **子网的划分**。这是对最基本的编址方法的改进，其标准[RFC 950]在 1985 年通过。
- **构成超网**。这是比较新的无分类编址方法。1993 年提出后很快就得到推广应用。



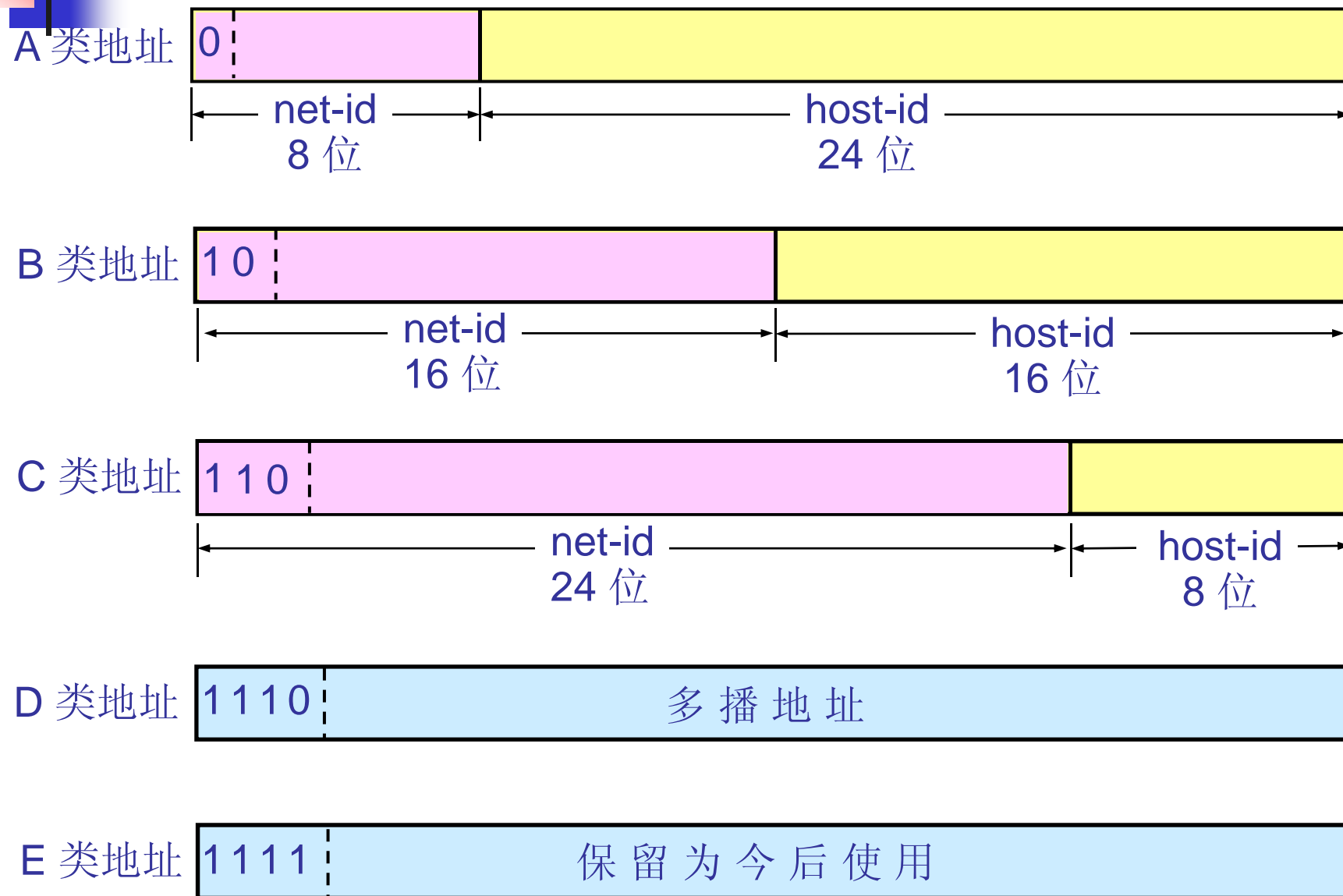
分类 IP 地址

- 每一类地址都由两个固定长度的字段组成，其中一个字段是**网络号 net-id**，它标志主机（或路由器）所连接到的网络，而另一个字段则是**主机号 host-id**，它标志该主机（或路由器）。
- 两级的 IP 地址可以记为：

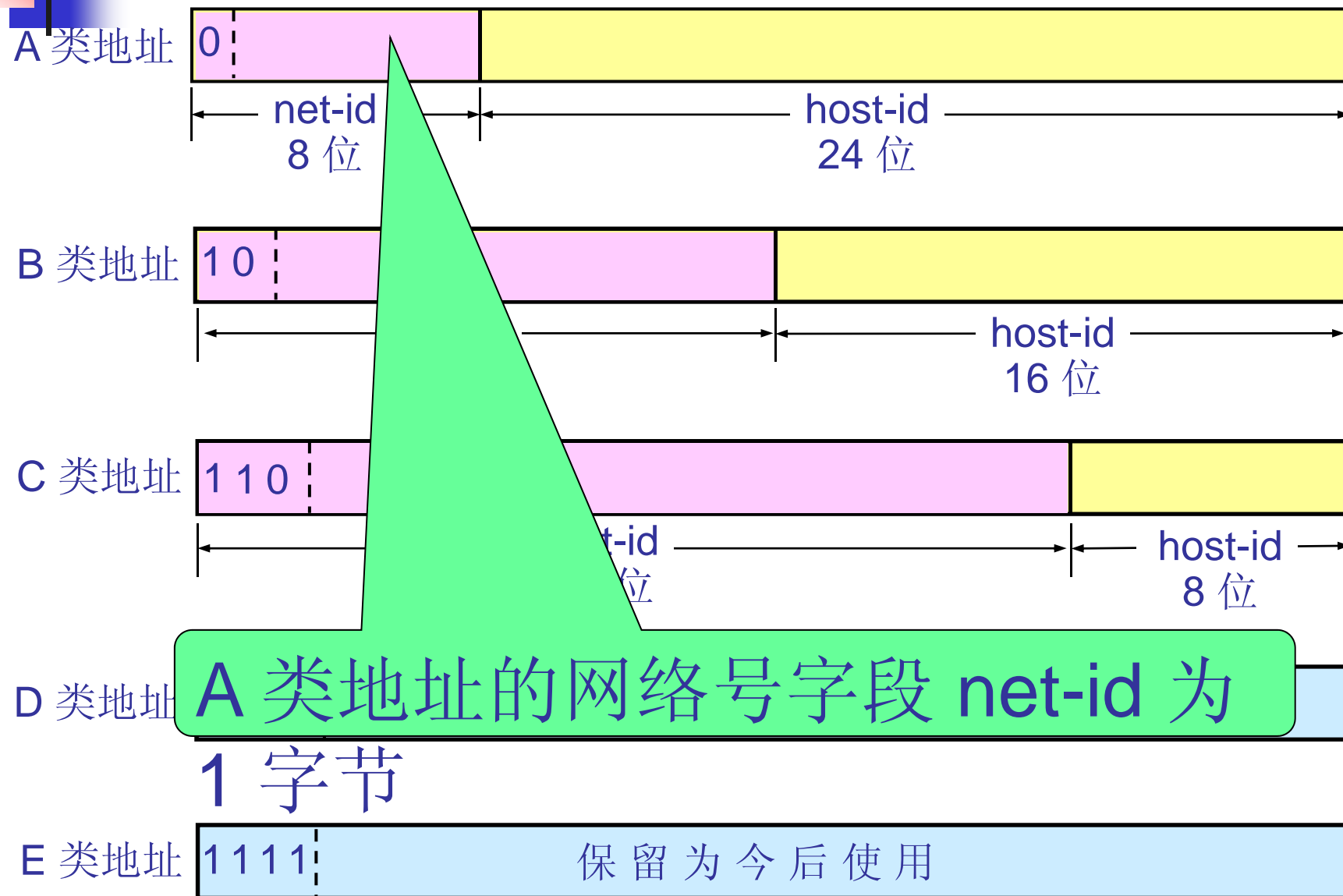
IP 地址 ::= { <网络号>, <主机号> } (4-1)

::= 代表 “**定义为**”

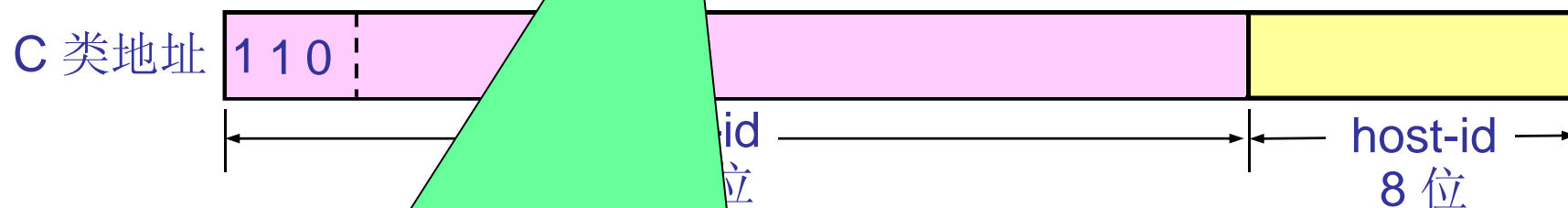
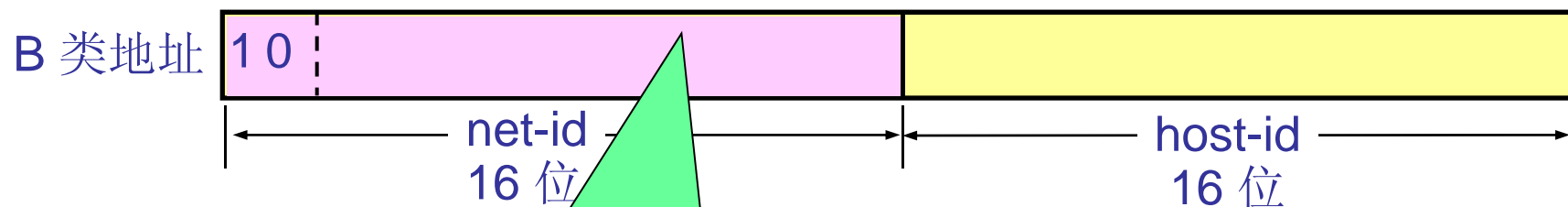
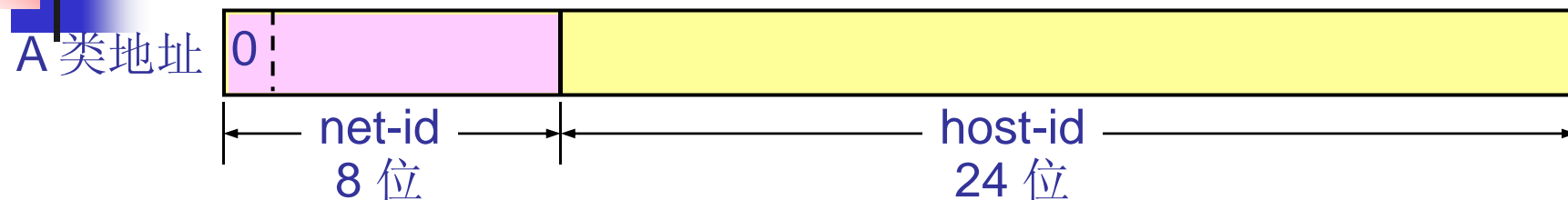
IP 地址中的网络号字段和主机号字段



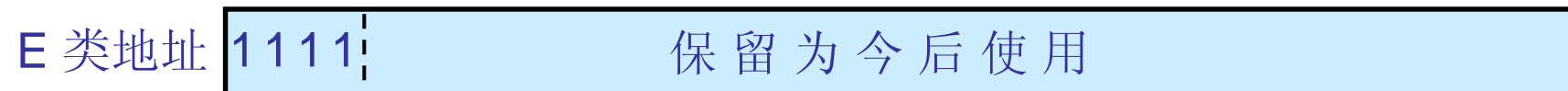
IP 地址中的网络号字段和主机号字段



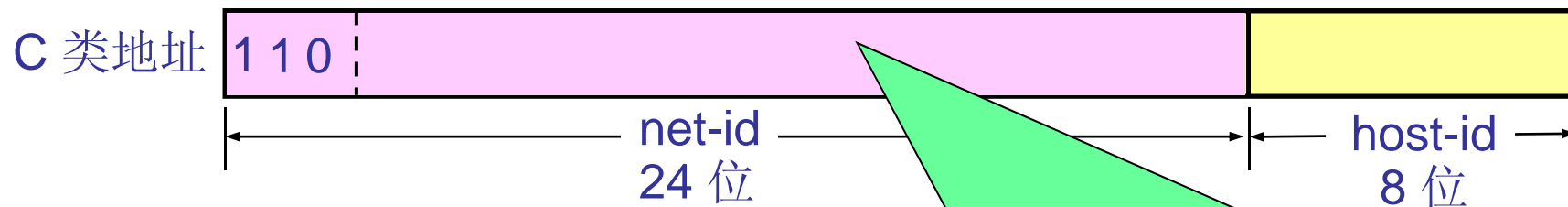
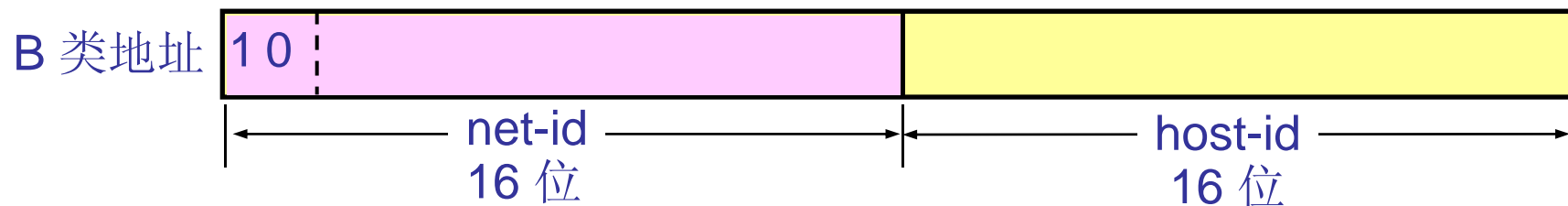
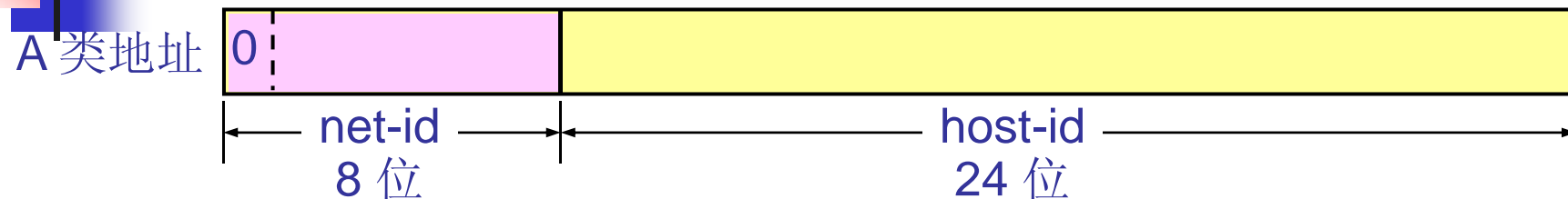
IP 地址中的网络号字段和主机号字段



D 类地址 **B 类地址的网络号字段 net-id 为 2 字节**

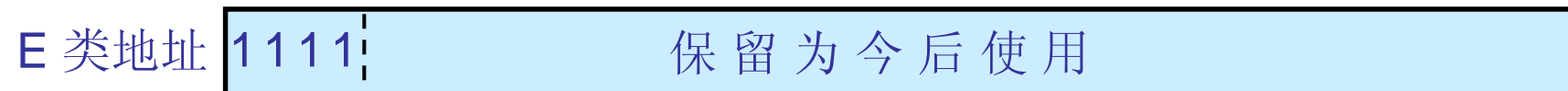


IP 地址中的网络号字段和主机号字段

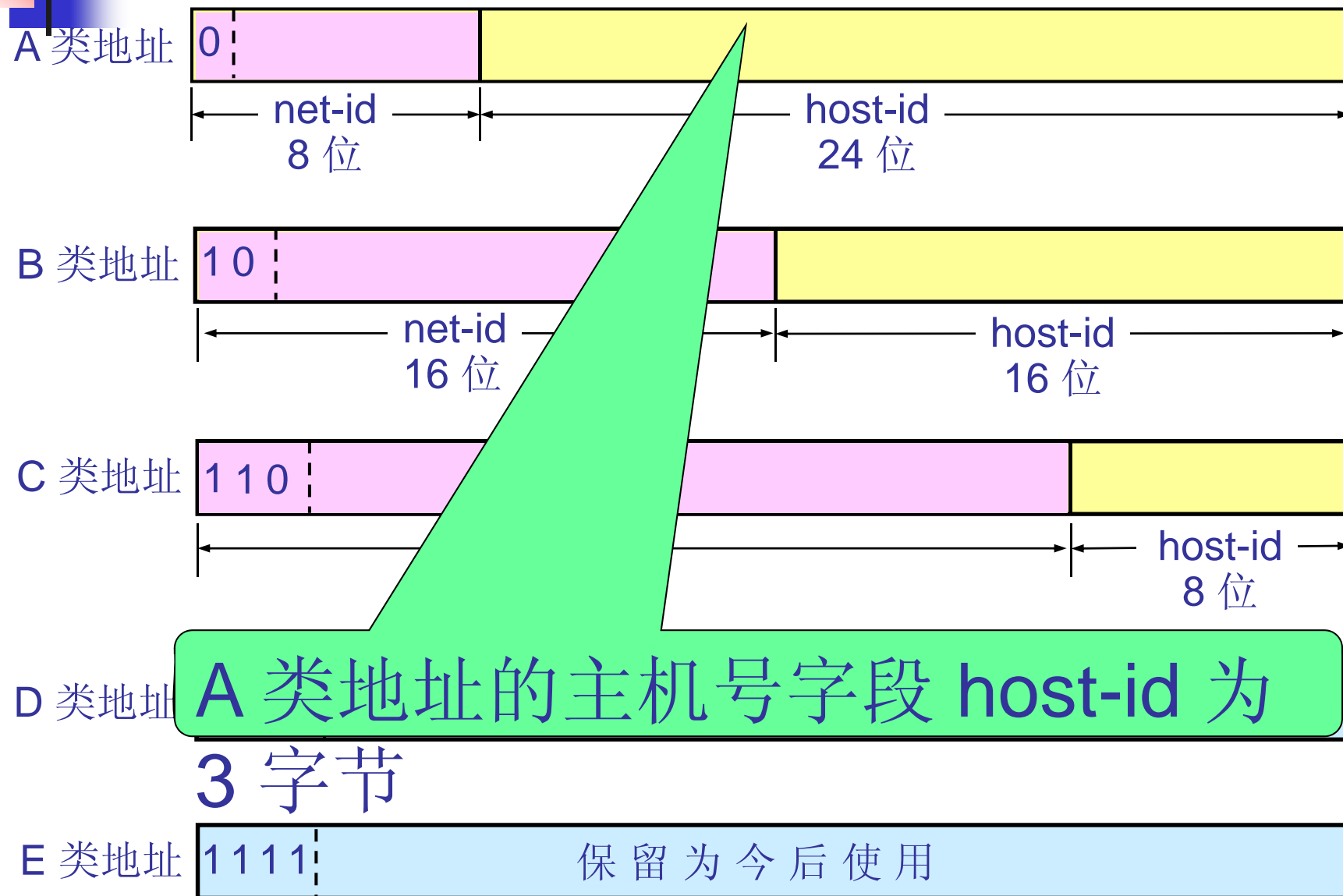


D 类地址

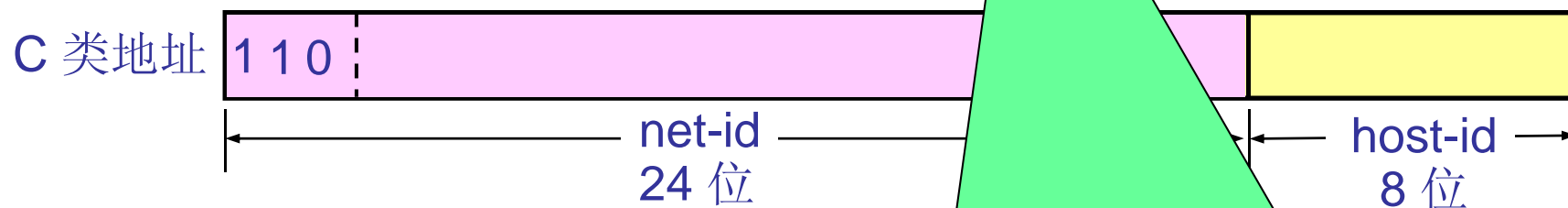
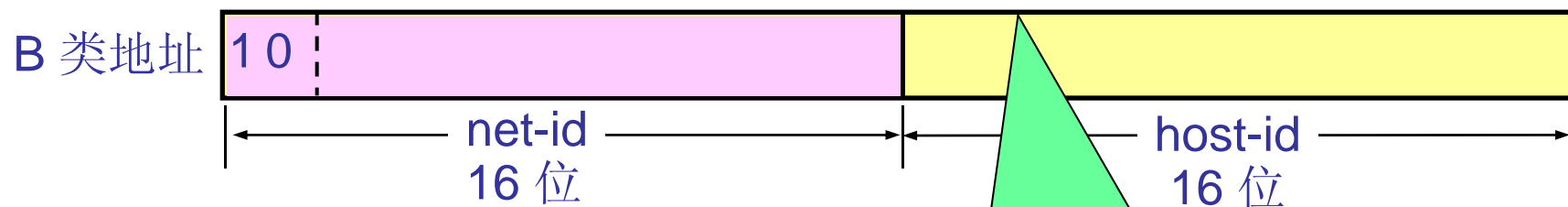
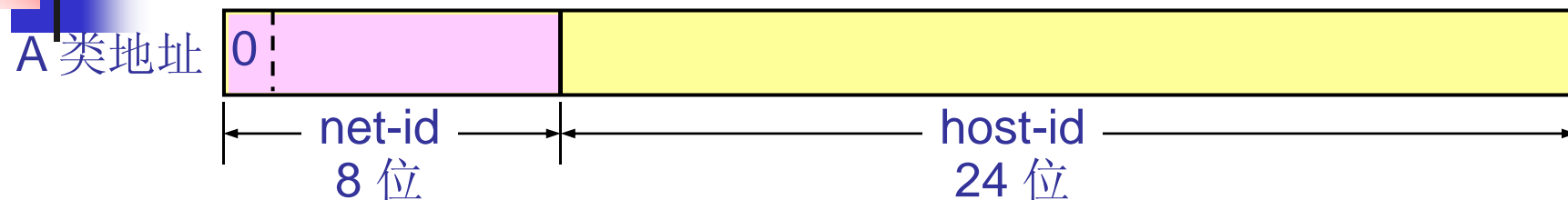
C 类地址的网络号字段 net-id 为 3 字节



IP 地址中的网络号字段和主机号字段

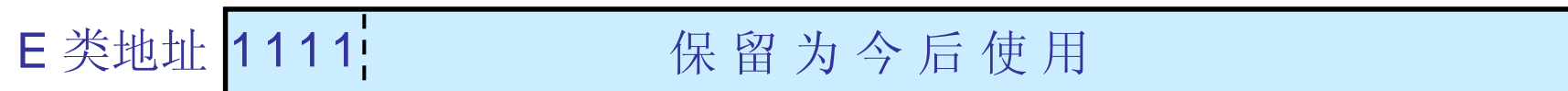


IP 地址中的网络号字段和主机号字段

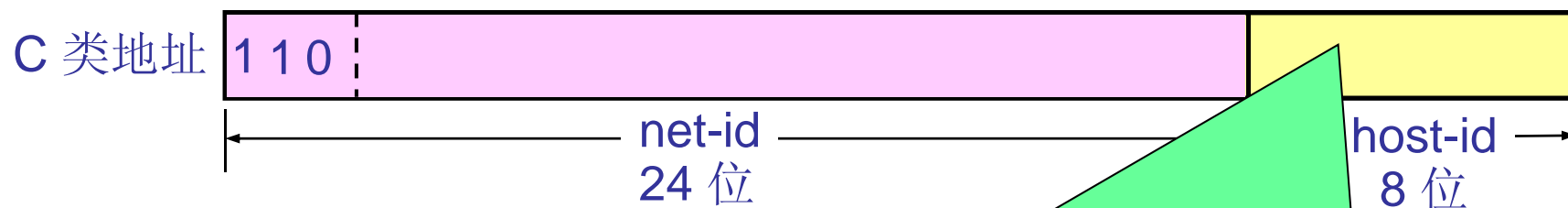
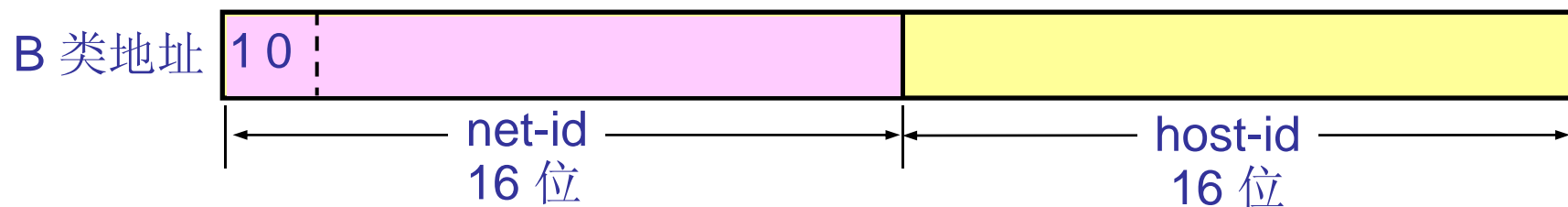
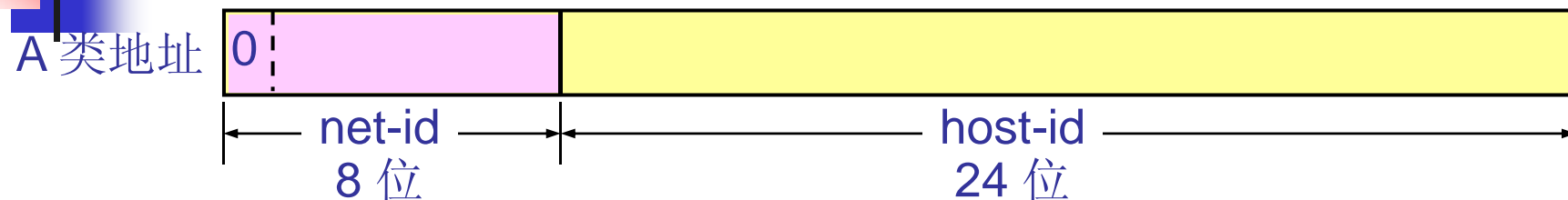


D 类地址

B 类地址的主机号字段 host-id 为 2 字节

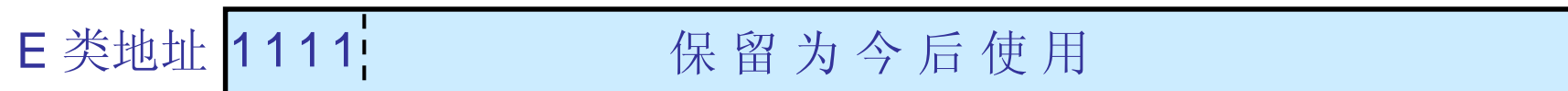


IP 地址中的网络号字段和主机号字段

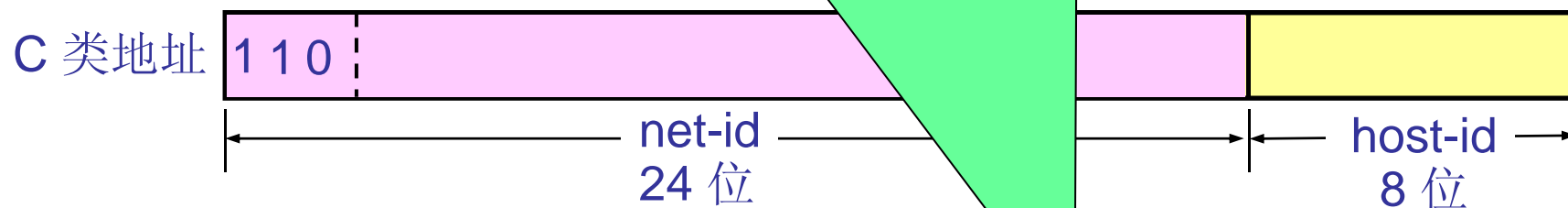
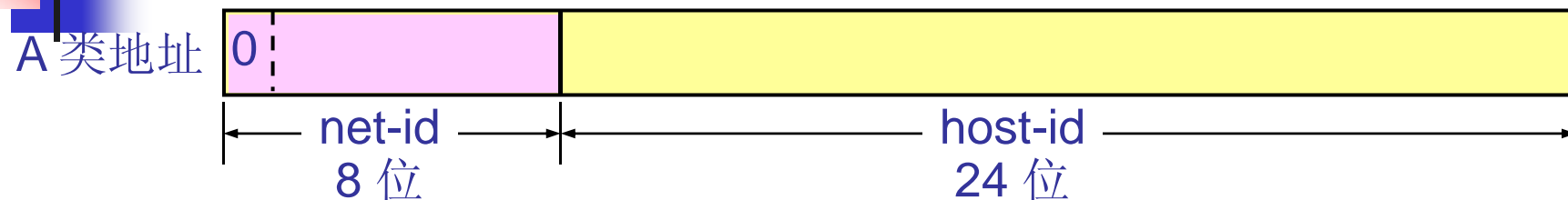


D 类地址

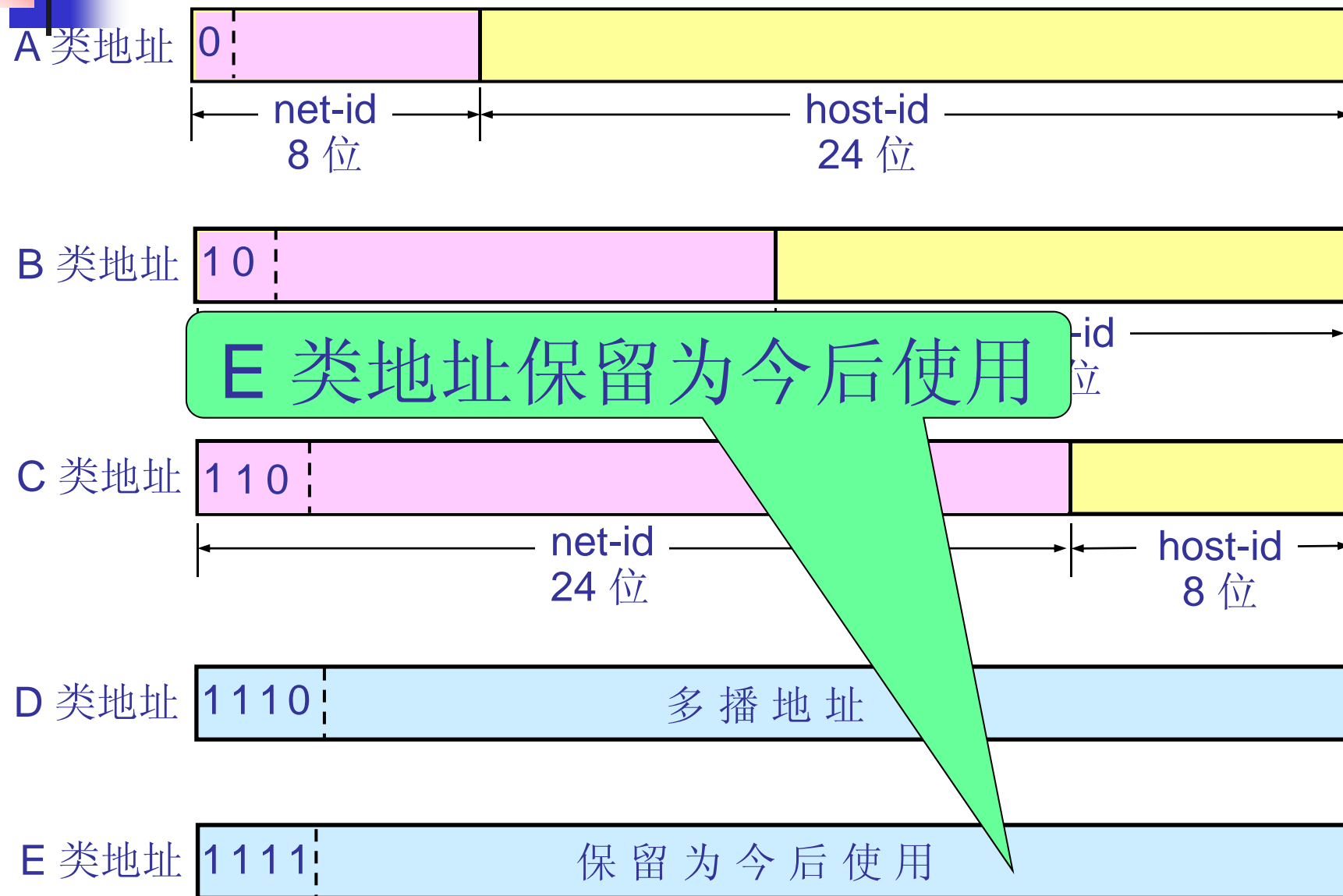
C 类地址的主机号字段 host-id 为 1 字节



IP 地址中的网络号字段和主机号字段



IP 地址中的网络号字段和主机号字段



点分十进制记法

机器中存放的 IP 地址
是 32 位 二进制代码

10000000000010110000001100011111

每隔 8 位插入一个空格
能够提高可读性

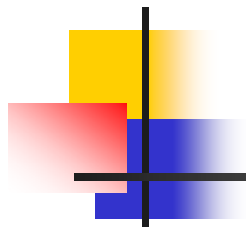
10000000 00001011 00000011 00011111

将每 8 位的二进制数
转换为十进制数

128 11 3 31

采用点分十进制记法
则进一步提高可读性

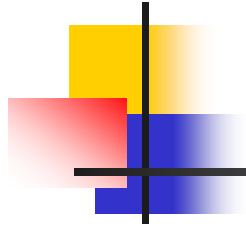
128.11.3.31



2. 常用的三种类别的 IP 地址

IP 地址的使用范围

| 网络类别 | 最大网络数 | 第一个可用的网络号 | 最后一个可用的网络号 | 每个网络中最大的主机数 |
|------|----------------------------|-----------|-------------|-------------|
| A | 126 ($2^7 - 2$) | 1 | 126 | 16,777,214 |
| B | 16,383($2^{14} - 1$) | 128.1 | 191.255 | 65,534 |
| C | 2,097,151 ($2^{21} - 1$) | 192.0.1 | 223.255.255 | 254 |



IP 地址的一些重要特点

(1) IP 地址是一种分等级的地址结构。分两个等级的好处:

- IP地址管理机构在分配 IP 地址时只分配网络号，而剩下的主机号则由得到该网络号的单位自行分配。这样就方便了 IP 地址的管理。
- 路由器仅根据目的主机所连接的网络号来转发分组（而不考虑目的主机号），这样就可以使路由表中的项目数大幅度减少，从而减小了路由表所占的存储空间。

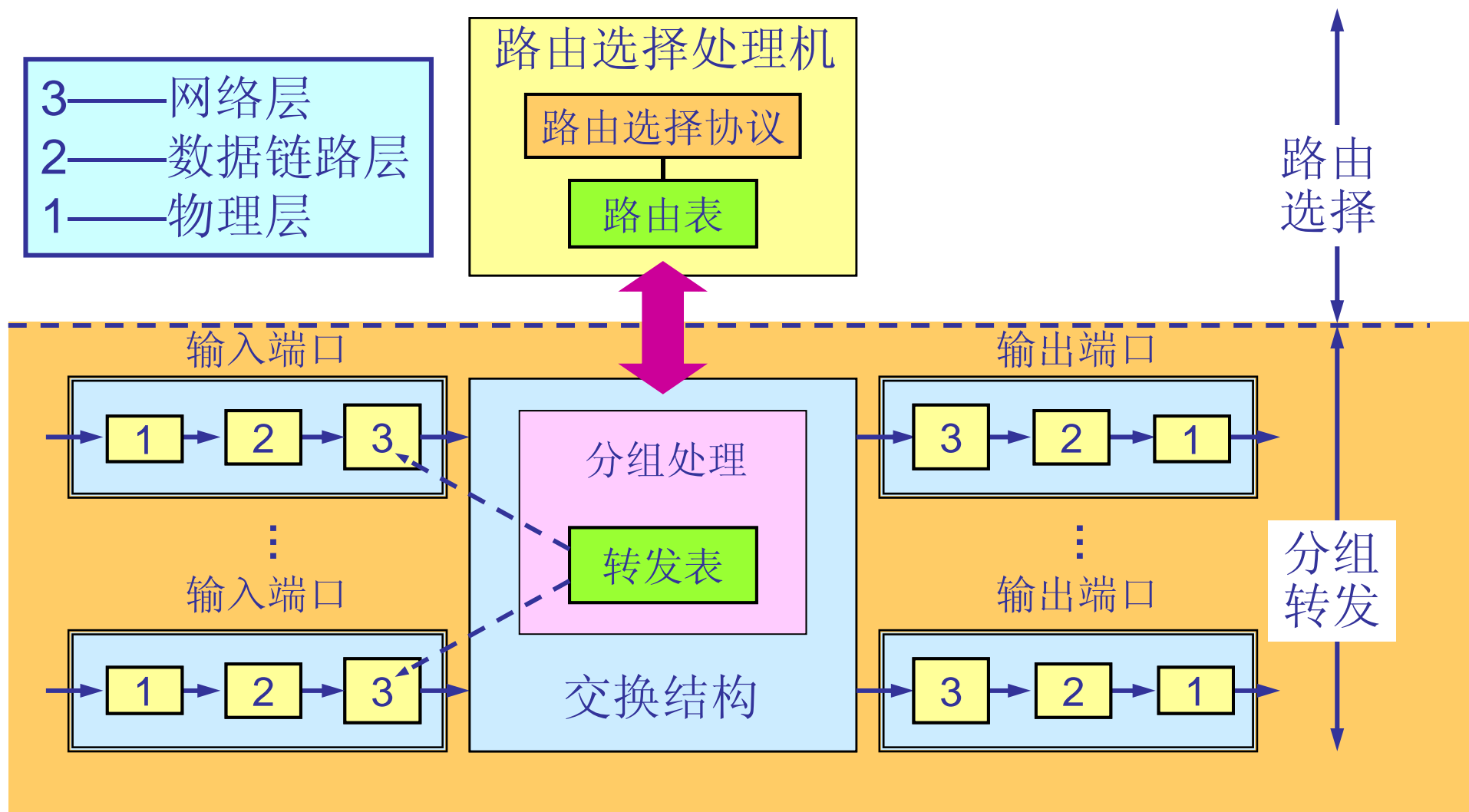


4.5.6 路由器在网际互连中的作用

1. 路由器的结构

- 路由器是一种具有多个输入端口和多个输出端口的专用计算机，其任务是转发分组。
- 路由器将某个输入端口收到的分组，按照分组要去的目的地（即目的网络），把该分组从路由器的某个合适的输出端口转发给下一跳路由器。
- 下一跳路由器也按照这种方法处理分组，直到该分组到达终点为止。

典型的路由器的结构



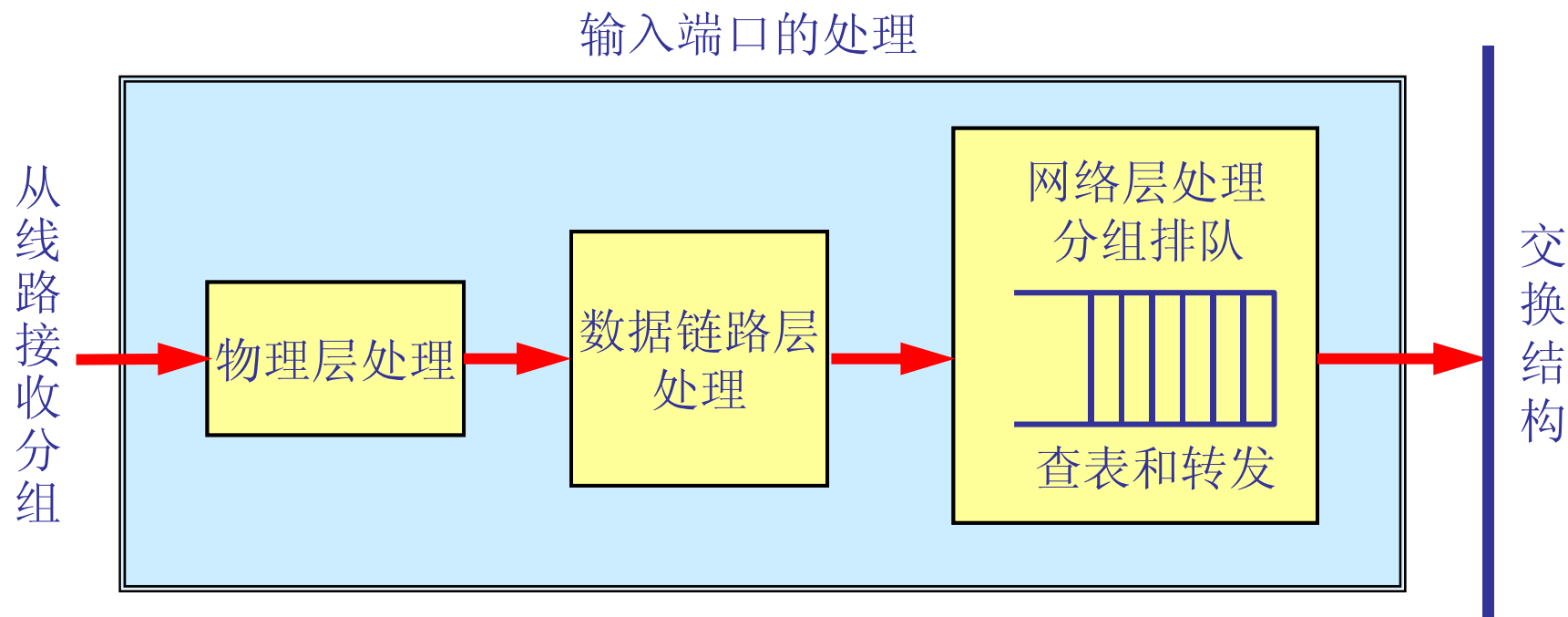


“转发”和“路由选择”的区别

- “**转发**” (forwarding)就是路由器根据转发表将用户的 IP 数据报从合适的端口转发出去。
- “**路由选择**” (routing)则是按照分布式算法，根据从各相邻路由器得到的关于网络拓扑的变化情况，动态地改变所选择的路由。
- 路由表是根据路由选择算法得出的。而转发表是从路由表得出的。
- 在讨论路由选择的原理时，往往不去区分转发表和路由表的区别，

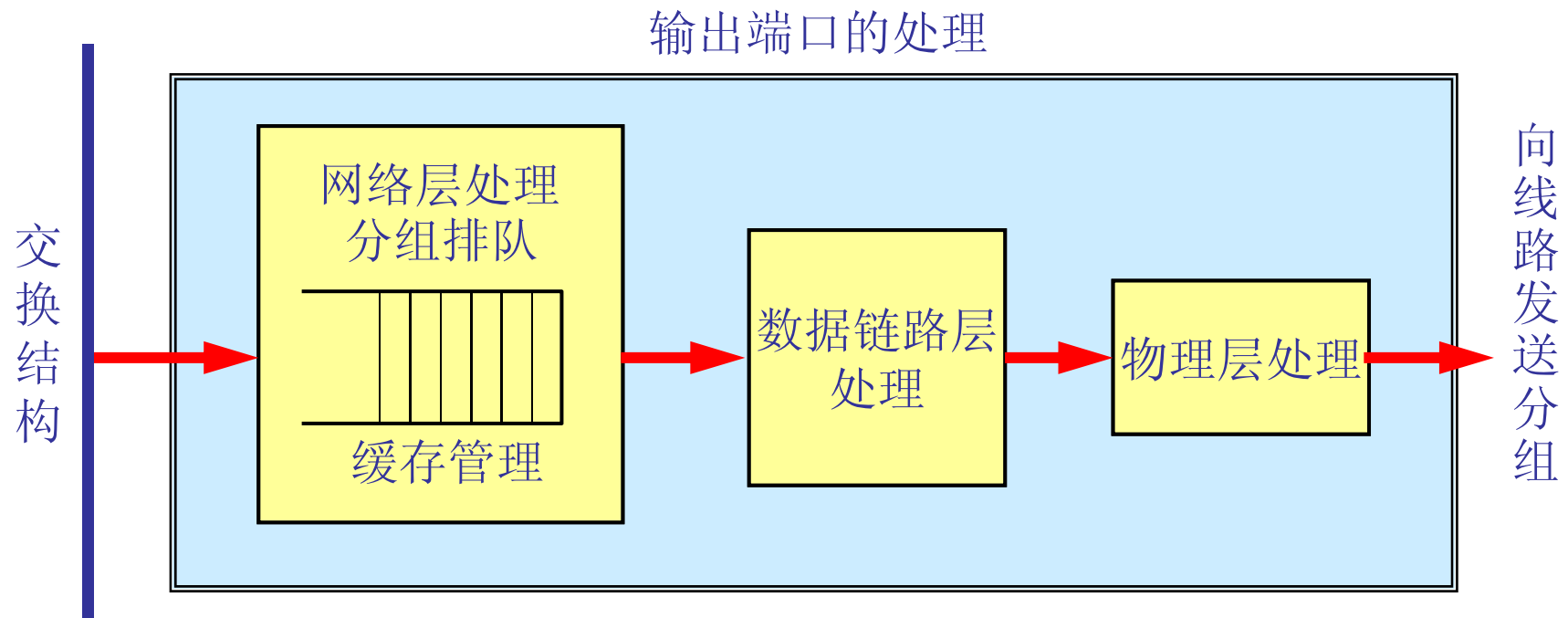
输入端口对线路上收到的分组的处理

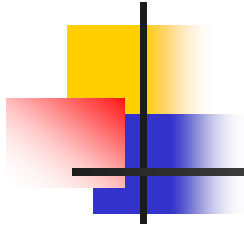
- 数据链路层剥去帧首部和尾部后，将分组送到网络层的队列中排队等待处理。这会产生一定的时延。



输出端口将交换结构传送来的分组发送到线路

- 当交换结构传送过来的分组先进行缓存。数据链路层处理模块将分组加上链路层的首部和尾部，交给物理层后发送到外部线路。

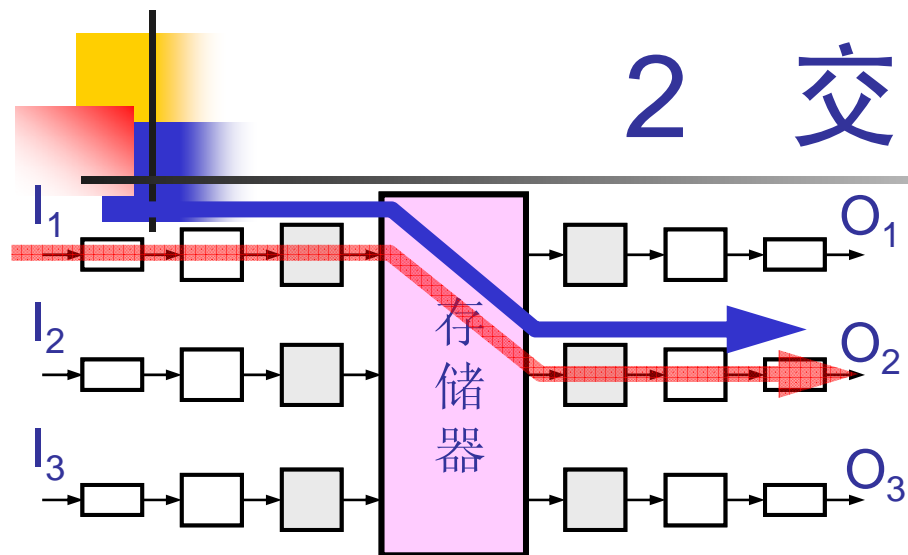




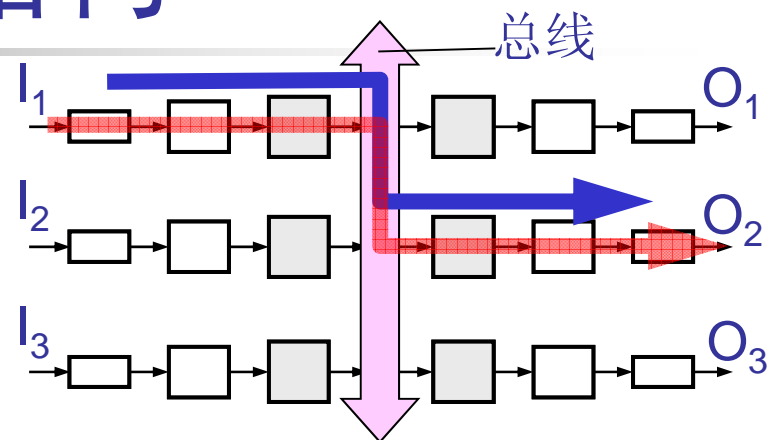
分组丢弃

- 若路由器处理分组的速率赶不上分组进入队列的速率，则队列的存储空间最终必定减少到零，这就使后面再进入队列的分组由于没有存储空间而只能被丢弃。
- 路由器中的输入或输出队列产生溢出是造成分组丢失的重要原因。

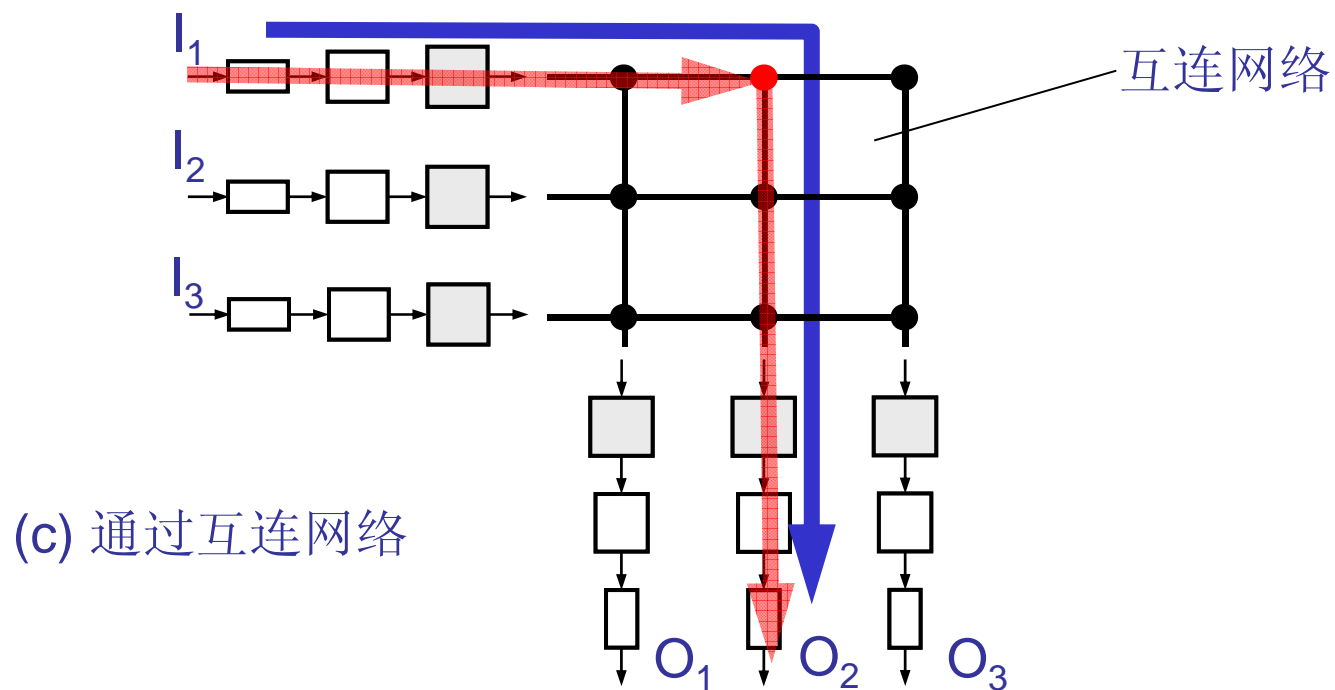
2 交换结构



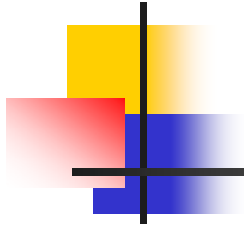
(a) 通过存储器



(b) 通过总线



(c) 通过互连网络



6.1 域名系统 DNS

6.1.1 域名系统概述

- 许多应用层软件经常直接使用**域名系统** DNS (Domain Name System)，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。
- 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**。



6.1.2 因特网的域名结构

- 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**。
- 域名的结构由标号序列组成，各标号之间用**点**隔开：

... . 三级域名 . 二级域名 . 顶级域名

- 各标号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。
 - 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。



顶级域名 TLD (Top Level Domain)

- 国家顶级域名 nTLD

- 如: .cn 表示中国, .us 表示美国, .uk 表示英国, 等等。

- 通用顶级域名 gTLD

- 早期顶级域名

.com (公司和企业)
.net (网络服务机构)
.org (非赢利性组织)
.edu (美国专用的教育机构)
.gov (美国专用的政府部门)
.mil (美国专用的军事部门)
.int (国际组织)

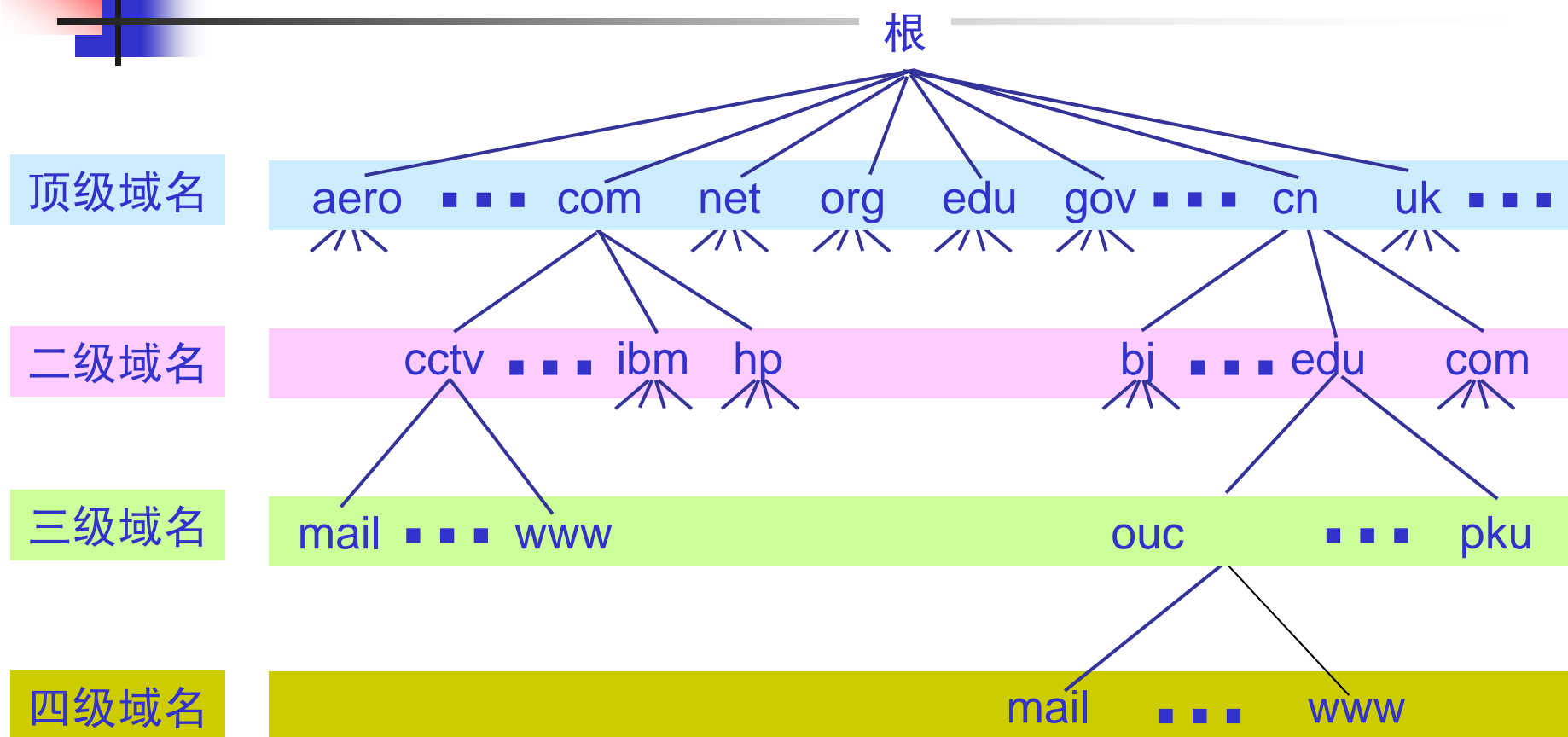
后期添加的顶级域名

.aero (航空运输企业)
.biz (公司和企业)
.cat (加泰隆人的语言和文化团体)
.coop (合作团体)
.info (信息服务)
.jobs (人力资源管理者)
.mobi (移动产品与服务的用户和提供者)
.museum (博物馆)
.name (个人)
.pro (有证书的专业人员)
.travel (旅游业)

- 基础结构域名(infrastructure domain)

- 这种顶级域名只有一个, 即 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

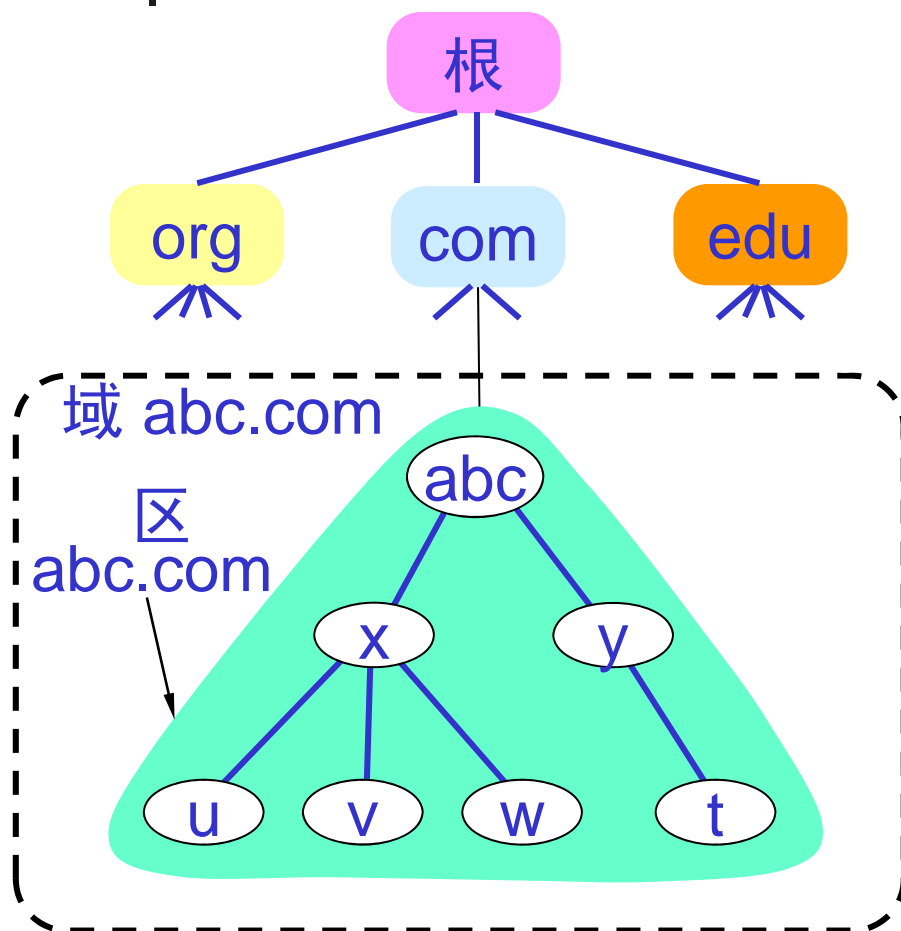




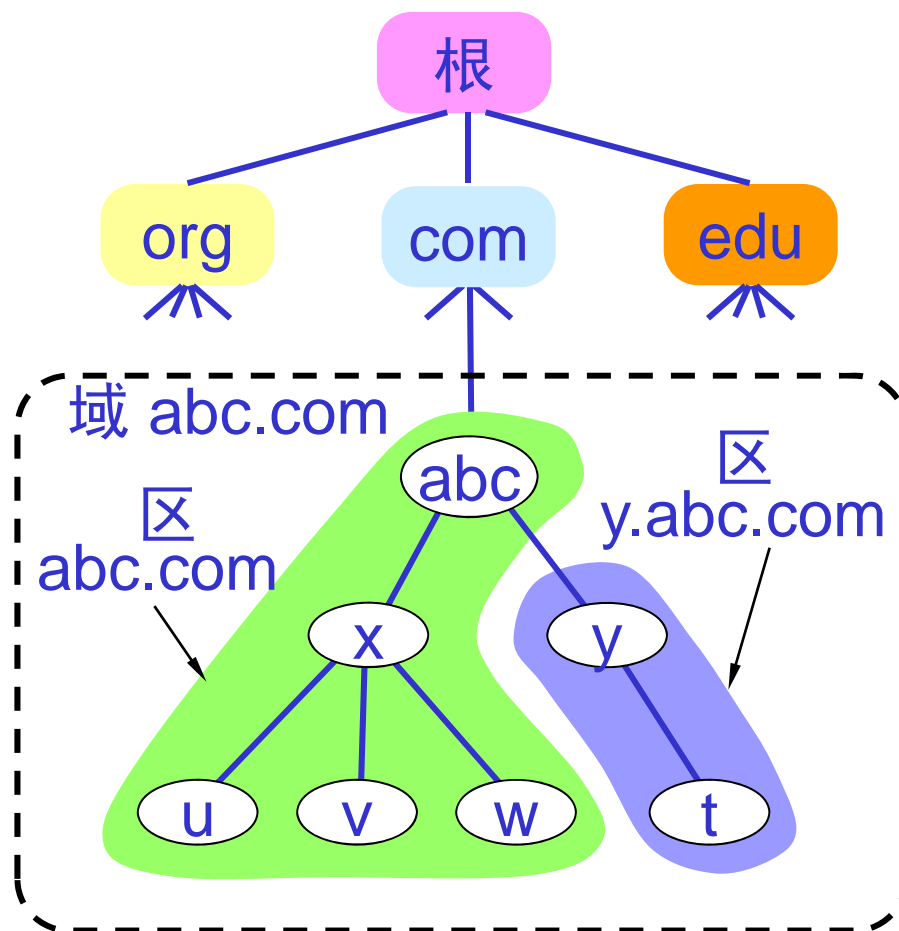
6.1.3 域名服务器

- 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**
 - 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

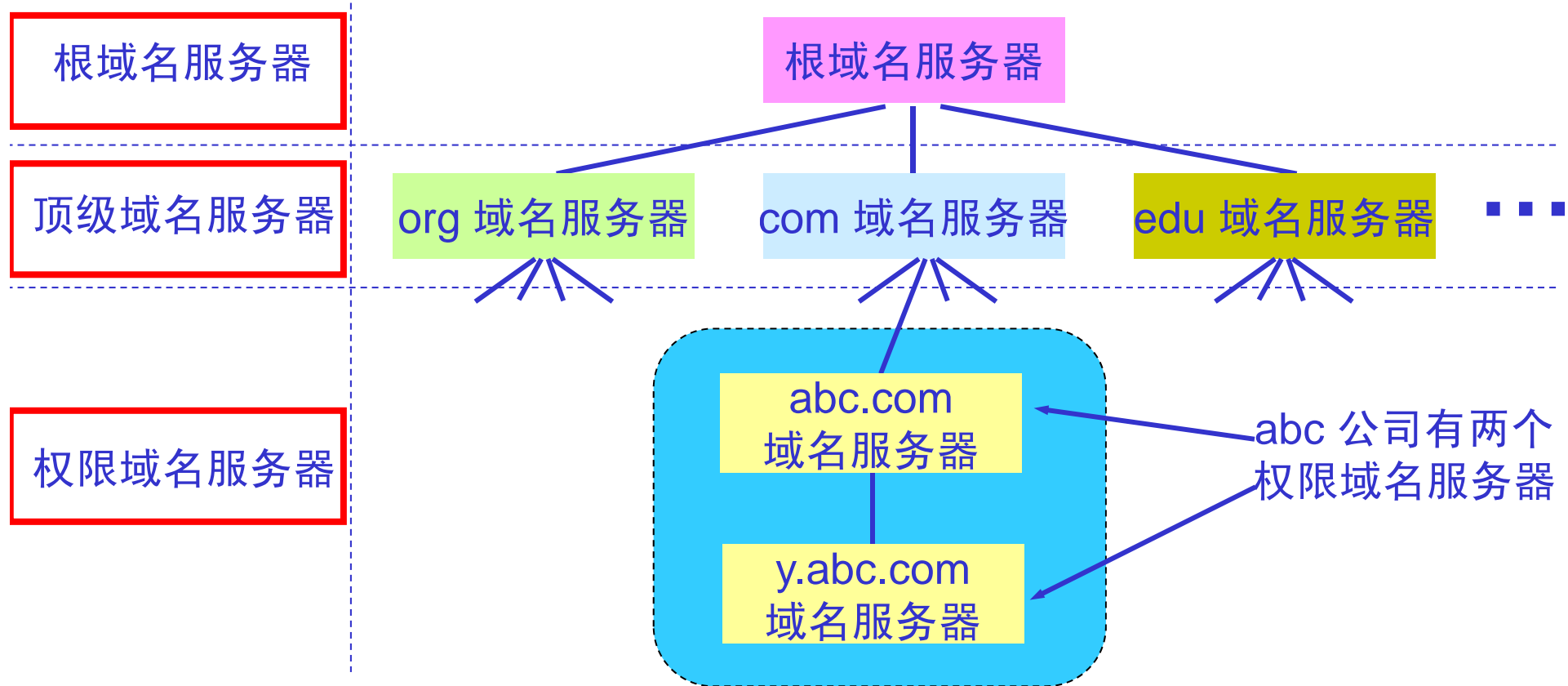


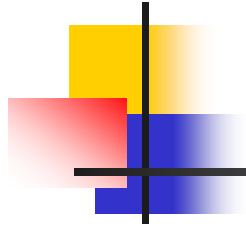
(a) 区 = 域



(b) 区 < 域

树状结构的 DNS 域名服务器





根域名服务器

——最高层次的域名服务器——

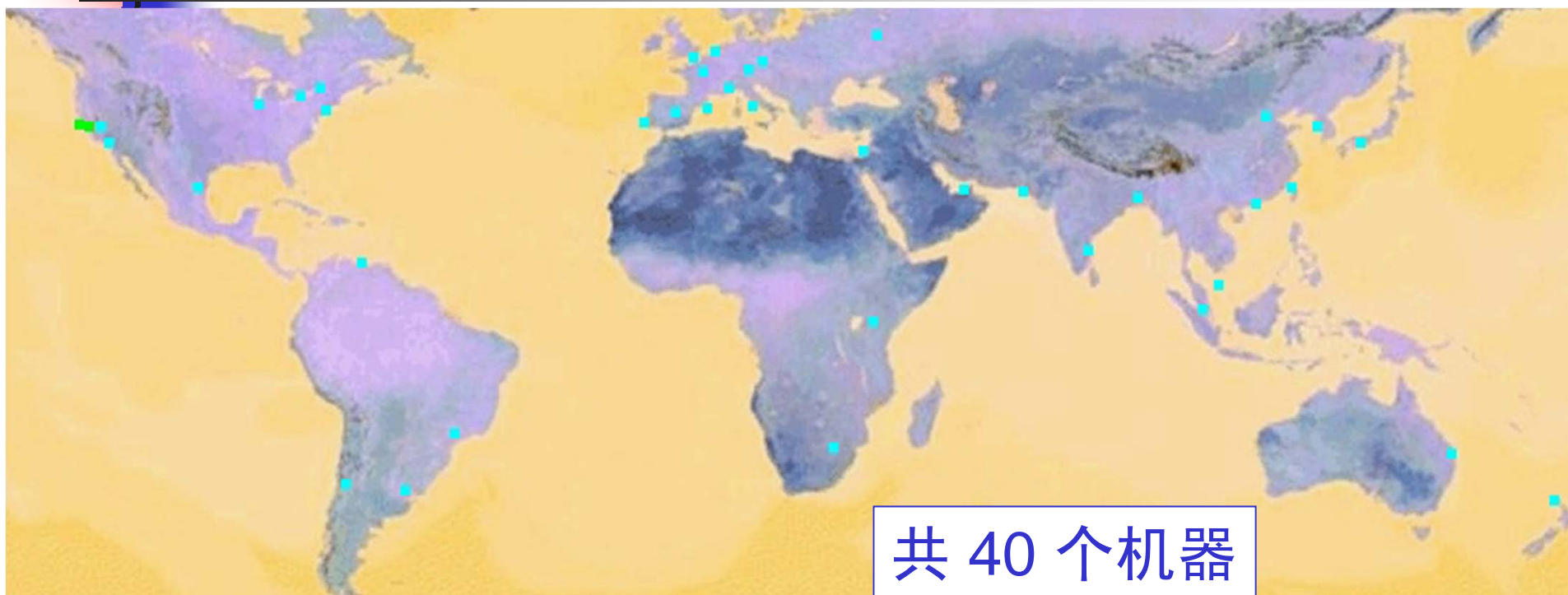
- 所有的根域名服务器都知道所有的顶级域名服务器的域名和 IP 地址。
 - 根域名服务器是最重要的域名服务器。
- 不管是哪一个本地域名服务器，若要对因特网上任何一个域名进行解析，只要自己无法解析，就首先求助于根域名服务器。
- 在因特网上共有13个不同IP地址的根域名服务器，它们的名字是用一个英文字母命名，从a一直到m（前13个字母）。



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 2006 年底全世界已经安装了一百多个根域名服务器机器，分布在世界各地。
 - 为了方便用户，使世界上大部分 DNS 域名服务器都能就近找到一个根域名服务器。

举例：根域名服务器 f 的地点分布图



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



顶级和权限域名服务器

■ 顶级域名服务器

- 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，大多数情况是下一步应当找的域名服务器的 IP 地址。

■ 权限域名服务器

- 负责一个区的域名服务器。
- 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

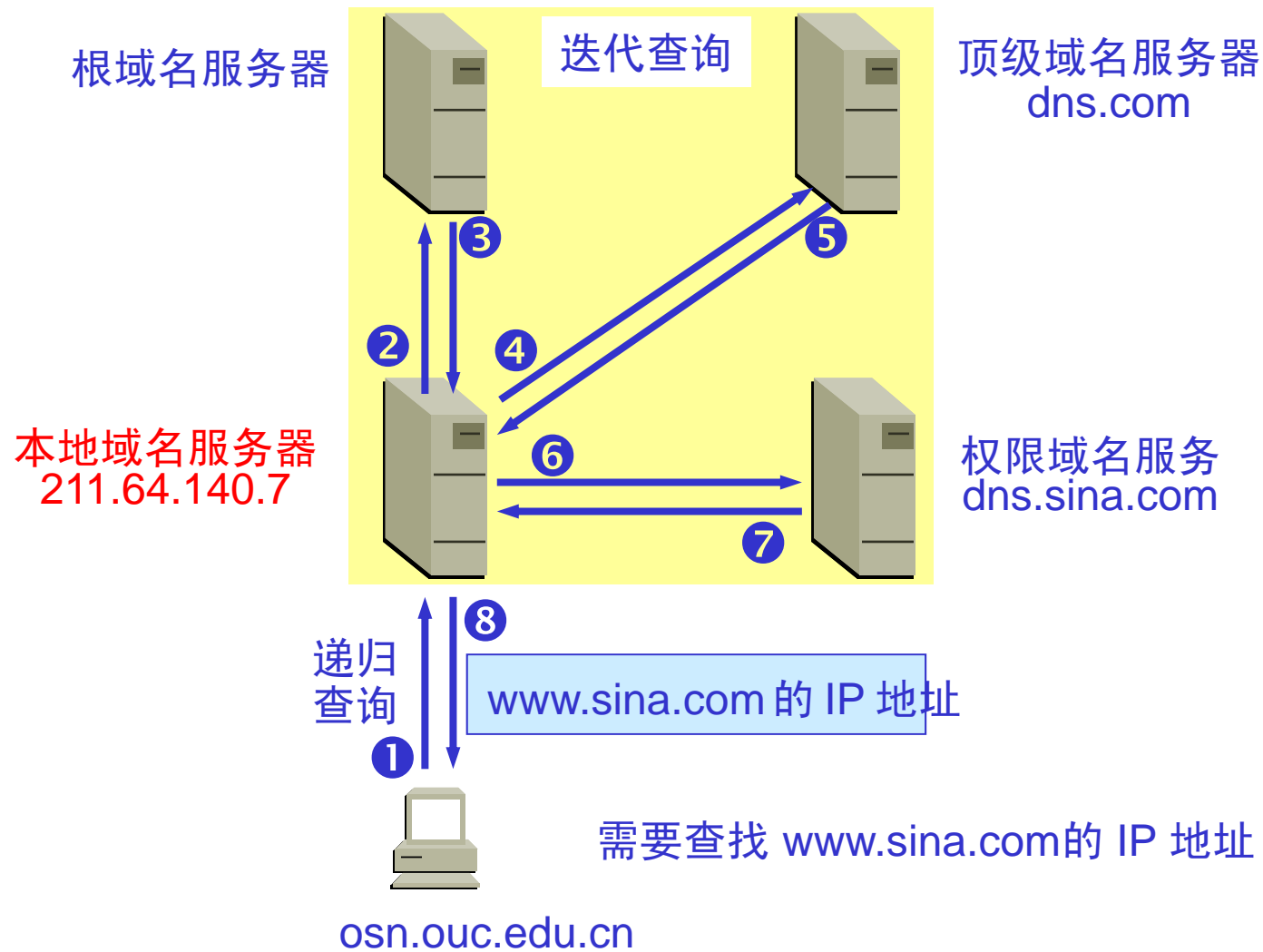
- 当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器。
- 本地域名服务器代表客户主机进行DNS查询。
 - 每一个因特网服务提供者 ISP，或一个大学，甚至一个大学里的系，都可以拥有一个本地域名服务器。
- 也称为默认域名服务器。



域名的解析过程

- 主机向本地域名服务器的查询一般都是采用**递归查询**。
 - 如果主机所询问的**本地域名服务器**不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他**根域名服务器**继续发出查询请求报文。
- 本地域名服务器向根域名服务器的查询通常是采用**迭代查询**。
 - 当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。

本地域名服务器采用迭代查询





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。
 - 可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。
 - 增加此时间值可减少网络开销，而减少此时间值可提高域名转换的准确性。



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。



6.4.2 统一资源定位符 URL

1. URL的格式

- 统一资源定位符 URL 是对可以从因特网上得到的资源的位置和访问方法的一种简洁的表示。
- URL 相当于一个文件名在网络范围的扩展。
 - 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。
- 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 的一般形式

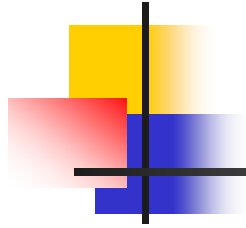
- 由以冒号隔开的两大部分组成，并且在 URL 中的字符对大写或小写没有要求。
- URL 的一般形式是：

<协议>://<主机>:<端口>/<路径>

ftp —— 文件传送协议 FTP

http —— 超文本传送协议 HTTP

News —— USENET 新闻

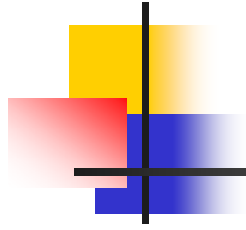


URL 的一般形式（续）

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

<协议>://<主机>:<端口>/<路径>

<主机> 是存放资源的主机
在因特网中的域名

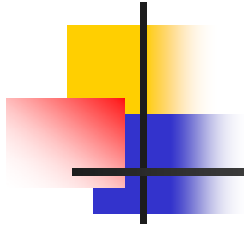


URL 的一般形式（续）

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

<协议>://<主机>:<端口>/<路径>

有时可省略



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

- 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

万维网之父的道歉

Tim Berners-Lee





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

- 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)。

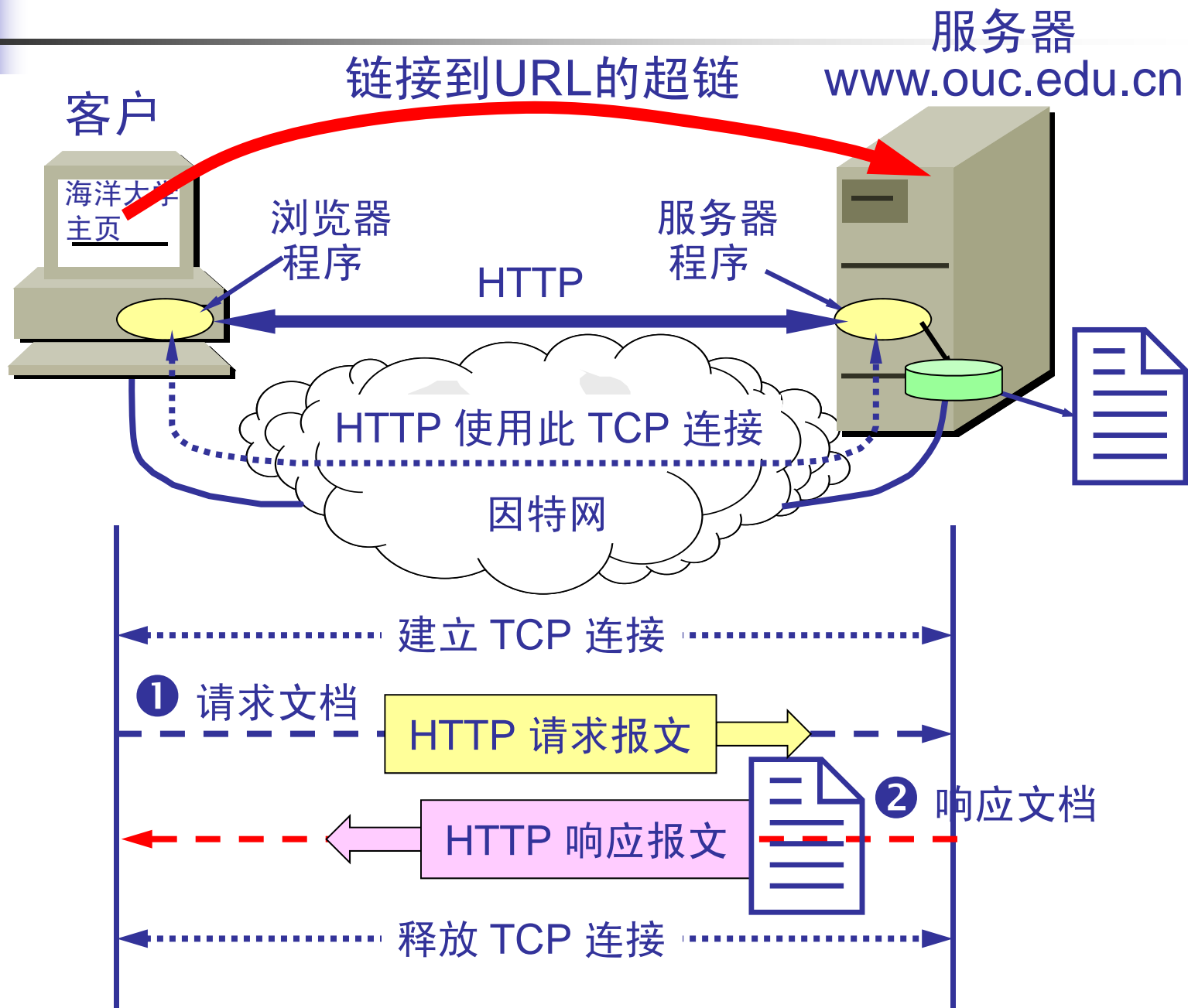


6.4.3 超文本传送协议 HTTP

1. HTTP 的操作过程

- 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，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。

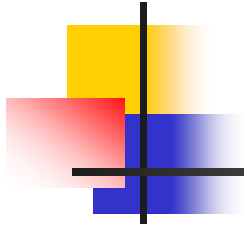
万维网的工作过程





用户点击鼠标后所发生的事件

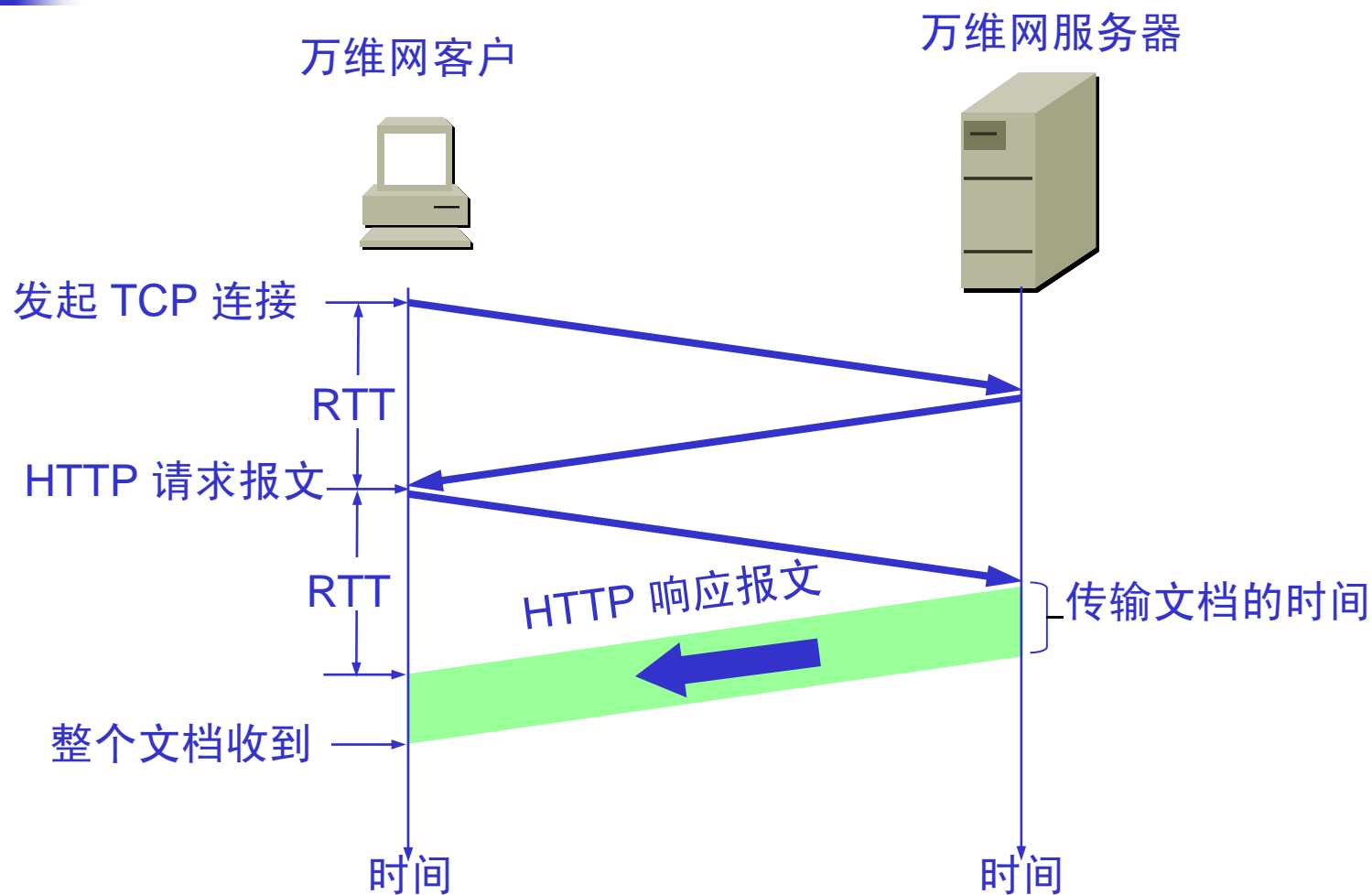
- (1) 浏览器分析超链指向页面的 URL。
- (2) 浏览器向 DNS 请求解析 `www.ouc.edu.cn` 的 IP 地址。
- (3) 域名系统 DNS 解析出我校Web服务器的 IP 地址。
- (4) 浏览器与服务器建立 TCP 连接
- (5) 浏览器发出取文件命令：
 `GET /index.html`。
- (6) 服务器给出响应，把文件 `index.html` 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示我校主页中的所有文本。

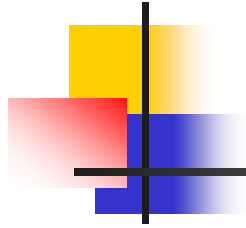


HTTP 的主要特点

- HTTP 是面向事务的客户服务器协议。
- HTTP 1.0 协议是**无状态的**(stateless)。
- HTTP 协议本身也是无连接的，虽然它使用了面向连接的 TCP 向上提供的服务。

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）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(如多媒体对象)效率。

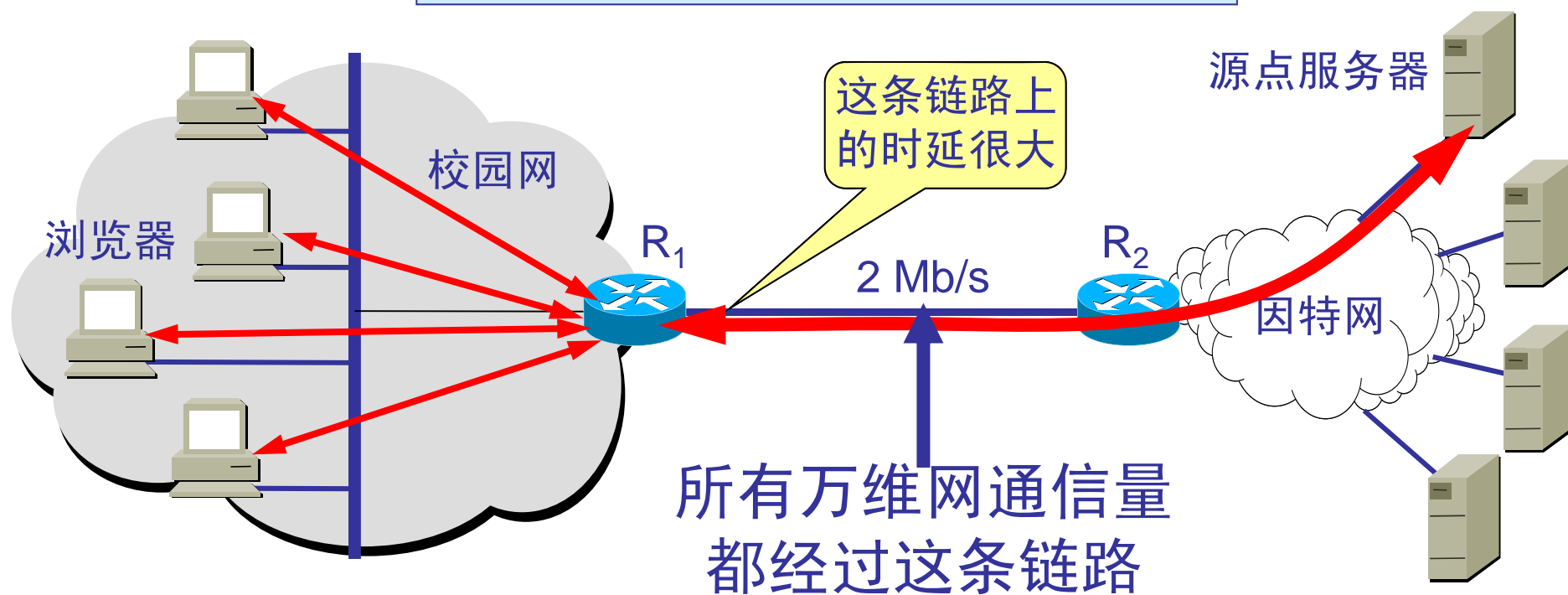


代理服务器(proxy server)

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

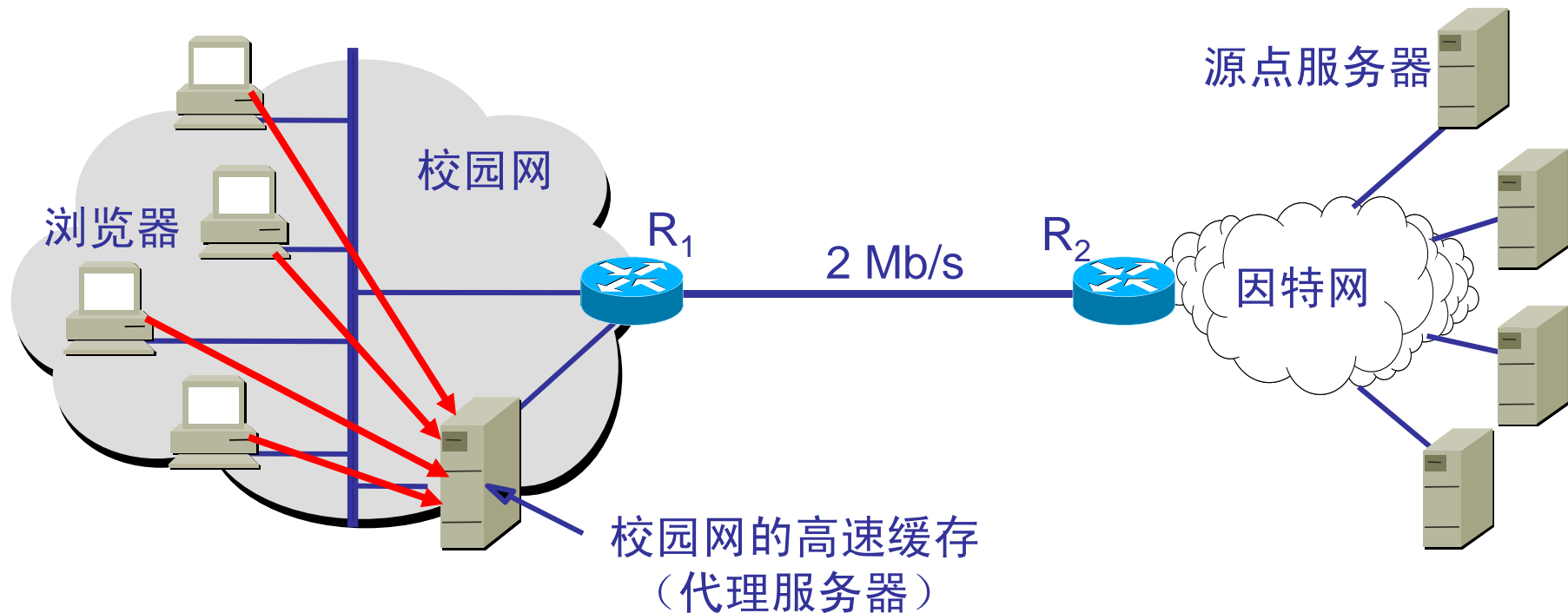
使用高速缓存可减少 访问因特网服务器的时延

没有使用高速缓存的情况



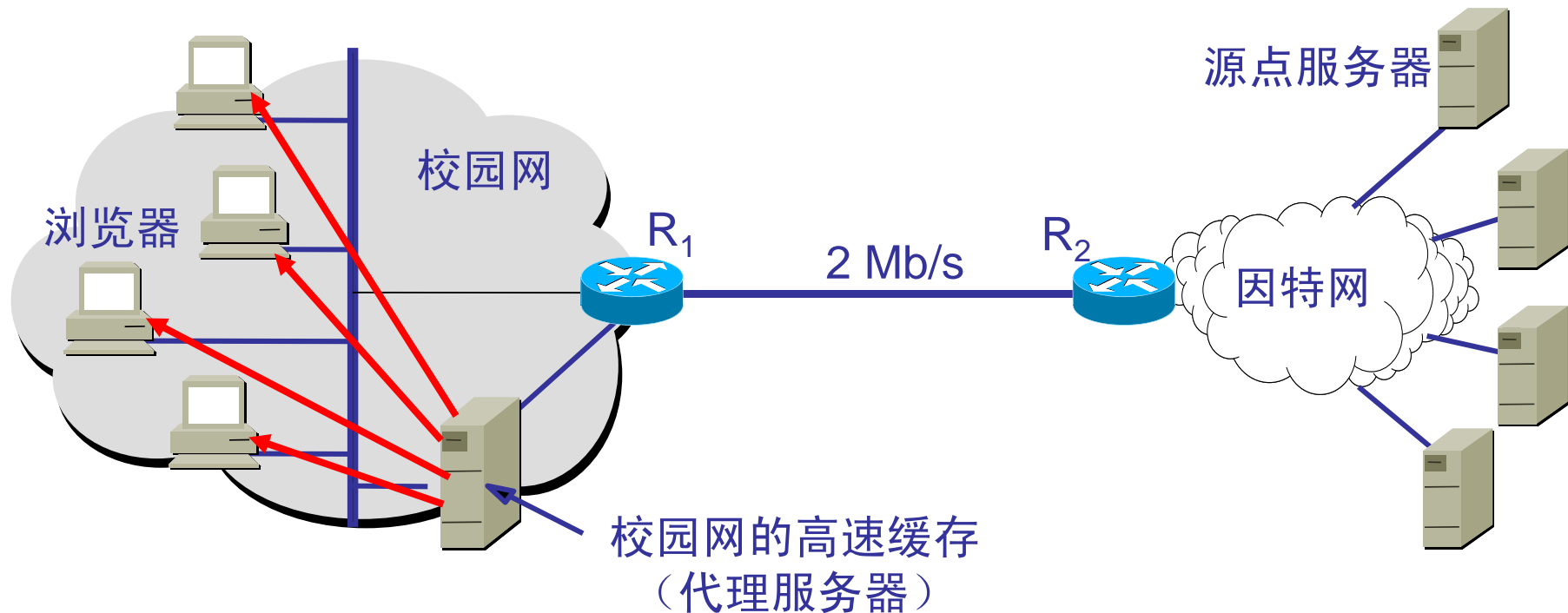
使用高速缓存的情况

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



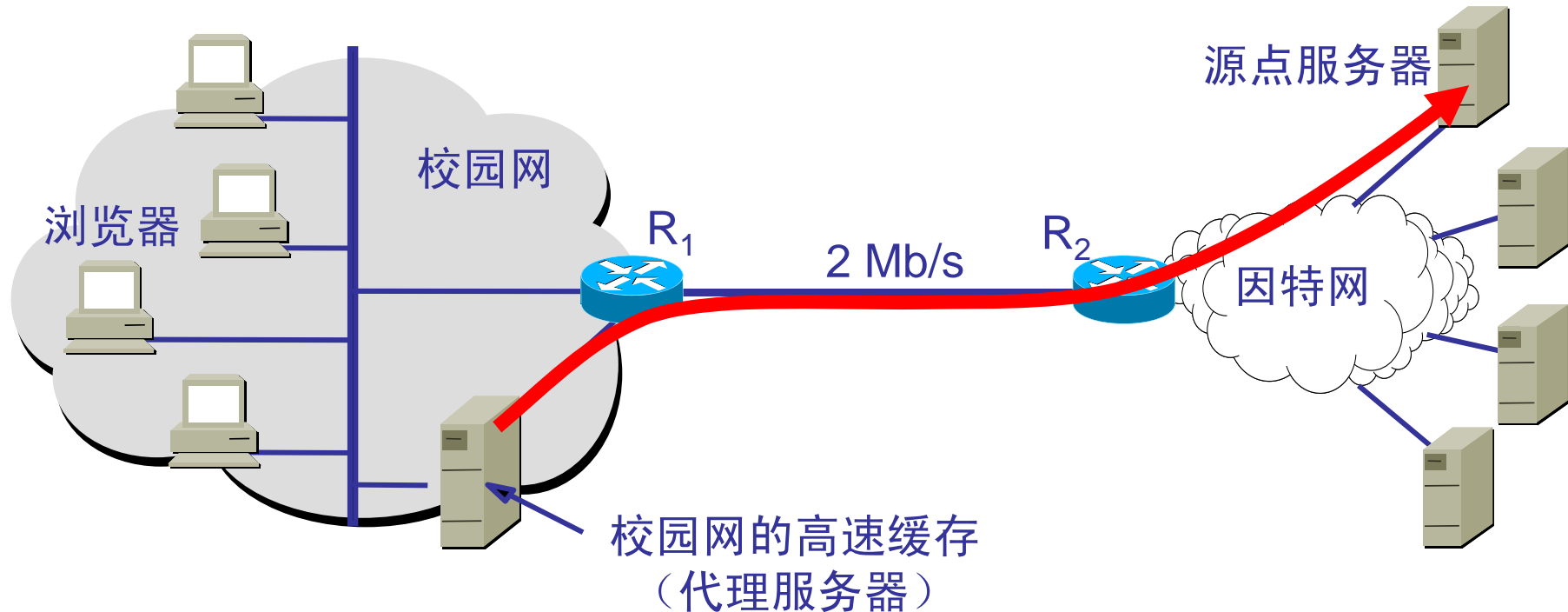
使用高速缓存的情况

(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。



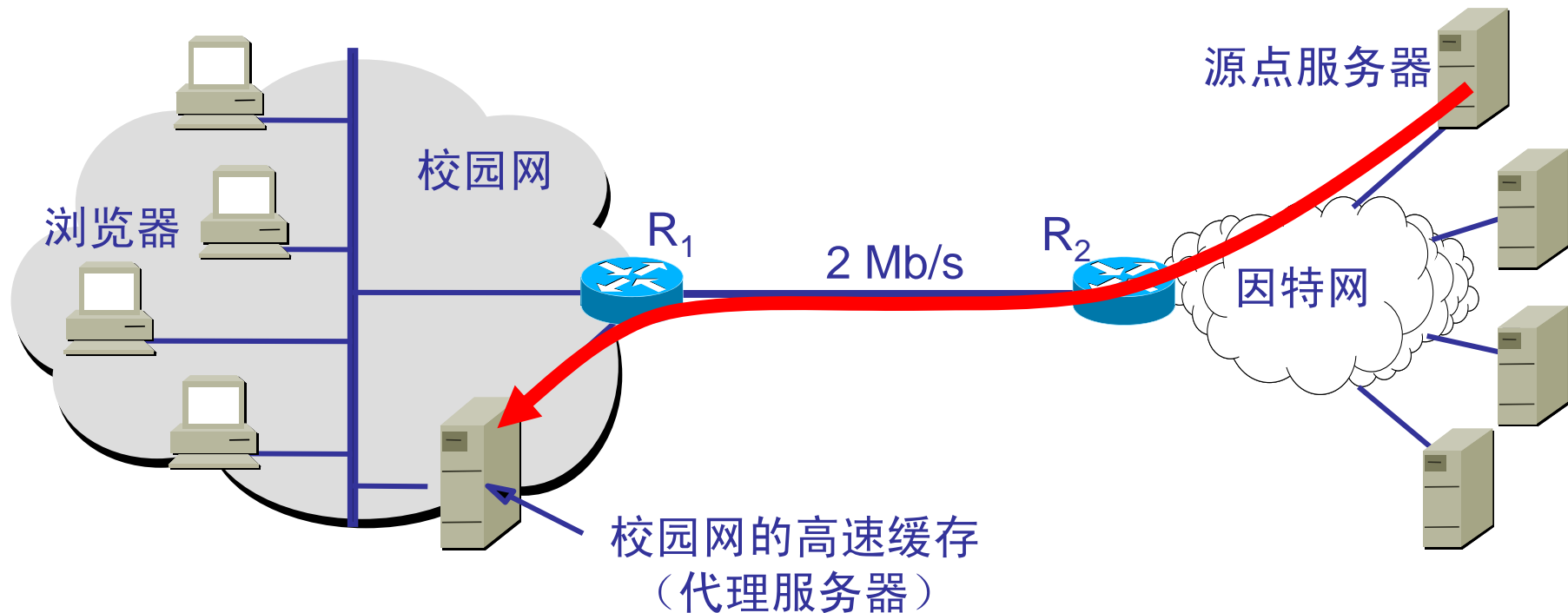
使用高速缓存的情况

(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。



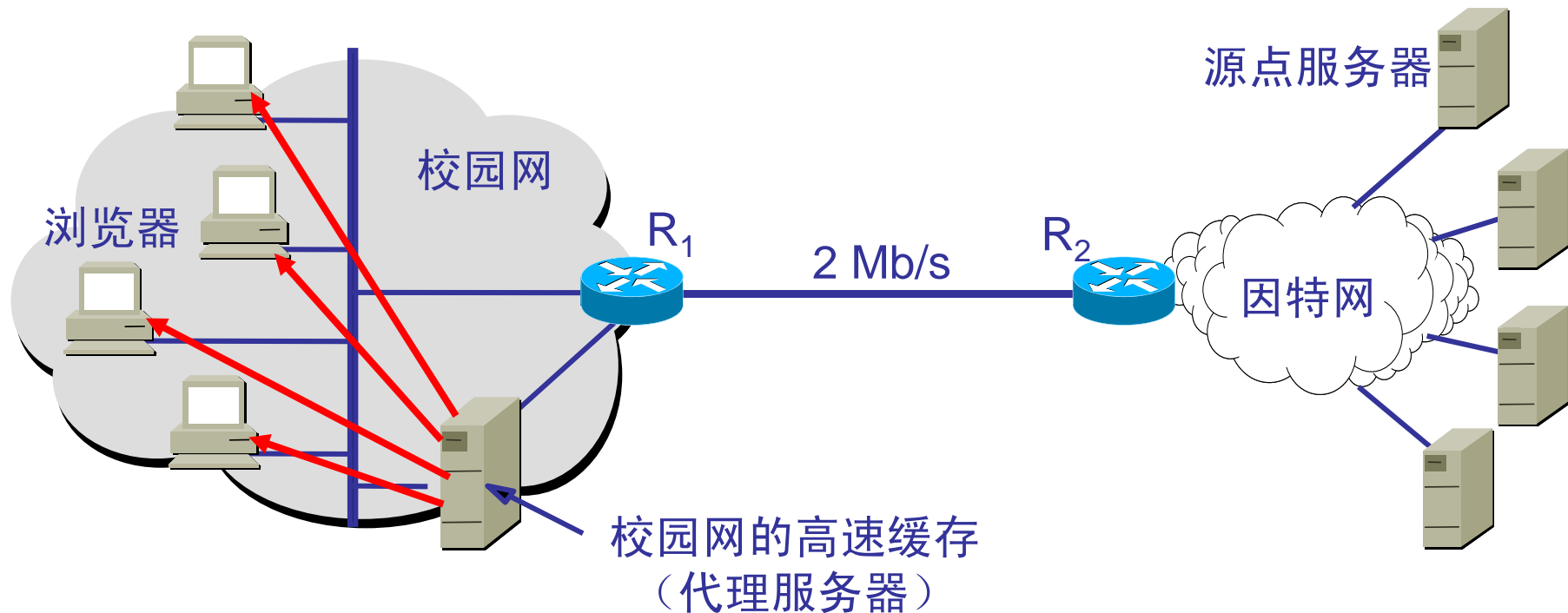
使用高速缓存的情况

(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。



使用高速缓存的情况

(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。



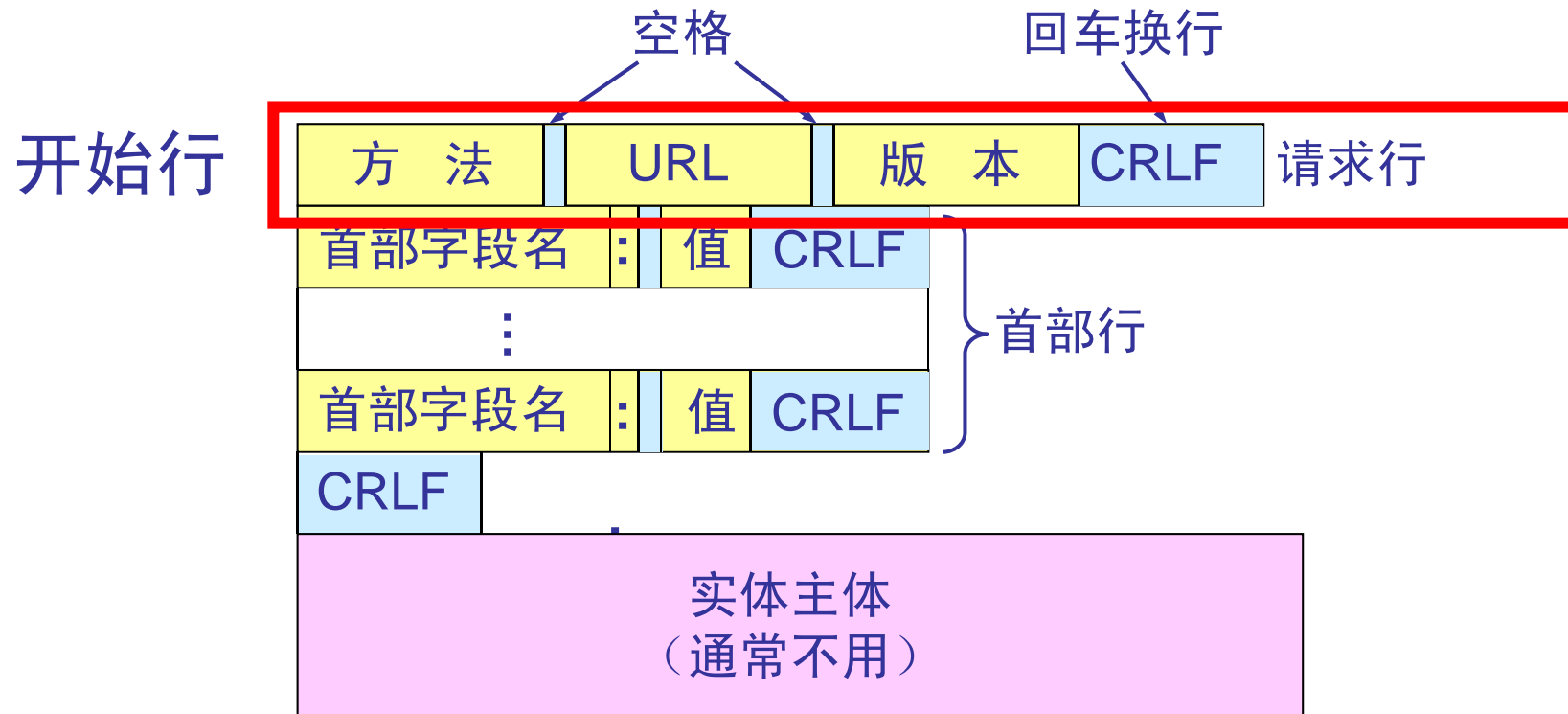


3. HTTP 的报文结构

HTTP 有两类报文：

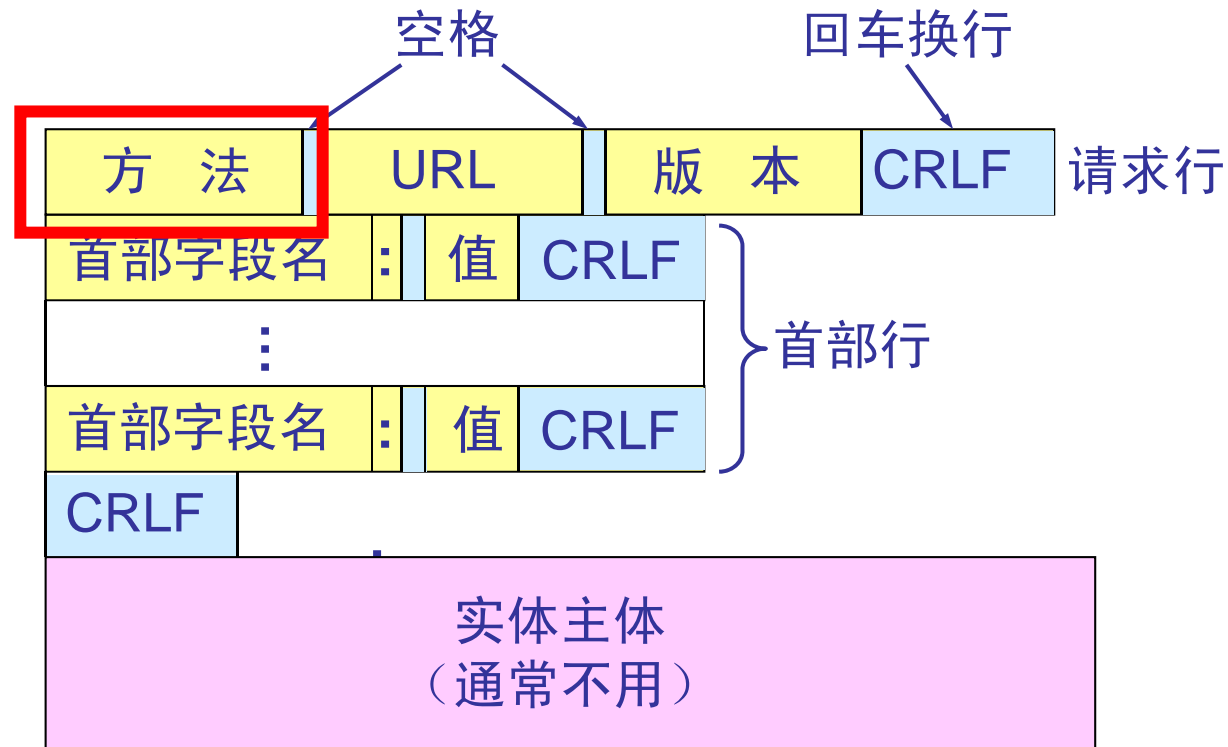
- 请求报文——从客户向服务器发送请求报文。
- 响应报文——从服务器到客户的回答。
- 由于 HTTP 是面向正文的(text-oriented)，因此在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的。

HTTP 的报文结构（请求报文）

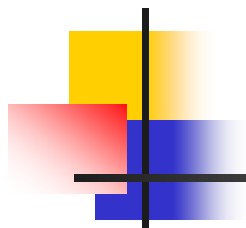


报文由三个部分组成，即开始行、首部行和实体主体。
在请求报文中，开始行就是请求行。

HTTP 的报文结构（请求报文）



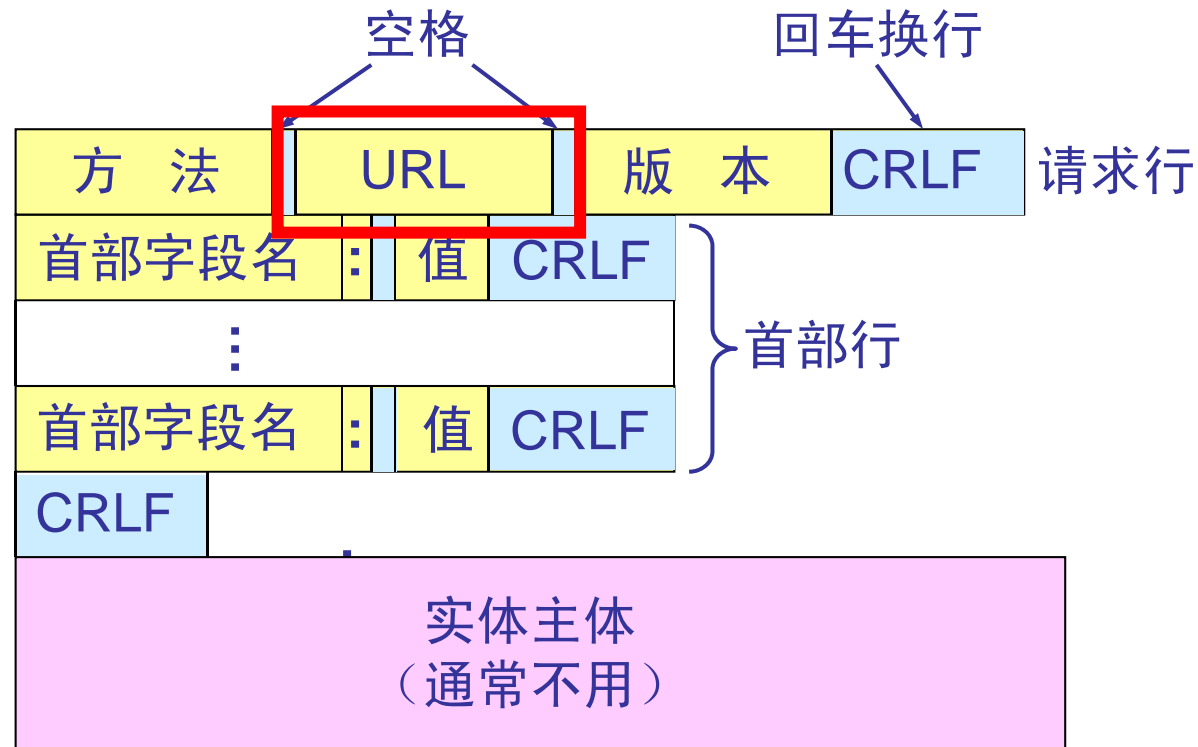
“**方法**”是面向对象技术中使用的专门名词。所谓“方法”就是**对所请求的对象进行的操作**，因此这些方法实际上也就是一些**命令**。因此，请求报文的类型是由它所采用的方法决定的。



HTTP 请求报文的一些方法

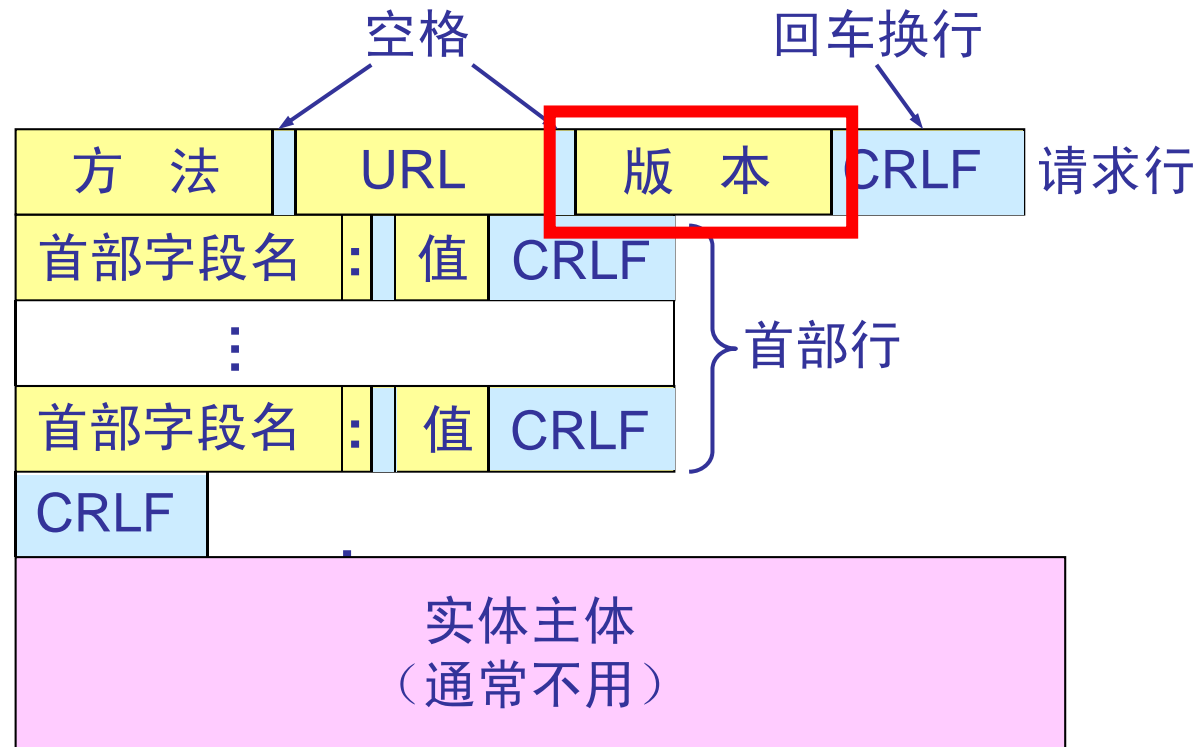
| 方法（操作） | 意义 |
|---------|---------------------|
| OPTION | 请求一些选项的信息 |
| GET | 请求读取由 URL 所标志的信息 |
| HEAD | 请求读取由 URL 所标志的信息的首部 |
| POST | 给服务器添加信息（例如，注释） |
| PUT | 在指明的 URL 下存储一个文档 |
| DELETE | 删除指明的 URL 所标志的资源 |
| TRACE | 用来进行环回测试的请求报文 |
| CONNECT | 用于代理服务器 |

HTTP 的报文结构（请求报文）



“URL”是所请求的资源的 URL。

HTTP 的报文结构（请求报文）



“版本”是 HTTP 的版本。



请求报文举例

request line
(GET, POST,
HEAD commands)

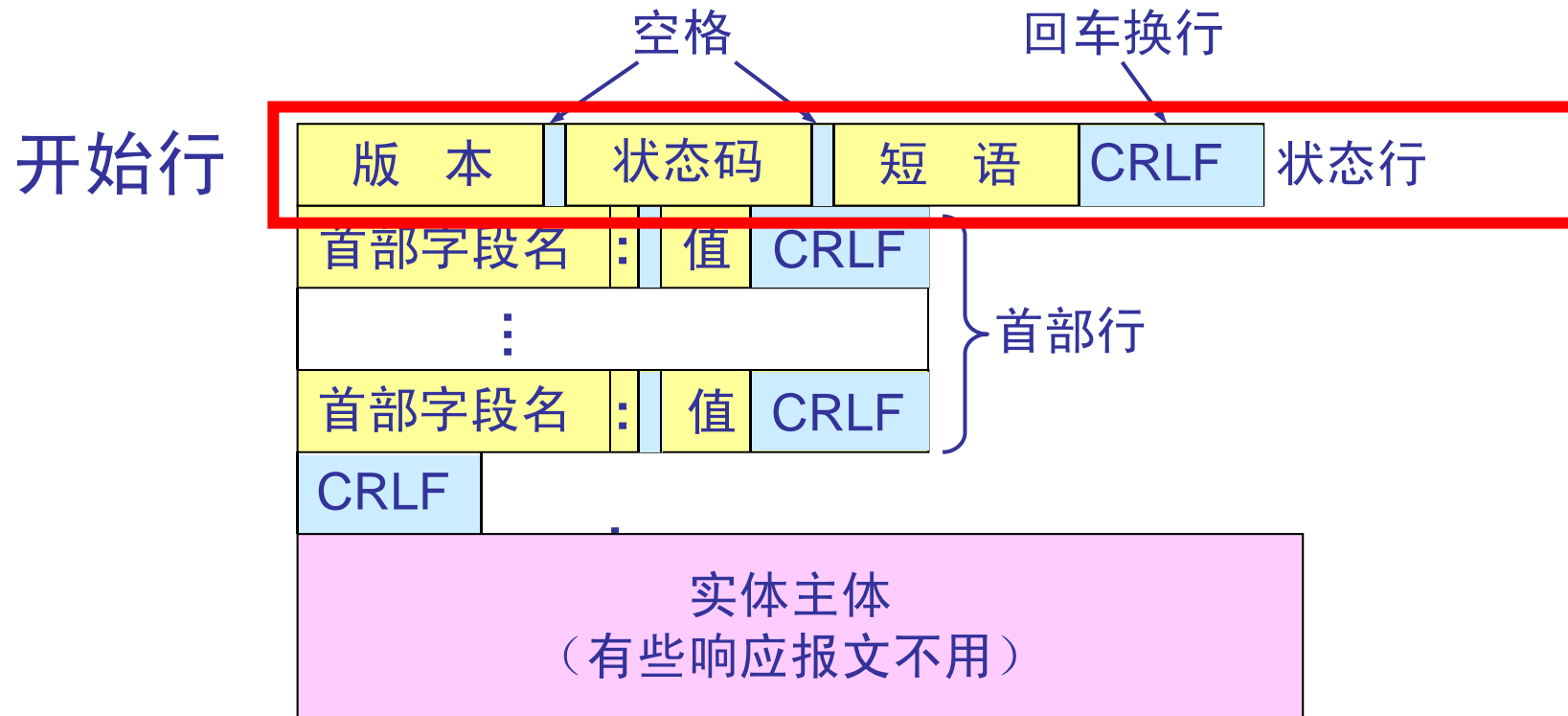
header
lines

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:en
```

Carriage return,
line feed (ASCII 010)
indicates end
of message

(extra carriage return, line feed)

HTTP 的报文结构（响应报文）



响应报文的开始行是**状态行**。

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



响应报文举例

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```



状态码都是三位数字

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



HTTP命令过程——telnet的妙用

1. Telnet 学校Web服务器

telnet www.ouc.edu.cn 80

Opens TCP connection to port 80
(default HTTP server port) at www.ouc.edu.cn
输入的所有文字被送到
port 80 at www.ouc.edu.cn

2. Type in a GET HTTP request:

GET /index.html HTTP/1.1
Host: www.ouc.edu.cn

盲打完成左边的命令后，输入两次回车，
将把最简单的GET request 送到我校的
HTTP server;
想想为什么盲打？

3. HTTP server将返回消息!

HTTP命令过程——结果

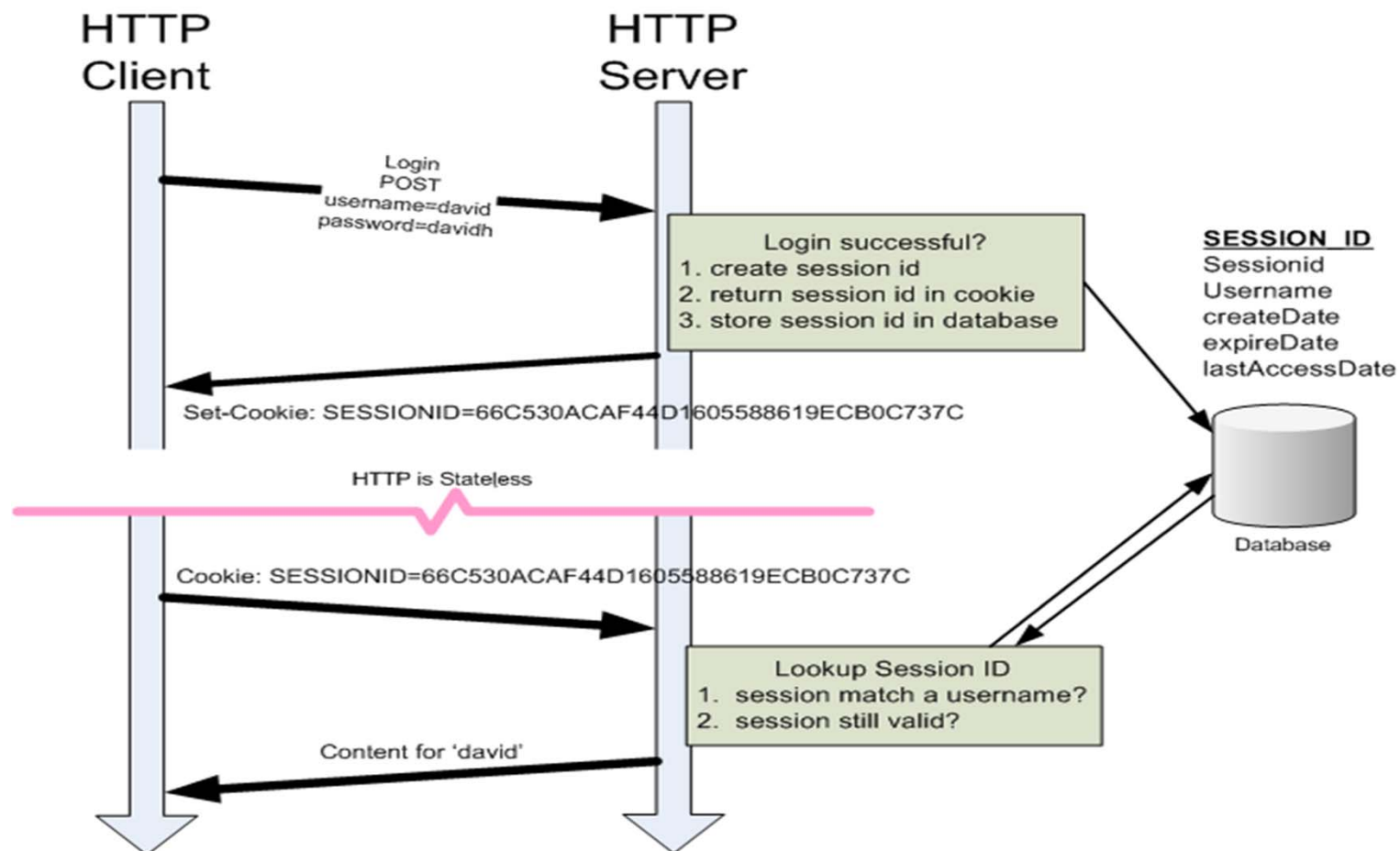
```
C:\Windows\system32\cmd.exe

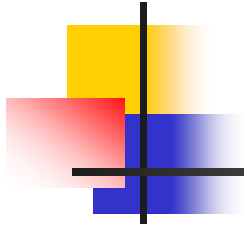
HTTP/1.1 200 OK
Content-Length: 1747
Content-Type: text/html
Last-Modified: Thu, 19 Jun 2008 13:32:14 GMT
Accept-Ranges: bytes
ETag: "0836ae210d2c81:75f"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Mon, 07 Dec 2009 12:07:40 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
                                                                    "http://www.w3.org
/TR/html4/loose.dtd">
    <HTML>
        <HEAD>
            <TITLE>Ocean University of China - Language Sel
ection</TITLE>
            <META http-equiv="pragma" content="no-cache">
            <META http-equiv="cache-control" content="private">
            <META NAME="Generator" CONTENT="EditPlus">
            <META NAME="Keywords" CONTENT="Ocean University of China">
            <META NAME="Description" CONTENT="Ocean University of China">
            <link href="style.css" type="text/css" rel="stylesheet">
```

4. 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。

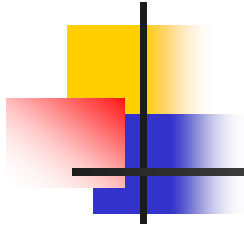




6.4.4 万维网的文档

1. 超文本标记语言 HTML

- 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（即标签）。
- 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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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 文档中标签的用法

<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>

第二个段落

</BODY>

</HTML>



HTML 文档中标签的用法

<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 文档中标签的用法

<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>

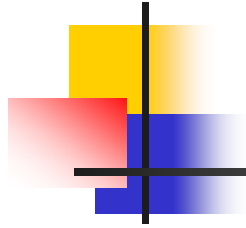
HTML 文档结束





2. 动态万维网文档

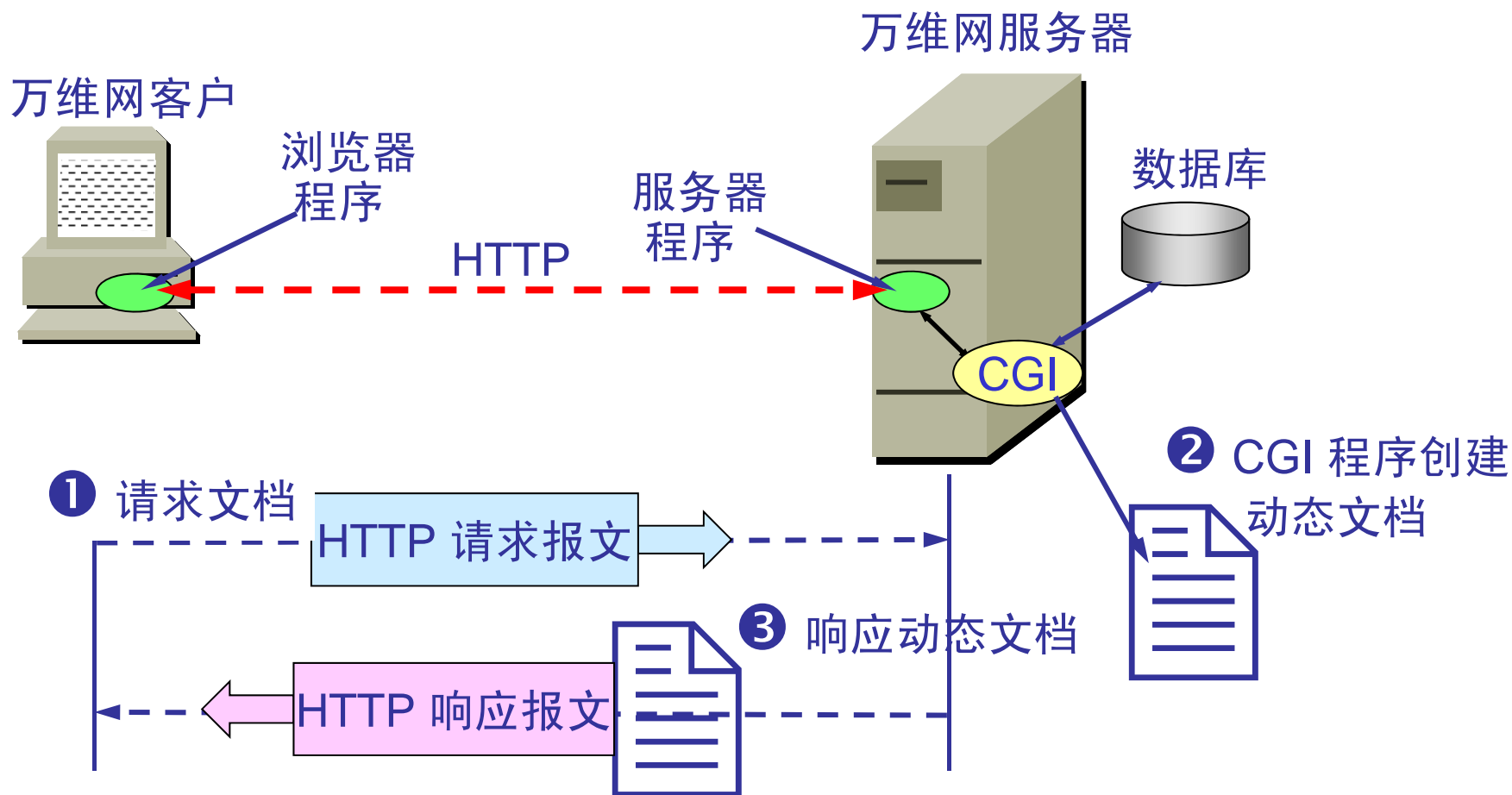
- **静态文档**是指该文档创作完毕后就存放在万维网服务器中，在被用户浏览的过程中，内容不会改变。
- **动态文档**是指文档的内容是在浏览器访问万维网服务器时才由应用程序动态创建。
- 动态文档和静态文档之间的主要差别体现在**服务器**一端。这主要是文档内容的生成方法不同。而从浏览器的角度看，这两种文档并没有区别。

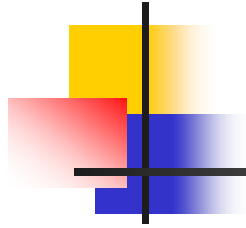


万维网服务器功能的扩充

- (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

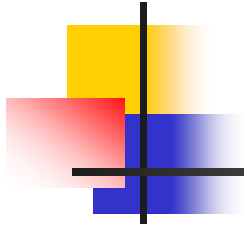




通用网关接口 CGI

(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

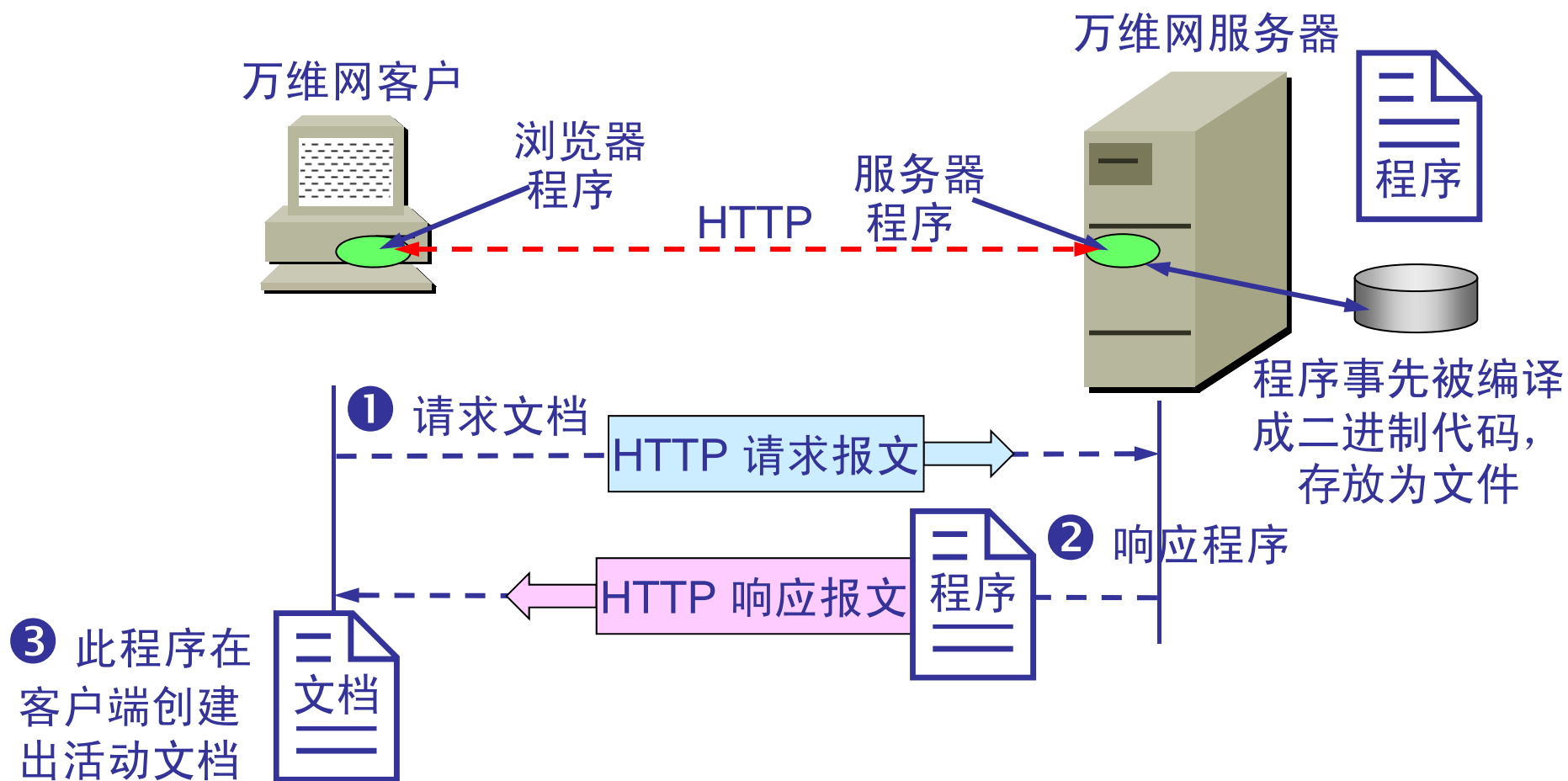
- CGI 程序的正式名字是 CGI 脚本(script)。
- “脚本”指的是一个程序，它被另一个程序（解释程序）而不是计算机的处理机来解释或执行。
- 脚本运行起来要比一般的编译程序要慢，因为它的每一条指令先要被另一个程序来处理（这就要一些附加的指令），而不是直接被指令处理器来处理。

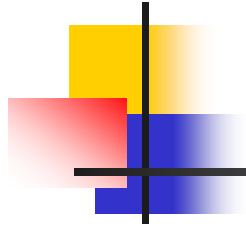


3. 活动万维网文档

- **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 技术装三个主要组成部分

- 程序设计语言。

- Java 包含一个新的程序设计语言，用来编写传统的计算机程序和 Java 小应用程序。

- 运行(runtime)环境。

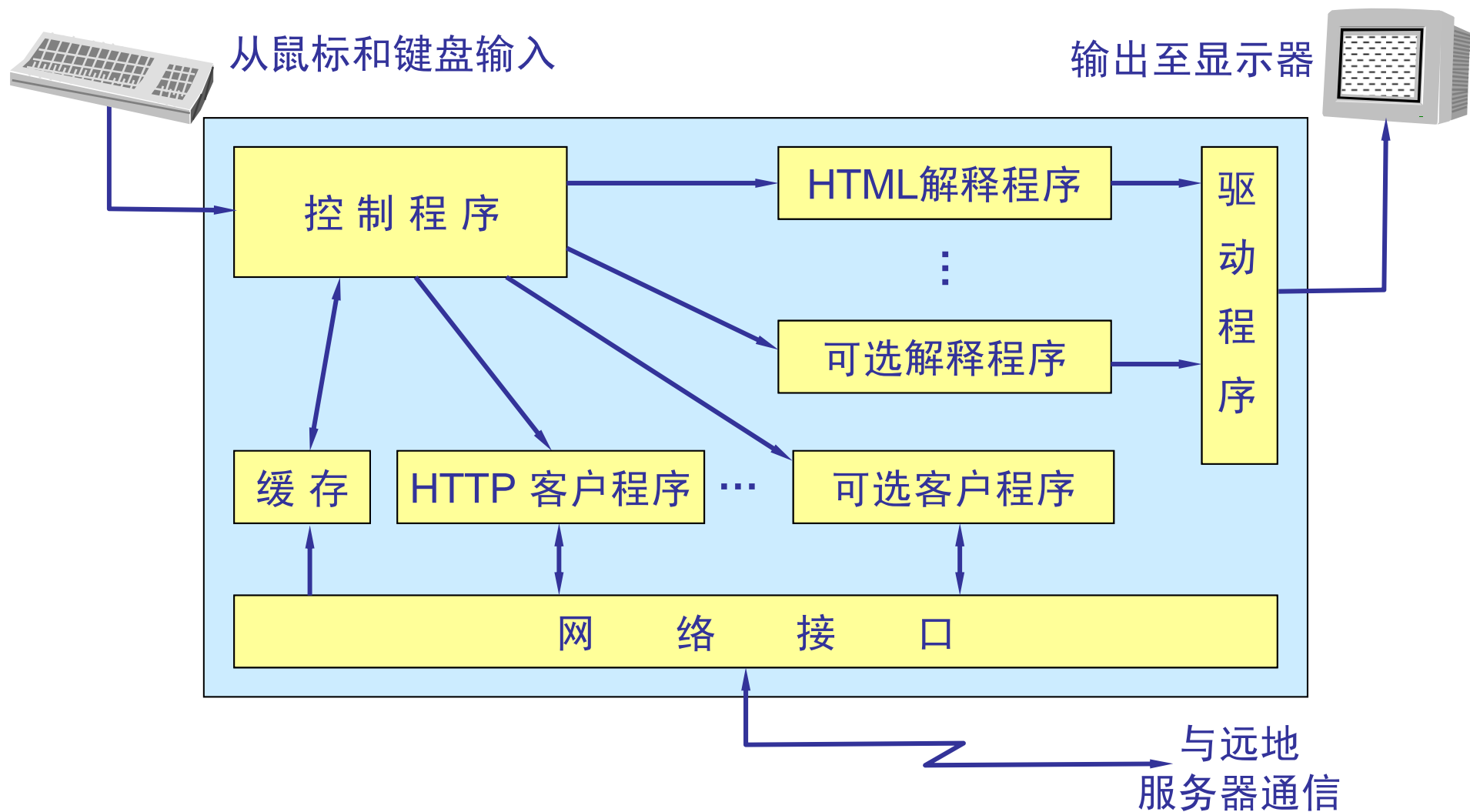
- 运行 Java 程序所必须的环境，主要包括包括 Java 虚拟机（简称为 JVM），该软件定义了 Java 二进制代码的执行模型。

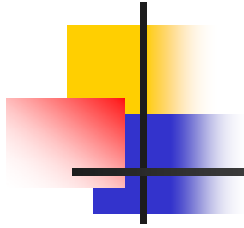
- 类库(class library)。

- 为了更容易编写 Java 小应用程序，Java 提供了强大的类库支持。

- Coding Once, Run Everywhere!

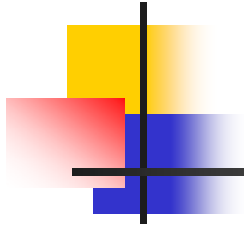
4. 浏览器的结构





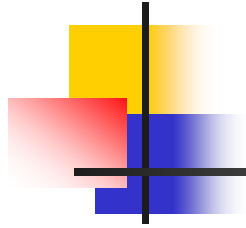
浏览器的主要组成部分

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



解释程序

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



浏览器中的缓存

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



6.4.5 万维网的信息检索系统

1. 全文检索搜索和分类目录搜索

- 搜索引擎(Search Engine, SE)是收集、整理和组织信息并为用户提供查询服务的信息系统。
 - 面向Web的SE是最典型的代表。
 - 三大特点：事先下载，事先组织，实时检索。
- **全文检索搜索引擎**是一种纯技术型的检索工具。
 - 通过网页收集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。
 - 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

- 不采集网站的任何信息，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。



体育



娱乐



财经



YAHOO!
中国雅虎®



论坛



邮箱



时尚

网页 资讯 音乐 图片 知识堂 淘宝购物

搜索

今日热点: 教育部网站撤下不满6岁儿童可入学回复 北京卫生局: 重症甲流患者治疗费需七八万 “钓鱼执法”案两官员受警告处分



资讯



体育



财经



娱乐



时尚



汽车



自然



旅游



育儿



男人



星座

今日焦点

青岛 6°C~8°C 更多



09年度人物揭晓 长江大学救人英雄入选

09年度人物中有功业卓著的大家，不屈不挠的小人物，还有从机遇中走到聚光灯下的富豪...[\[详细\]](#)

“人梯”英雄永长存

八宝山送别钱学森

[更多>>](#)

☐ 留在首页 ☐ 进入邮箱

登录

 雅虎邮箱

 机票酒店

 我的彩票

雅虎文摘

- 哥本哈根 期待政治家的正确选择 精品文摘
- 陈冠希否认富婆投资 与阿娇上海擦肩 体育头条
- 时尚消费文摘: 年末血拼 送您购物宝典 晒包得大奖
- 举报互联网和手机色情低俗信息奖励办法发布

千寻Cianxun.com

女人我最大



垂直搜索引擎(Vertical Search Engine)

- 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。



著名搜索引擎历史

- 1986年，Internet正式形成。
- 现代搜索引擎的祖先：1990年由加拿大蒙特利尔McGill大学学生Alan Emtage发明的Archie，是对FTP文件名搜索，首次采用“机器人”自动爬行程序。
- 第一个用于监测互联网发展规模的“机器人”程序是1993年MIT的Matthew Gray开发的World wide Web Wanderer。刚开始它只用来统计互联网上的服务器数量，后来则发展为能够检索网站域名。
- Lycos：第一个现代意义上的WEB搜索引擎，CMU机器翻译中心的Michael Mauldin于1994年7月创建。
- Yahoo：斯坦福大学博士生David Filo和Jerry Yang(杨致远)创建1995年。
- Google：斯坦福大学博士生Larry Page与Sergey Brin于1998年9月创建，目前是全球最受欢迎的搜索引擎。
- Baidu：超链分析专利发明人、前Infoseek资深工程师李彦宏与好友徐勇发布于2001年10月，是目前最受欢迎的中文搜索引擎之一。

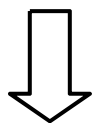
网络搜索引擎工作原理*

WWW上的文本数据

1 Kilobyte = a
very short story

今天晴空高照，万里无云，
我们去中山公园春游，
很好玩！

WWW表层有
35 Terabytes



35所大学的图书馆
(700,000 米书架!)

1 Megabyte =
1本小说



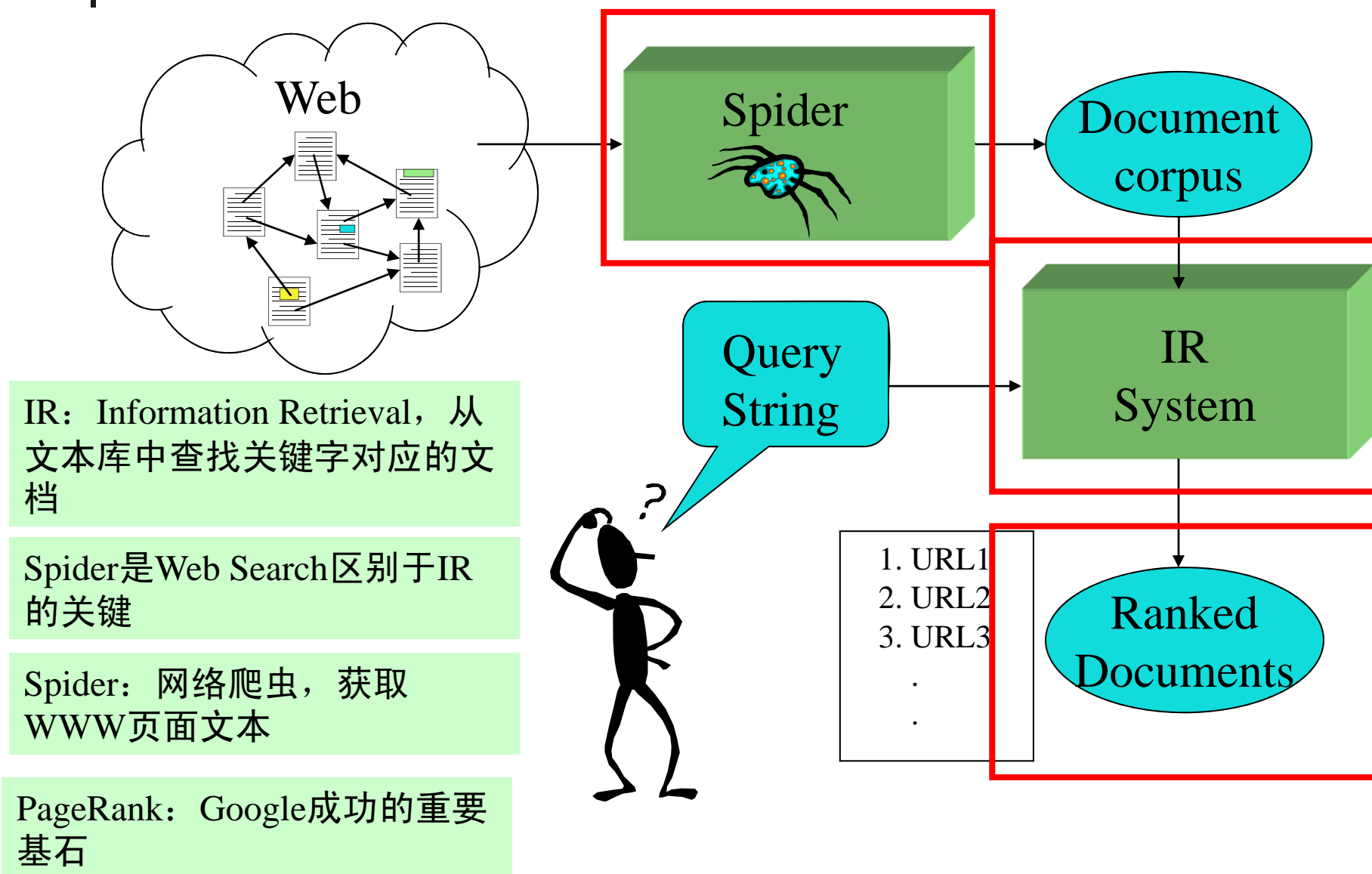
1 Gigabyte =
20米书架



1 Terabyte =
1所大学的图书馆



网络搜索引擎工作原理功能图*





信息检索(Information Retrieval)

■ 信息检索

- 广义：从文档集合中返回满足用户需求的相关信息的过程。
 - 作为一门学科，是研究信息的获取(acquisition)、表示(representation)、存储(storage)、组织(organization)和访问(access)的一门学问。
- 狭义：从非结构化的文档集中找出与用户需求相关的信息。

■ 两种研究方式

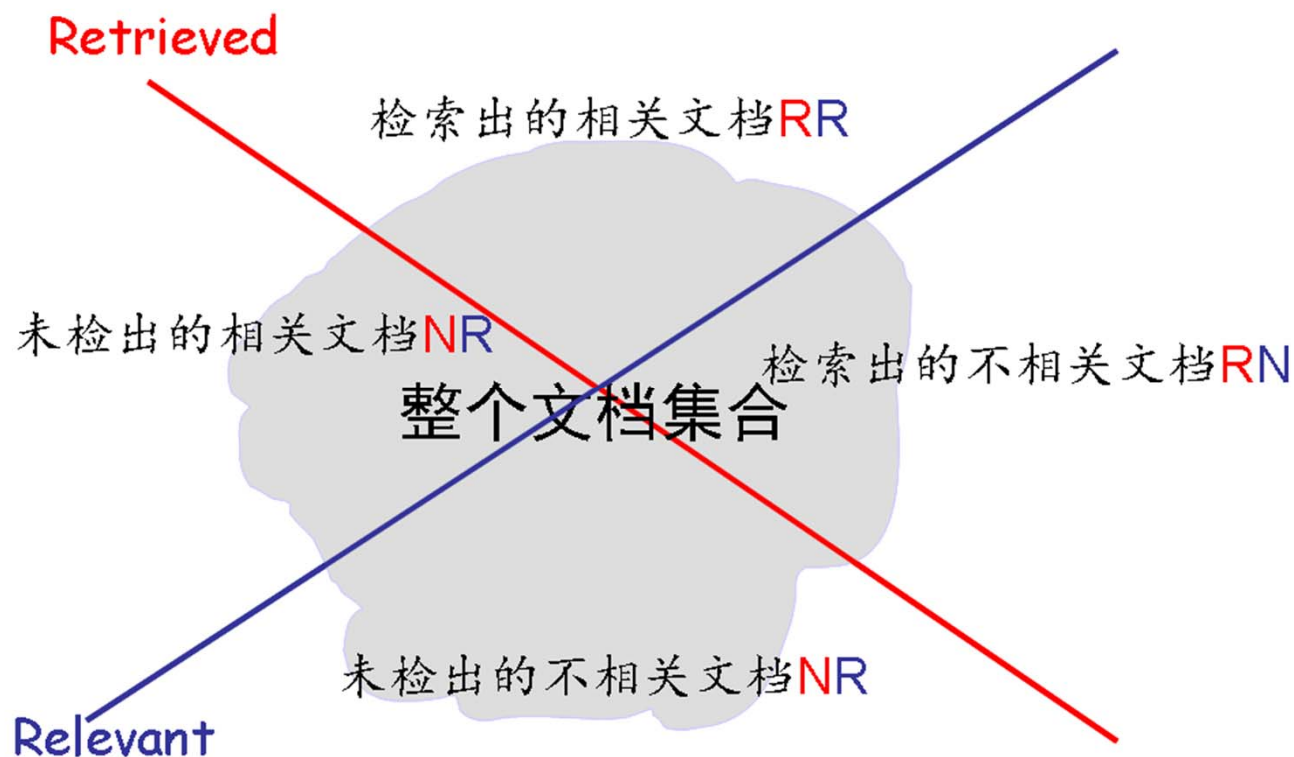
- 以计算机为中心：IR的工作主要是建立索引、对用户查询进行处理、排序算法等等
- 以用户为中心：IR的主要工作是考察用户的行为、理解用户的需求、这些行为和需求如何影响检索系统的组织
- 以计算机为中心的IR问题，目前是主流



信息检索评价

- 信息检索系统的目标是较少消耗情况下尽快、全面返回准确的结果。
 - 效率(Efficiency)—可以采用通常的评价方法：时间开销、空间开销、响应速度
- 效果(Effectiveness)
 - 返回的文档中有多少相关文档
 - 所有相关文档中返回了多少
 - 返回得靠不靠前（排序）

信息检索评价说明图

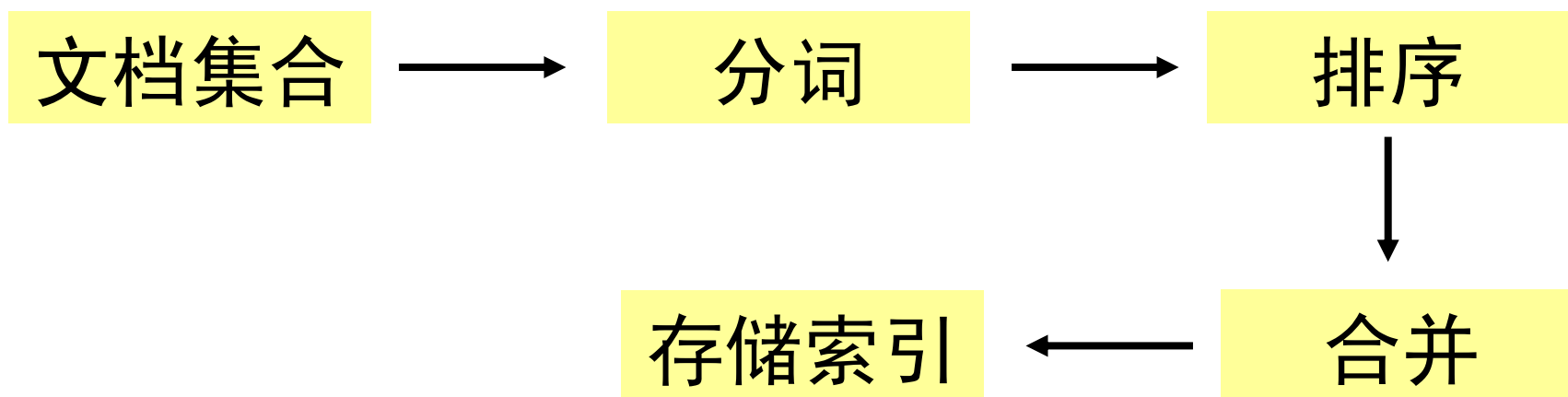


- 召回率(Recall): $RR/(RR + NR)$
 - 返回的相关结果数占实际相关结果总数的比率，也称为查全率。
- 正确率(Precision): $RR/(RR + RN)$
 - 返回的结果中真正相关结果的比率，也称为查准率。



信息检索主要内容

■ 信息存储——建索引



■ 查询

建索引步骤一：分词

■ (词汇, 文档号) 表

Doc 1

I did enact Julius
Caesar I was killed
i' the Capitol;
Brutus killed me.



So let it be with
Caesar. The noble
Brutus hath told you
Caesar was ambitious

Doc 2



| Term | docID |
|-----------|-------|
| I | 1 |
| did | 1 |
| enact | 1 |
| julius | 1 |
| caesar | 1 |
| I | 1 |
| was | 1 |
| killed | 1 |
| i' | 1 |
| the | 1 |
| capitol | 1 |
| brutus | 1 |
| killed | 1 |
| me | 1 |
| so | 2 |
| let | 2 |
| it | 2 |
| be | 2 |
| with | 2 |
| caesar | 2 |
| the | 2 |
| noble | 2 |
| brutus | 2 |
| hath | 2 |
| told | 2 |
| you | 2 |
| caesar | 2 |
| was | 2 |
| ambitious | 2 |
| | |
| | |
| | |

建索引步骤二：排序

- 词汇顺序
 - 文档号

| Term | docID |
|-----------|-------|
| I | 1 |
| did | 1 |
| enact | 1 |
| julius | 1 |
| caesar | 1 |
| I | 1 |
| was | 1 |
| killed | 1 |
| i' | 1 |
| the | 1 |
| capitol | 1 |
| brutus | 1 |
| killed | 1 |
| me | 1 |
| so | 2 |
| let | 2 |
| it | 2 |
| be | 2 |
| with | 2 |
| caesar | 2 |
| the | 2 |
| noble | 2 |
| brutus | 2 |
| hath | 2 |
| told | 2 |
| you | 2 |
| caesar | 2 |
| was | 2 |
| ambitious | 2 |
| | |
| | |
| | |



| Term | docID |
|-----------|-------|
| ambitious | 2 |
| be | 2 |
| brutus | 1 |
| brutus | 2 |
| capitol | 1 |
| caesar | 1 |
| caesar | 2 |
| caesar | 2 |
| did | 1 |
| enact | 1 |
| hath | 1 |
| I | 1 |
| I | 1 |
| i' | 1 |
| it | 2 |
| julius | 1 |
| killed | 1 |
| killed | 1 |
| let | 2 |
| me | 1 |
| noble | 2 |
| so | 2 |
| the | 1 |
| the | 2 |
| told | 2 |
| you | 2 |
| was | 1 |
| was | 2 |
| with | 2 |
| | |
| | |
| | |

建索引步骤二：合并

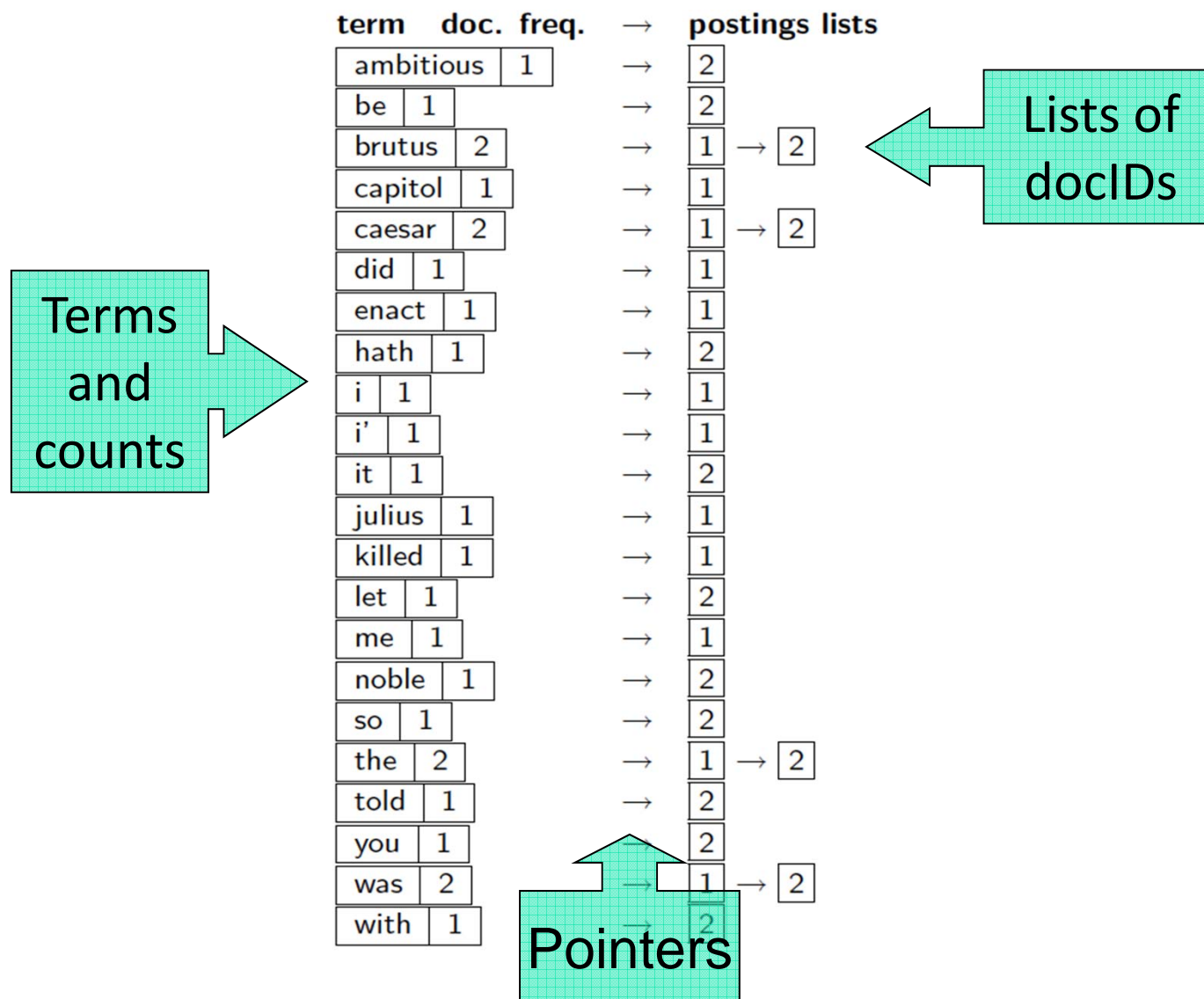
- 按关键字进行列表合并
- 合并过程中计算关键词出现的频率

| Term | docID |
|-----------|-------|
| ambitious | 2 |
| be | 2 |
| brutus | 1 |
| brutus | 2 |
| capitol | 1 |
| caesar | 1 |
| caesar | 2 |
| caesar | 2 |
| did | 1 |
| enact | 1 |
| hath | 1 |
| I | 1 |
| I | 1 |
| i' | 1 |
| it | 2 |
| julius | 1 |
| killed | 1 |
| killed | 1 |
| let | 2 |
| me | 1 |
| noble | 2 |
| so | 2 |
| the | 1 |
| the | 2 |
| told | 2 |
| you | 2 |
| was | 1 |
| was | 2 |
| with | 2 |
| | |
| | |



| term | doc. freq. | → | postings lists |
|-----------|------------|---|----------------|
| ambitious | 1 | → | [2] |
| be | 1 | → | [2] |
| brutus | 2 | → | [1] → [2] |
| capitol | 1 | → | [1] |
| caesar | 2 | → | [1] → [2] |
| did | 1 | → | [1] |
| enact | 1 | → | [1] |
| hath | 1 | → | [2] |
| i | 1 | → | [1] |
| i' | 1 | → | [1] |
| it | 1 | → | [2] |
| julius | 1 | → | [1] |
| killed | 1 | → | [1] |
| let | 1 | → | [2] |
| me | 1 | → | [1] |
| noble | 1 | → | [2] |
| so | 1 | → | [2] |
| the | 2 | → | [1] → [2] |
| told | 1 | → | [2] |
| you | 1 | → | [2] |
| was | 2 | → | [1] → [2] |
| with | 1 | → | [2] |

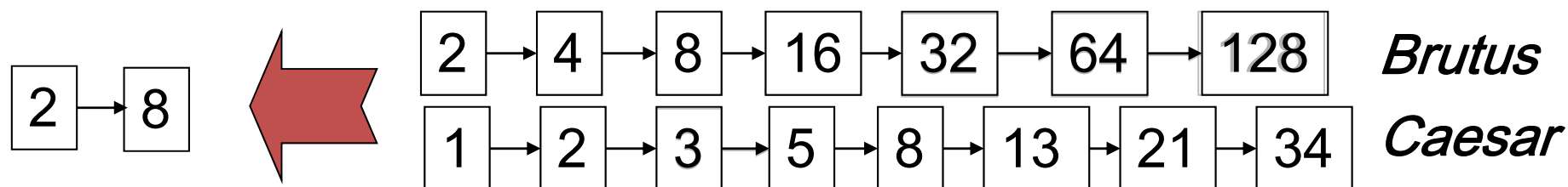
建索引步骤三：存储



布尔查询: AND

■ 查询目标: *Brutus AND Caesar*

- 在字典中找到 *Brutus*, 取出相关链表
- 在字典中找到 *Caesar*, 取出相关链表



■ 遍历链表，求取交集

- 如果链表的长度分别为 x 和 y , 则求交的开销为 $O(x+y)$
- 文档必须是按照ID有序排列的



求交算法

INTERSECT(p_1, p_2)

```
1  answer  $\leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then  $\text{ADD}(\text{answer}, \text{docID}(p_1))$ 
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8          then  $p_1 \leftarrow \text{next}(p_1)$ 
9          else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return answer
```



信息收集工具：Web Crawler

- 在服务器运行的并行代码：
 - 从根URL集合中取出一条URL，取回对应URL所对应的页面
 - 分析该页面
 - 构造该页面的索引表
 - 找出该页面的所有超链
 - 按照深度优先和广度优先的办法，递归的沿着超链前进
- 对所有页面定期的重新建索引，保证索引的有效性



结果排序算法

■ 什么样的页面更重要？

■ 链接分析方法(Link Analysis)

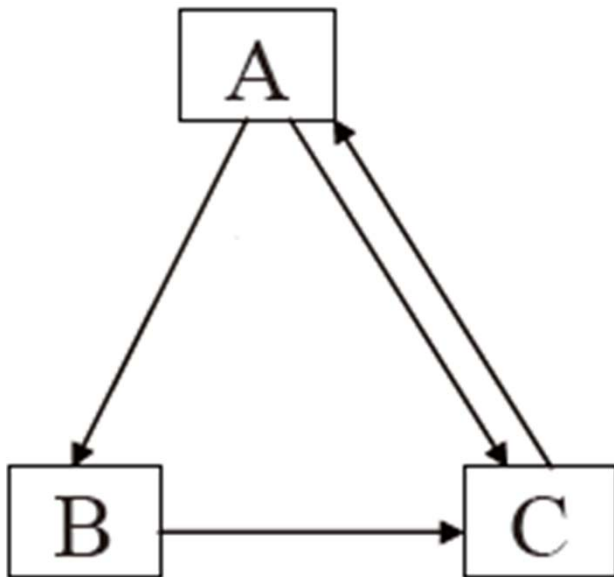
- Google 的PageRank：与查询无关
- 受启发于文献引用，越多越重要的文献引用的文献越重要
- WEB上的链接关系看成引用

■ IBM的HITS算法：与查询相关

- 查询的越多的文档越重要

Google PageRank算法

- 一个网页的PageRank等于所有的指向它的网页的PageRank的分量之和(c 为归一化参数)。



$$\begin{cases} R(A)=R(C) \\ R(B)=0.5R(A) \\ R(C)=R(B)+0.5R(A) \\ R(A)+R(B)+R(C)=1 \end{cases}$$

解上述方程得

$$R(A)=R(C)=0.4$$

$$R(B)=0.2$$



Web搜索引擎总结

- 搜索引擎的关键技术

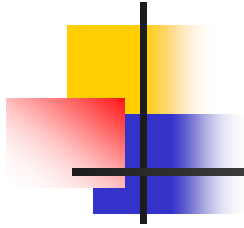
- 信息收集、信息检索、结果排序

- Google

- 1998, 26Mpages
 - 2000, 1Gpages
 - 2008, 1Tpages

- 但我们看几页结果？

- 查全率、查准率问题



6.2 文件传送协议

6.2.1 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。



文件传送并非很简单的问题

- 网络环境中的一项基本应用就是将文件从一台计算机中复制到另一台可能相距很远的计算机中。
 - 初看起来，在两个主机之间传送文件是很简单的事情。
 - 其实这往往非常困难。原因是众多的计算机厂商研制出的文件系统多达数百种，且差别很大。
- 网络环境下复制文件的复杂性
 - 计算机存储数据的格式不同。
 - 文件的目录结构和文件命名的规定不同。
 - 对于相同的文件存取功能，操作系统使用的命令不同。
 - 访问控制方法不同。



6.2.2 FTP 的基本工作原理

- 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。



FTP服务器主进程的工作步骤

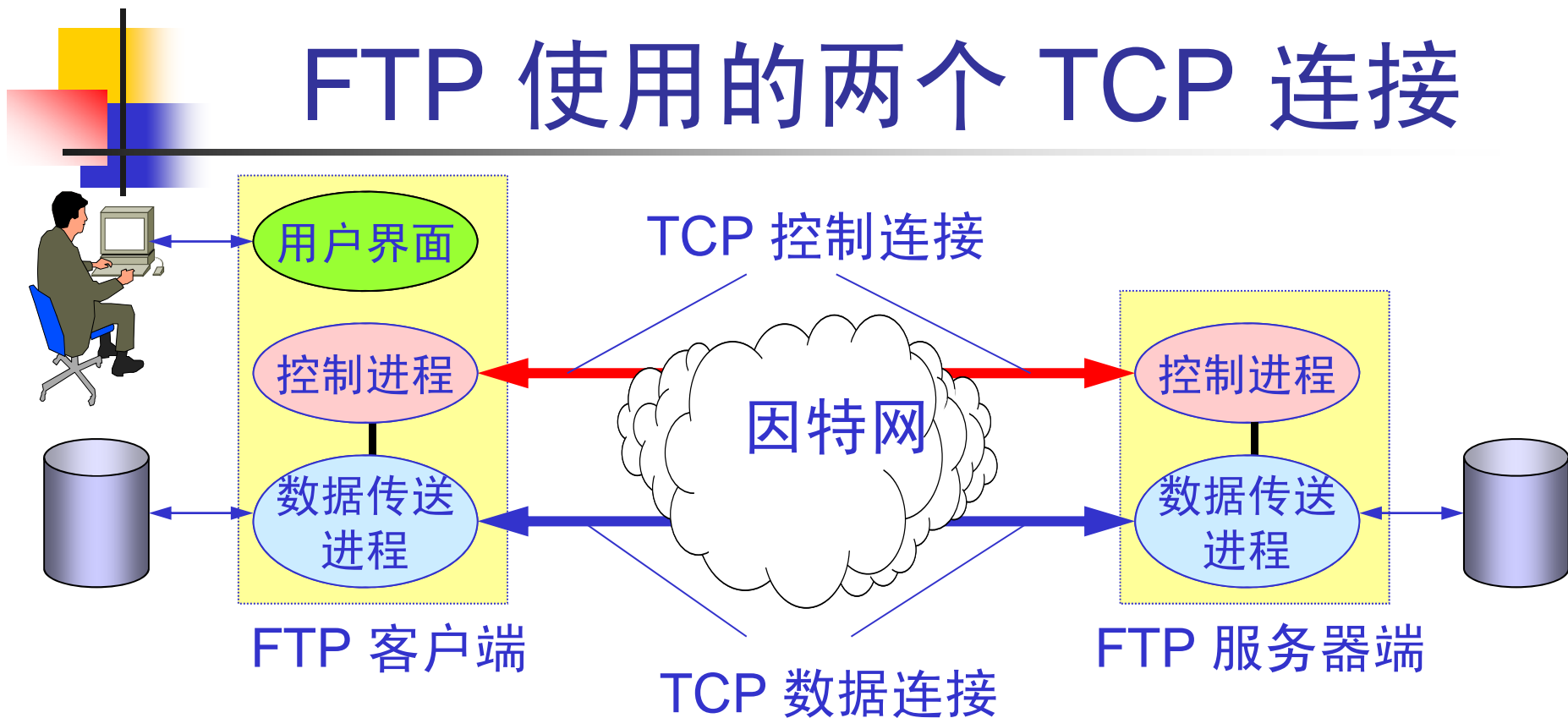
- 打开熟知端口（端口号为 21），使客户进程能够连接上。
- 等待客户进程发出连接请求。
- 启动从属进程来处理客户进程发来的请求。
 - 从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。
- 回到等待状态，继续接受其他客户进程发来的请求。
 - 主进程与从属进程的处理是并发地进行。



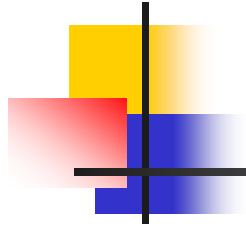
两个连接

- **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 使用的两个 TCP 连接



- 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码，用于建立数据传送连接。
- 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码建立数据传送连接。



使用两个不同端口号的好处

- 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）。



NFS (Net File System)

- NFS 允许应用进程打开一个远地文件，并能在该文件的某一个特定的位置上开始读写数据。
- NFS 可使用户只复制一个大文件中的一个很小的片段，而不需要复制整个大文件。
 - 计算机 A 的 NFS 客户软件，把要添加的数据和在文件后面写数据的请求一起发送到远地的计算机 B 的 NFS 服务器。
 - NFS 服务器更新文件后返回应答信息。
 - 在网络上传送的只是少量的修改数据。



6.3 按远程终端协议 TELNET

- TELNET 是一个简单的远程终端协议，也是因特网的正式标准。
- 用户用 TELNET 就可在其所在地通过 TCP 连接注册（即登录）到远地的另一个主机上（使用主机名或 IP 地址）。
- TELNET 能将用户的击键传到远地主机，同时也能将远地主机的输出通过 TCP 连接返回到用户屏幕。
- 这种服务是透明的，因为用户感觉到好像键盘和显示器是直接连在远地主机上。

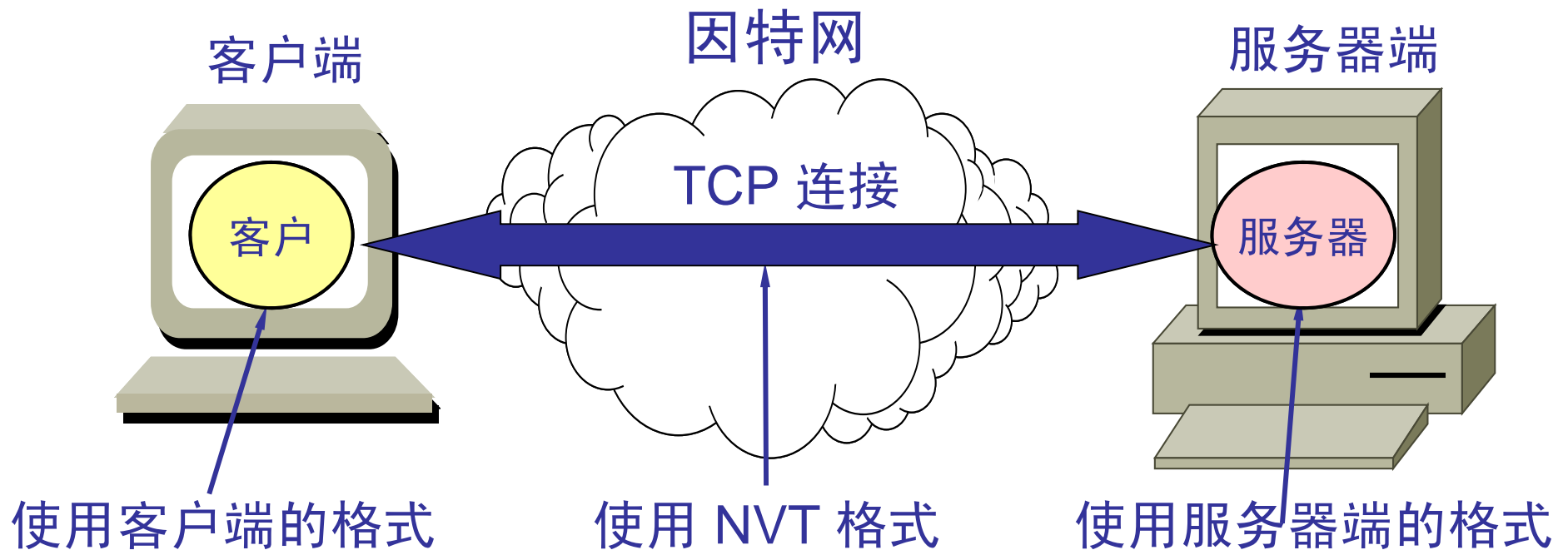


客户服务器方式

- 现在由于 PC 机的功能越来越强，用户已较少使用 TELNET 了。
- TELNET 也使用客户服务器方式。
 - 在本地系统运行 TELNET 客户进程，而在远地主机则运行 TELNET 服务器进程。
- 和 FTP 的情况相似，服务器中的主进程等待新的请求，并产生从属进程来处理每一个连接。

TELNET 使用

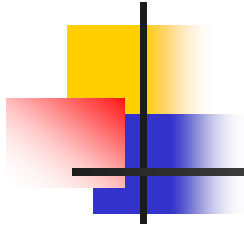
网络虚拟终端 NVT 格式





BBS(Bulletin Board System)

- BBS向用户提供了一块公共电子白板，每个用户都可以在上面发布信息或提出看法。早期的BBS由教育机构或研究机构管理。
- 国内著名论坛：
 - 技术论坛
 - 水木清华 smth.org.cn
 - 饮水思源 bbs.sjtu.edu.cn
 - 程序员社区 csdn.net
 - 热点论坛
 - 天涯 www.tianya.cn-天涯煮酒



6.5 电子邮件

6.5.1 概述

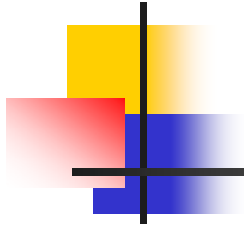
- **电子邮件**(e-mail)是因特网上使用得最多的和最受用户欢迎的一种应用。
- 电子邮件把邮件发送到收件人使用的邮件服务器，并放在其中的收件人邮箱中，收件人可随时上网到自己使用的邮件服务器进行读取。
- 电子邮件不仅使用方便，而且还具有传递迅速和费用低廉的优点。
- 现在电子邮件不仅可传送文字信息，而且还可附上声音和图像。

Hotmail的故事



- 1995年12月，Sabeer Bhatia和Jack Smith拜访因特网风险投资人D. F. Jurvetson，建议开发免费的基于Web的电子邮件系统。
- 1996年7月，3个全职员工和12~14个兼职人员（为自己的股份工作）开发了Hotmail服务。
- 1996年8月，100K用户
- 1997年12月，超过12M用户，4亿美元被微软收购。
- Hotmail的成功在于：“先行者优势”和“病毒行销”

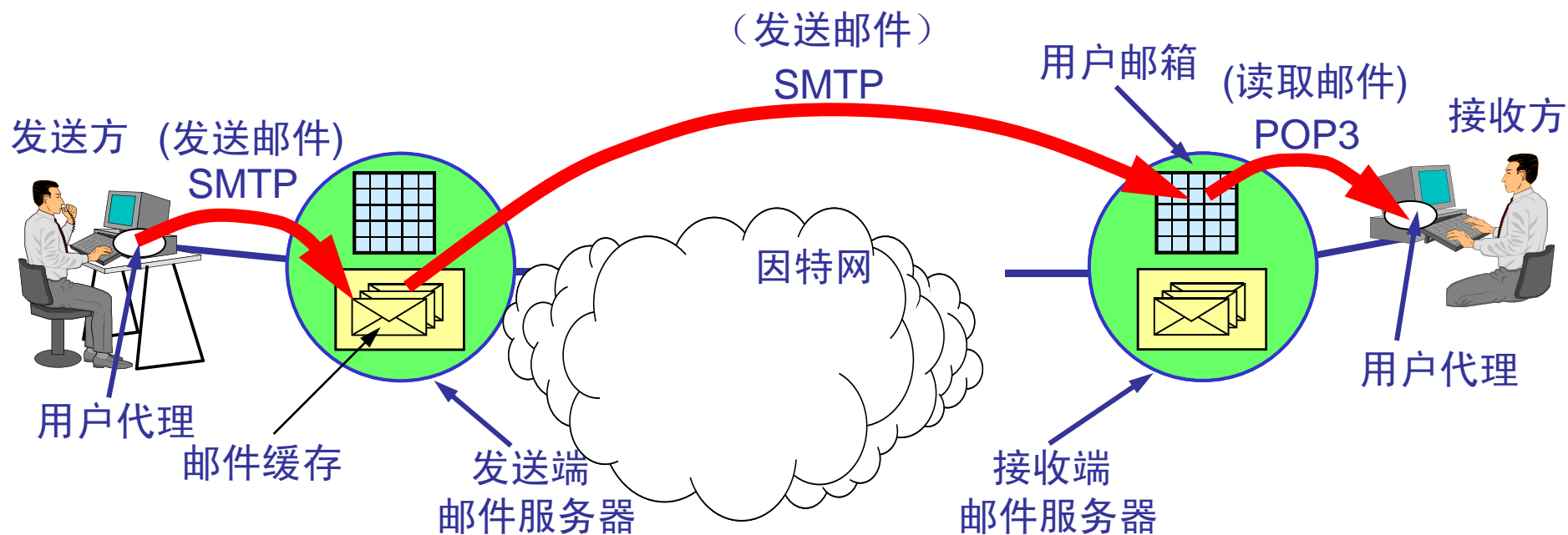


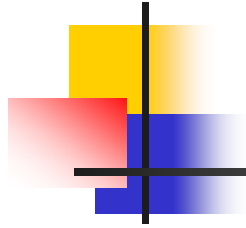


电子邮件的一些标准

- 发送邮件的协议：SMTP
- 读取邮件的协议：POP3 和 IMAP
- 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。

电子邮件的最主要的组成构件

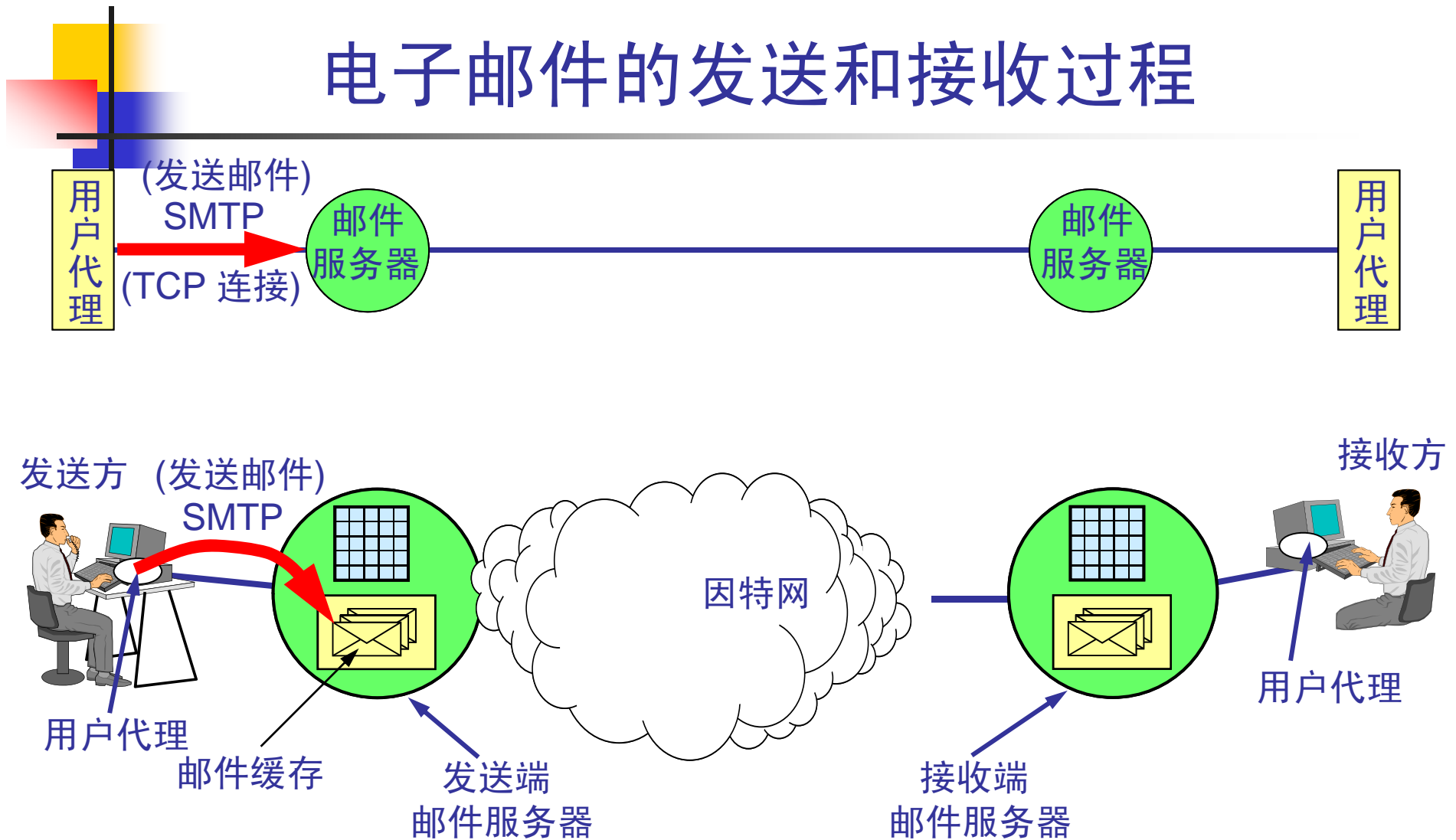




用户代理 UA (User Agent)

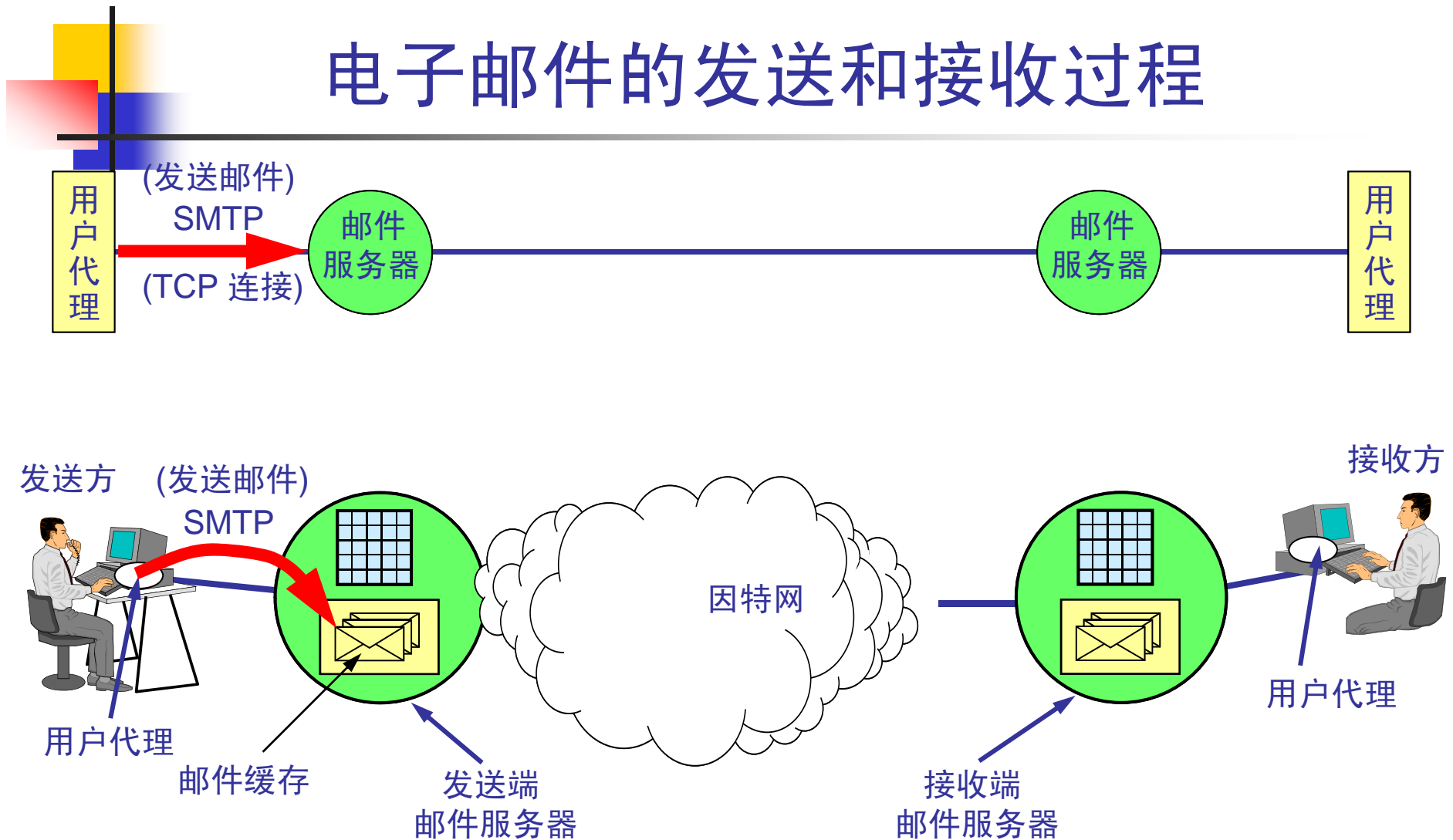
- 用户代理 UA 就是用户与电子邮件系统的接口，是电子邮件客户端软件。
- 用户代理的功能：撰写、显示、处理和通信。
- 邮件服务器的功能是发送和接收邮件，同时还要向发信人报告邮件传送的情况（已交付、被拒绝、丢失等）。
- 邮件服务器按照客户服务器方式工作。邮件服务器需要使用发送和读取两个不同的协议。

电子邮件的发送和接收过程



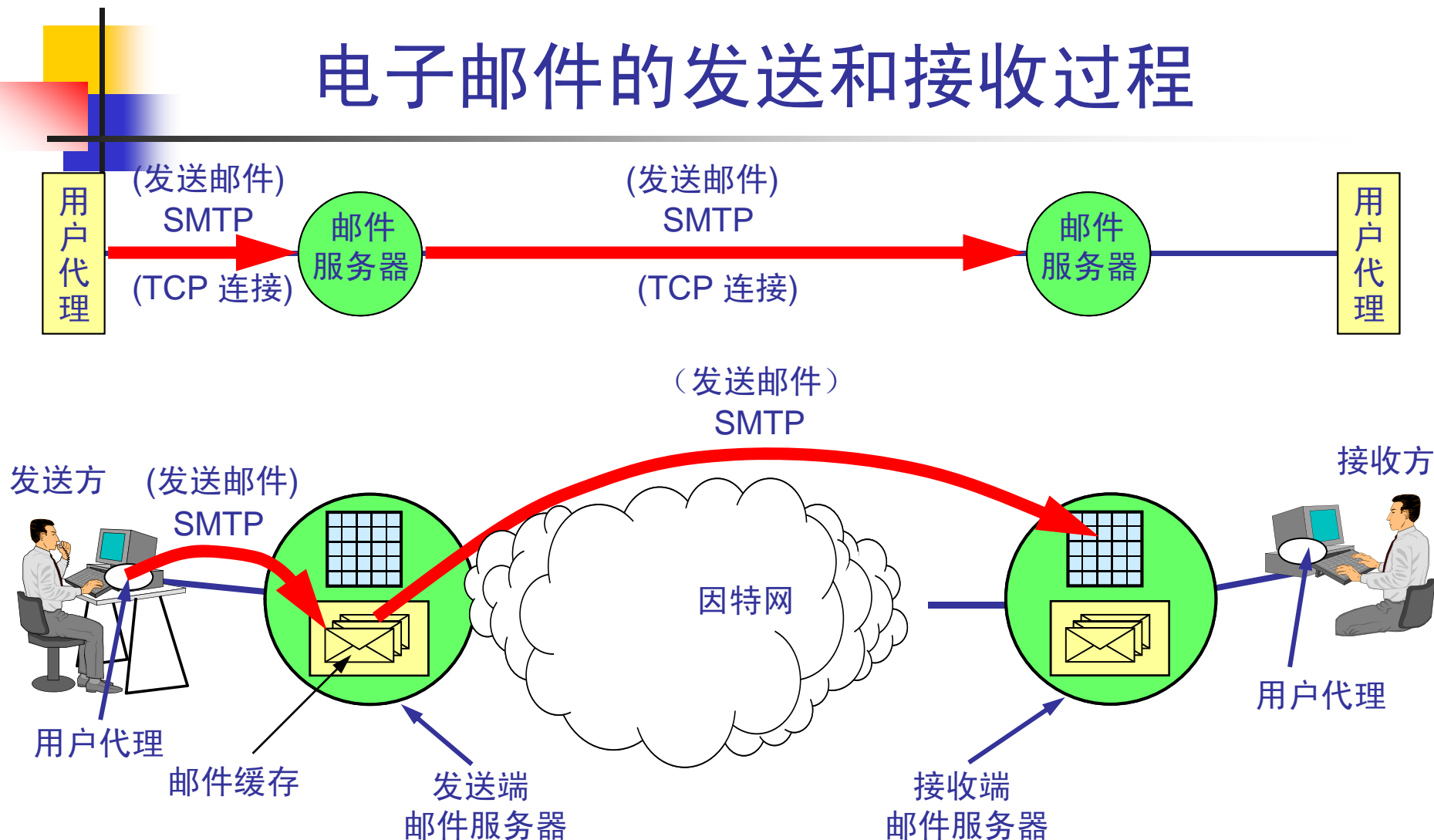
- (1) 发信人调用用户代理来编辑要发送的邮件。
用户代理用 SMTP 把邮件传送给发送端邮件服务器。

电子邮件的发送和接收过程



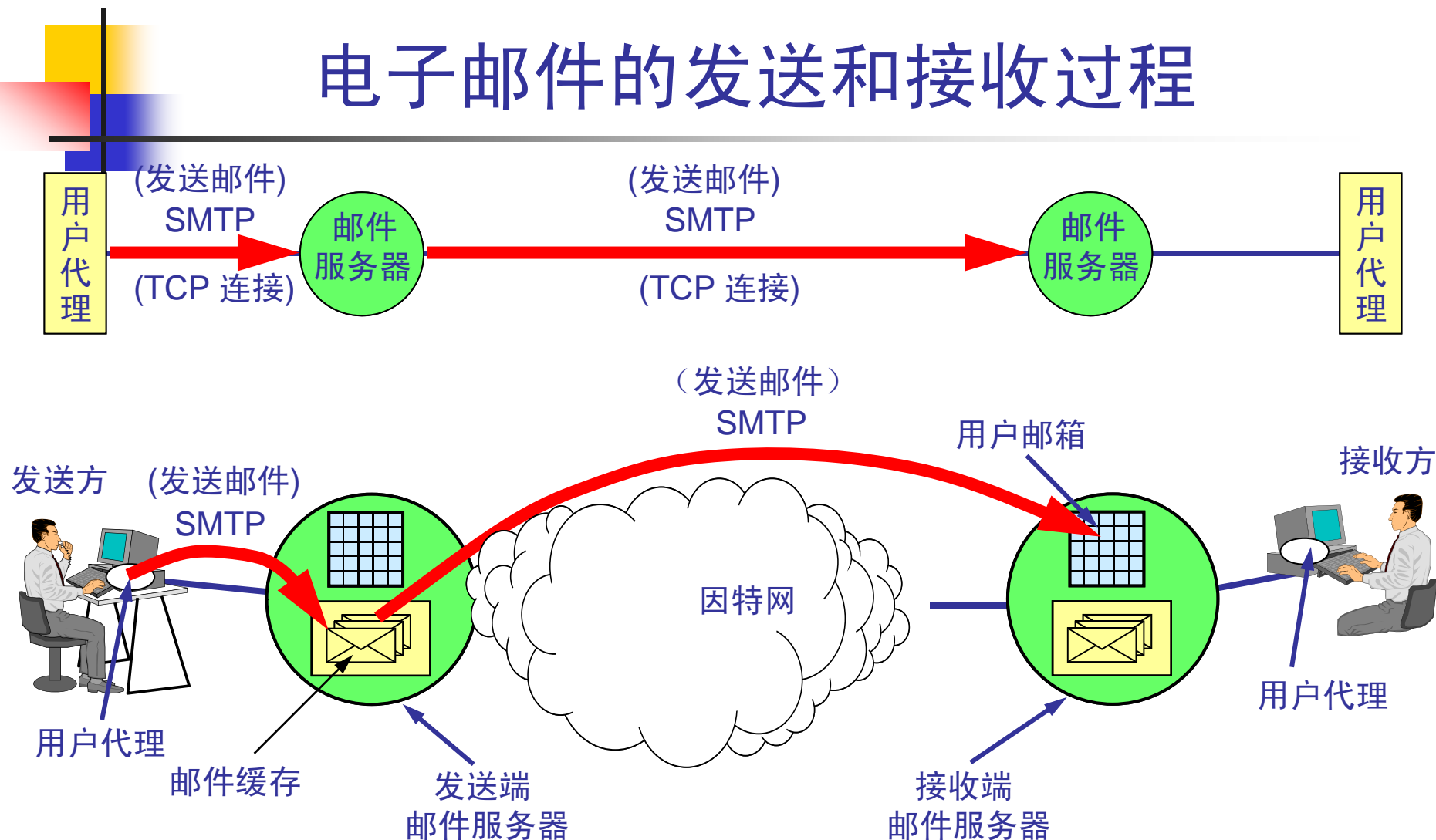
(2) 发送端邮件服务器将邮件放入邮件缓存队列中，等待发送。

电子邮件的发送和接收过程



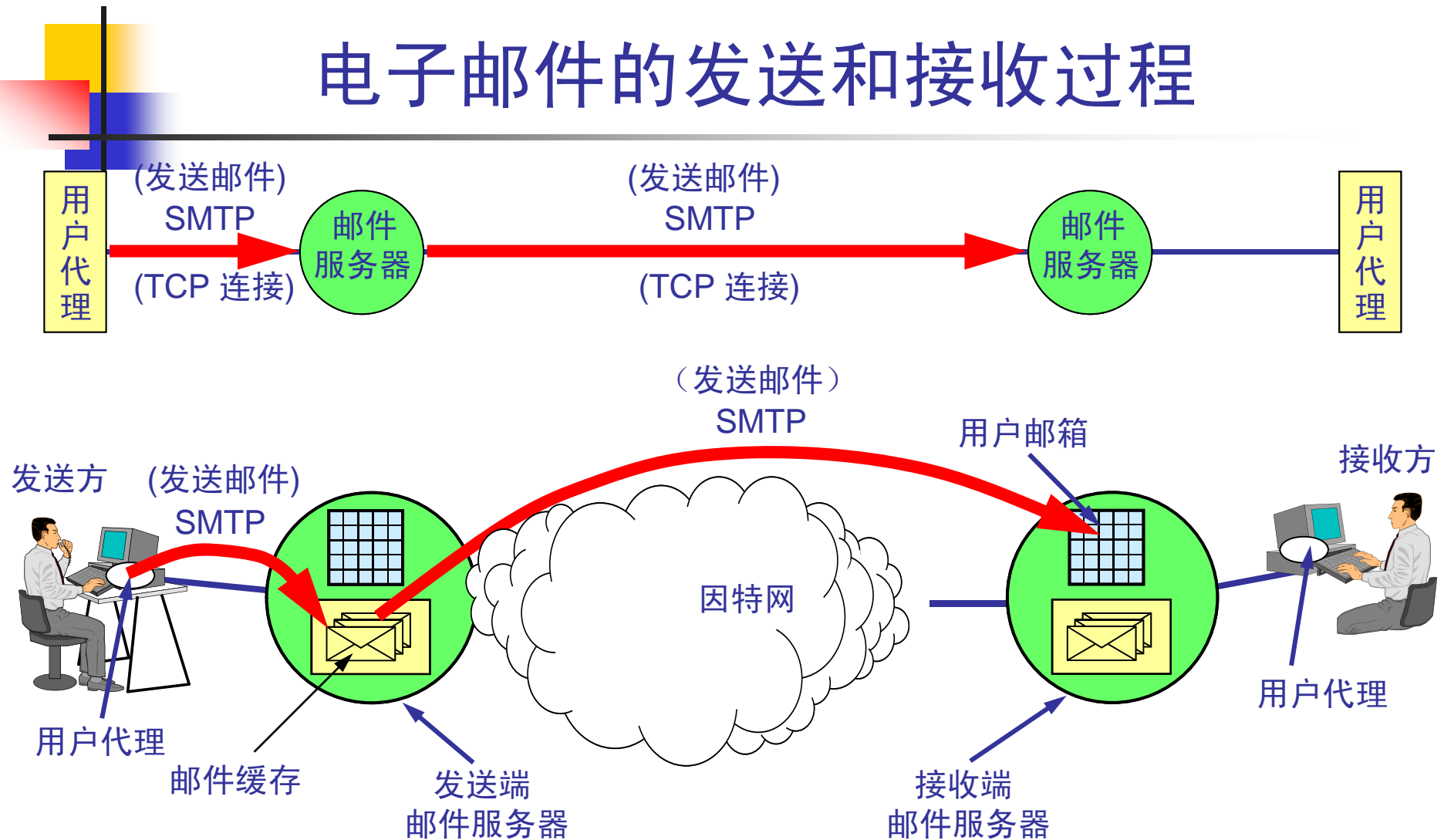
(3) 运行在发送端邮件服务器的 SMTP 客户进程，发现在邮件缓存中有待发送的邮件，就向运行在接收端邮件服务器的 SMTP 服务器进程发起 TCP 连接的建立。

电子邮件的发送和接收过程



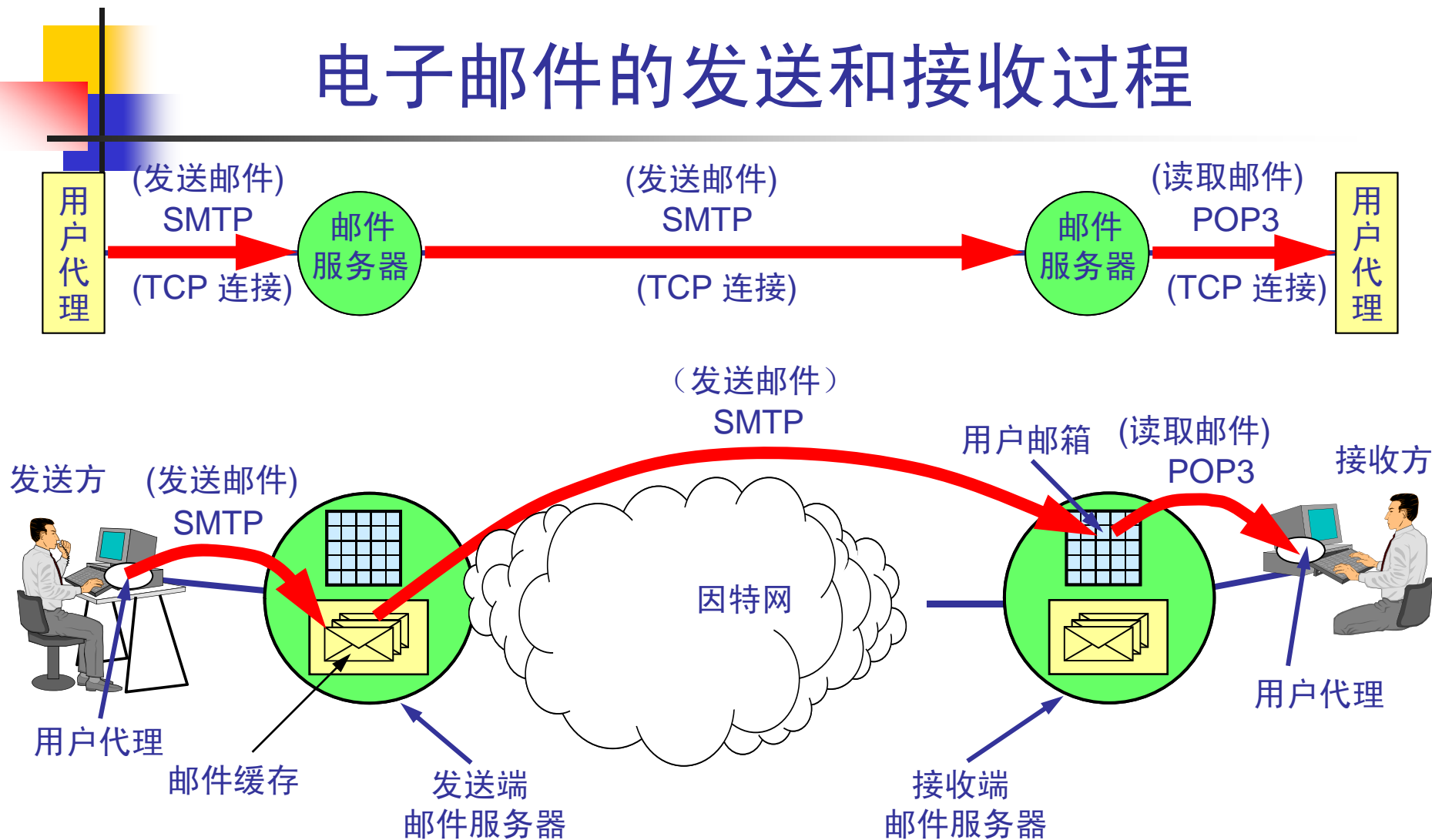
(4) TCP 连接建立后，SMTP 客户进程开始向远程的 SMTP 服务器进程发送邮件。当所有的待发送邮件发完了 SMTP 就关闭所建立的 TCP 连接。

电子邮件的发送和接收过程



(5) 运行在接收端邮件服务器中的 SMTP 服务器进程收到邮件后，将邮件放入收信人的用户邮箱中，等待收信人在方便时进行读取。

电子邮件的发送和接收过程



(6) 收信人在打算收信时，调用用户代理，使用 POP3（或 IMAP）协议将自己的邮件从接收端邮件服务器的用户邮箱中的取回（如果邮箱中有来信的话）。



电子邮件的组成

- 电子邮件由信封(envelope)和内容(content)两部分组成。
- 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。
- 在邮件的信封上，最重要的就是收件人的地址。



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 (6-1)

- 符号 “@” 读作 “at”，表示 “在” 的意思。
- 例如，电子邮件地址 hongfeng@ouc.edu.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



6.5.2 简单邮件传送协议 SMTP

- SMTP 所规定的就是在两个相互通信的 SMTP 进程之间应如何交换信息。
- SMTP 使用客户服务器方式：
 - 负责发送邮件的 SMTP 进程就是 SMTP 客户
 - 负责接收邮件的 SMTP 进程就是 SMTP 服务器。
- SMTP通信过程：
 - 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。
 - 邮件传送
 - 连接释放：邮件发送完毕后，SMTP 应释放 TCP 连接。



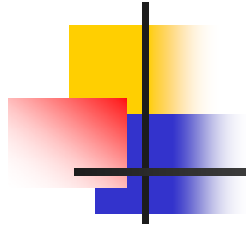
6.5.3 电子邮件的信息格式

- 一个电子邮件分为信封和内容两大部分。
- RFC 822 只规定了邮件内容中的首部(header)格式，而对邮件的主体(body)部分则让用户自由撰写。
- 用户写好首部后，邮件系统将自动地将信封所需的信息提取出来并写在信封上。所以用户不需要填写电子邮件信封上的信息。
- 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：To 和 Subject。



邮件内容的首部

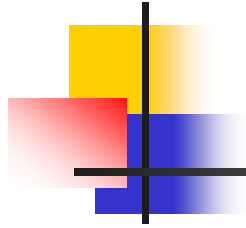
- “To:”后面填入一个或多个收件人的电子邮件地址。用户只需打开地址簿，点击收件人名字，收件人的电子邮件地址就会自动地填入到合适的位置上。
- “Subject:”是邮件的主题。它反映了邮件的主要内容，便于用户查找邮件。
- 抄送 “Cc:”表示应给某某人发送一个邮件副本。
- “From”和“Date”表示发信人的电子邮件地址和发信日期。“Reply-To”是对方回信所用的地址。



6.5.4 邮件读取协议

POP3 和 IMAP

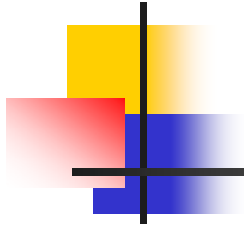
- 邮局协议 POP 是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 POP3。
- POP 也使用客户服务器的工作方式。
- 在接收邮件的用户 PC 机中必须运行 POP 客户程序，而在用户所连接的 ISP 的邮件服务器中则运行 POP 服务器程序。



IMAP 协议

(Internet Message Access Protocol)

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

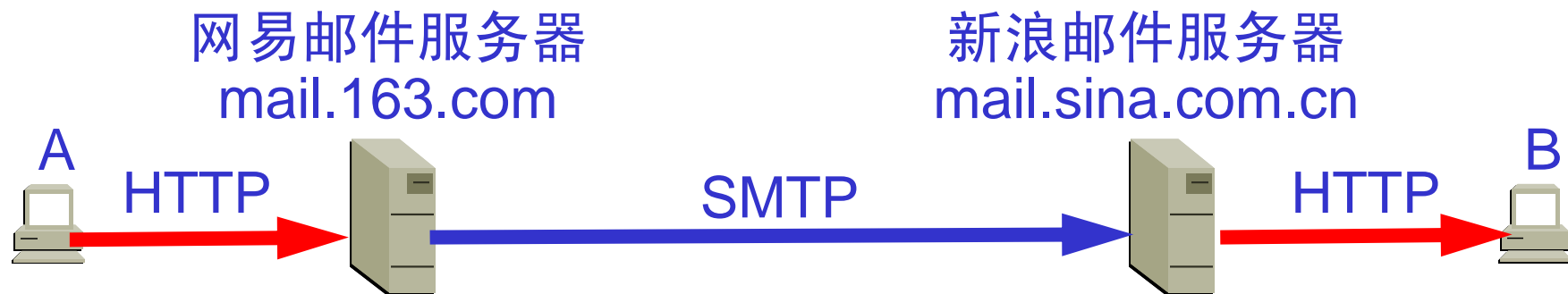


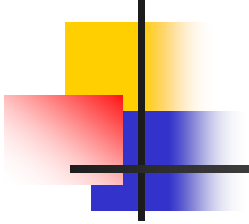
IMAP 的特点

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

6.5.5 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。





6.5.6 通用因特网邮件扩充 MIME

MIME 概述

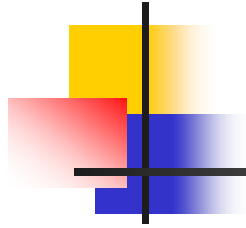
SMTP 有以下缺点：

- SMTP 不能传送可执行文件或其他的二进制对象。
- SMTP 限于传送 7 位的 ASCII 码。许多其他非英语国家的文字（如中文、俄文，甚至带重音符号的法文或德文）就无法传送。
- SMTP 服务器会拒绝超过一定长度的邮件。
- 某些 SMTP 的实现并没有完全按照[RFC 821]的 SMTP 标准。

MIME 的特点

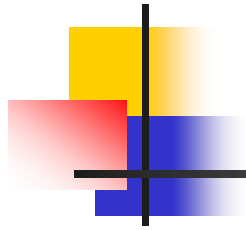
- MIME 并没有改动 SMTP 或取代它。
- MIME 的意图是继续使用目前的SMTP报文格式，但增加了邮件主体的结构，即定义了传送非 ASCII 码的编码规则。





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-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。
- Content-Id: 邮件的唯一标识符。
- Content-Transfer-Encoding: 在传送时邮件的主体是如何编码的。
- Content-Type: 说明邮件的性质，即邮件对象如何操作。

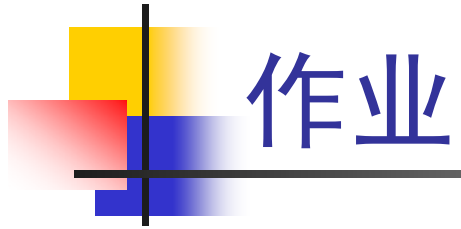


MIME 举例

MIME version
method used
to encode data
multimedia data
type, subtype,
parameter declaration
encoded data

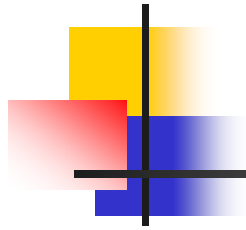
```
From: hongfeng@ouc.edu.cn
To: tangrc@ouc.edu.cn
Subject: Picture of Keith Ross.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....
.....base64 encoded data
```



作业

- 6-3、5、7、8、12、20、23



信息检索评价说明图—画图材料

