广东开放大学
The Open University of Guangdong

网络应用

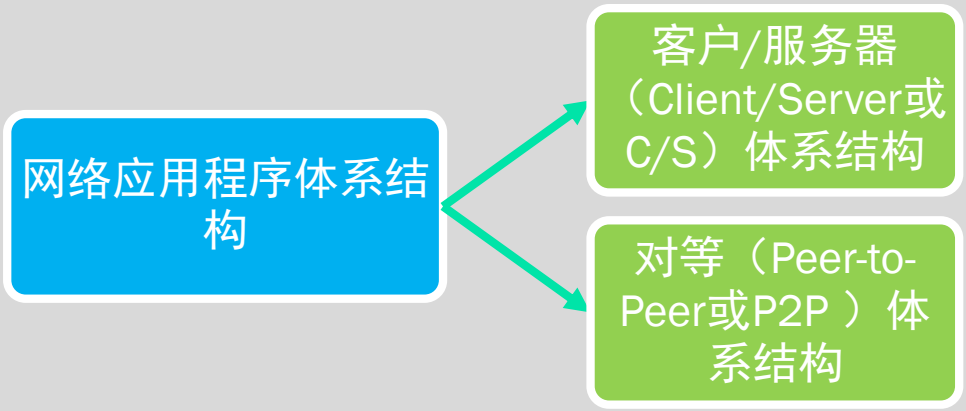
谢剑刚
广东开放大学

6.1 应用层概述

6.1.1 网络应用程序体系结构

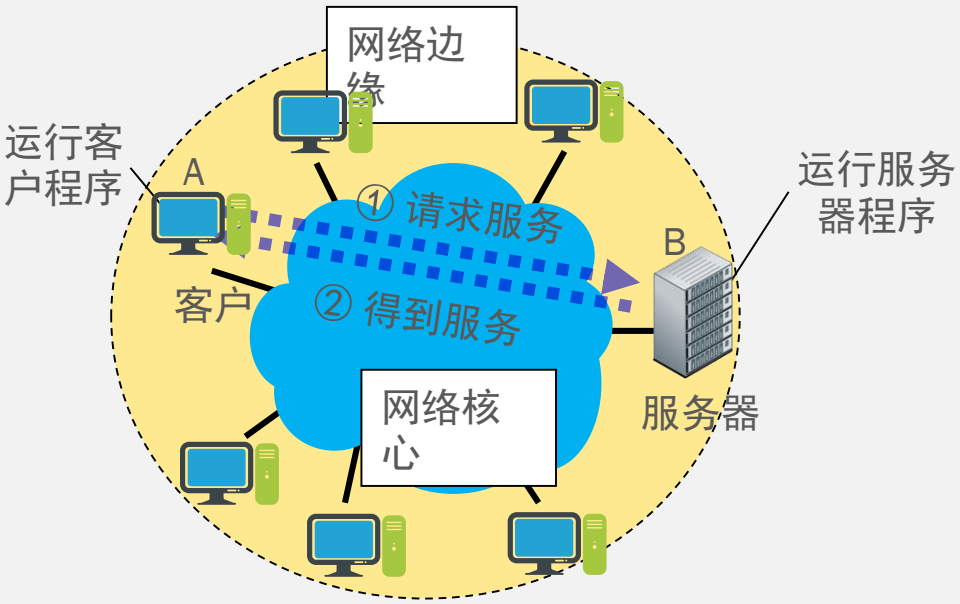
网络应用程序运行在网络边缘的端系统上，通过彼此间的通信来共同完成某项任务。

网络应用程序在各种端系统上的组织方式和它们之间的关系，即网络应用程序体系结构：



1. 客户/服务器体系结构

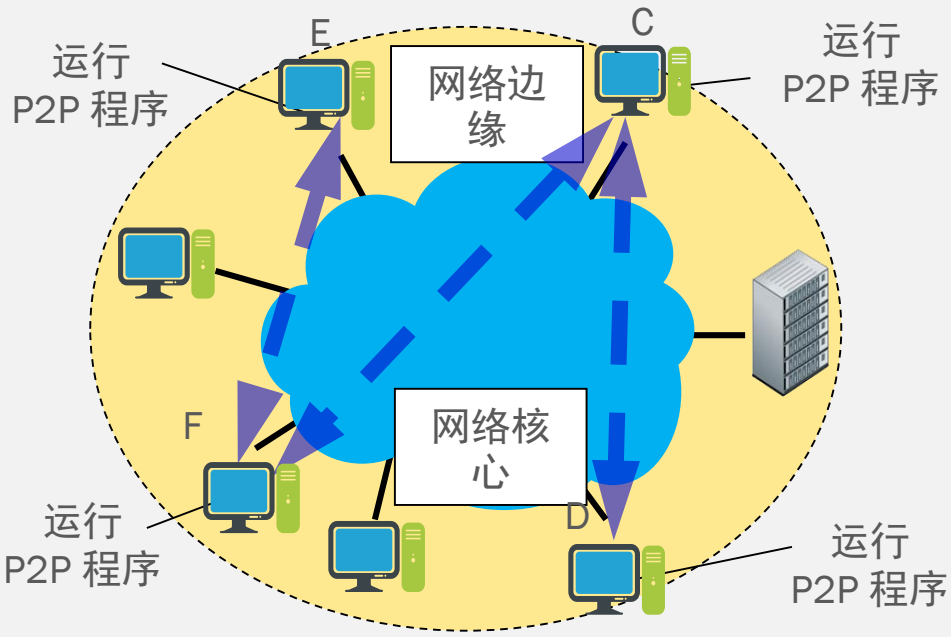
- **客户(client)和服务器(server)都是指通信中所涉及的两个应用进程。**
- **客户服务器方式所描述的是进程之间服务和被服务的关系。**
- **客户是服务的请求方，服务器是服务的提供方。客户相互之间不直接进行通信。**
- **服务器具有固定的IP地址和端口号，并且总是处于运行状态，并等待客户的服务请求。**



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。

2. P2P对等体系结构

- 在对等(P2P, Peer-to-Peer)方式中，没有固定的服务请求者和服务提供者，应用进程是对等的，被称为对等方（peer）。
- 对等方相互之间直接通信，每个对等方即是服务的请求者，又是服务的提供者。
- 基于P2P的应用是服务分散型的，因为服务不是集中在少数几个服务器计算机中，而是分散在大量对等方计算机中。
- 对等方式的应用如：BT文件下载软件



对等方相互之间直接通信，每个对等方即是服务的请求者，又是服务的提供者

6.1.2 应用层协议

- 为实现某种网络应用，不论是客户进程和服务器进程之间，还是对等方之间，都需要遵循的某种通信协议——应用层协议
- 可以是公开的，如RFC文档定义的因特网公共领域的应用层协议，HTTP，FTP等
- 还有很多其他应用的应用层协议不是公开的，而是专用的，如很多P2P应用

6.1.3 选择运输层协议

应用	应用层协议	运输层协议
电子邮件	SMTP	TCP
远程终端访问	TELNET	TCP
万维网	HTTP	TCP
文件传送	FTP	TCP
IP电话	专用协议	通常用UDP
流式多媒体通信	专用协议	UDP或TCP

6.2.1 域名系统概述

- **域名系统DNS (Domain Name System)并不是直接和用户打交道的网络应用。相反，DNS为其他各种网络应用提供一种核心服务，即名字服务，用来把计算机的名字转换为对应的IP地址。**
- **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。**

- 除了进行主机名到IP地址的转换外，DNS还提供了一些重要的服务：

01 OPTION 主机别名

02 OPTION 负载分配

03 OPTION 反向域名解析

6.2.2 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：

- 各标号分别代表不同级别的域名。

域名只是个逻辑概念

01 OPTION

域名只是个逻辑概念，并不代表计算机所在的物理地点。

02 OPTION

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。

03 OPTION

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)

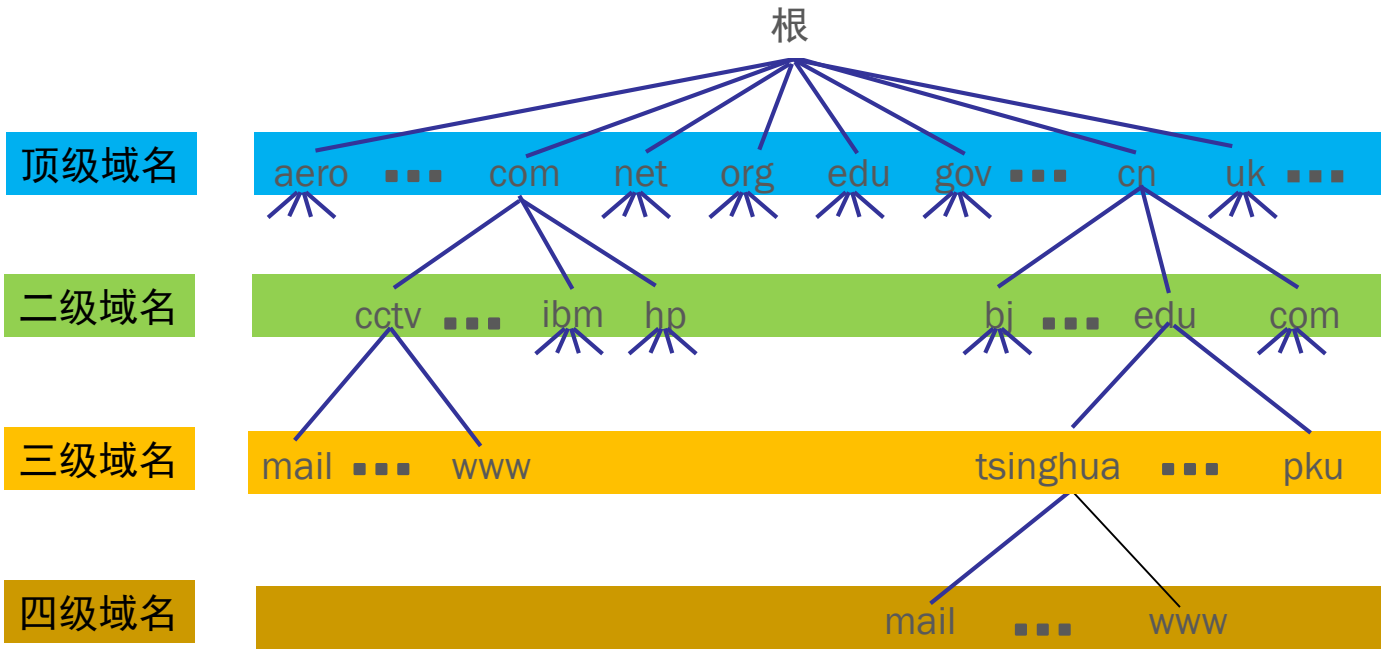
(1) 国家顶级域名 nTLD：如: .cn 表示中国，.us 表示美国，.uk 表示英国，等等。

(2) 通用顶级域名 gTLD：最早的顶级域名是：

.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)**

(3) 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。

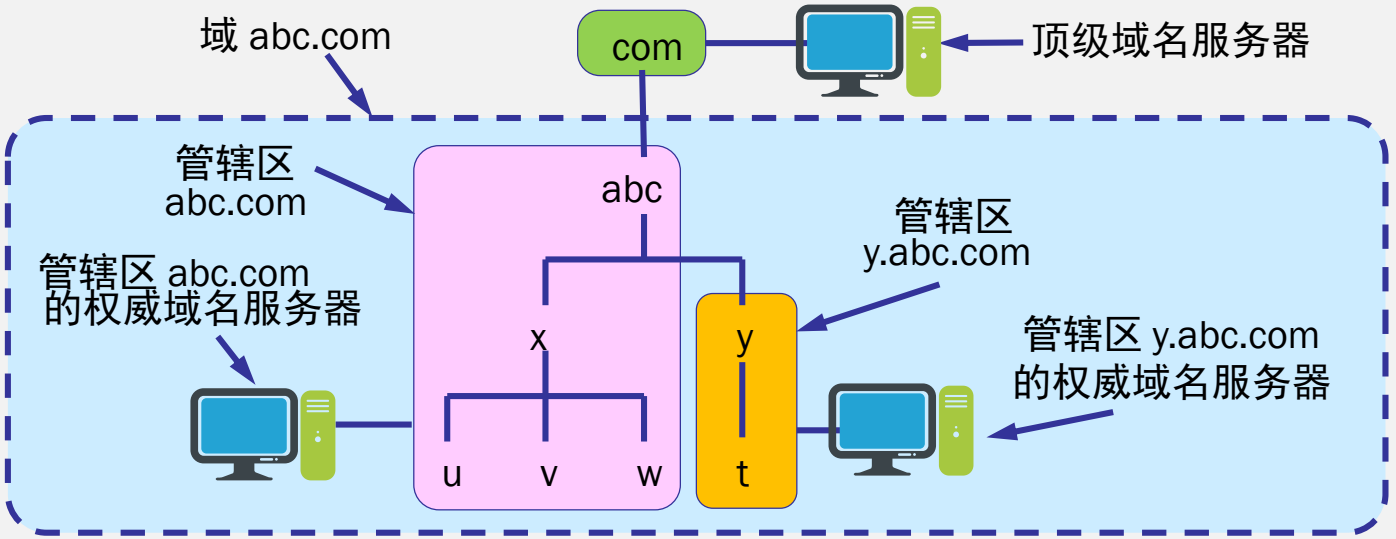
因特网的域名空间


根
顶级域名
aero ... com net org edu gov ... cn uk ...
二级域名
cctv ... ibm hp bj ... edu com
三级域名
mail ... www tsinghua ... pku
四级域名
mail ... www

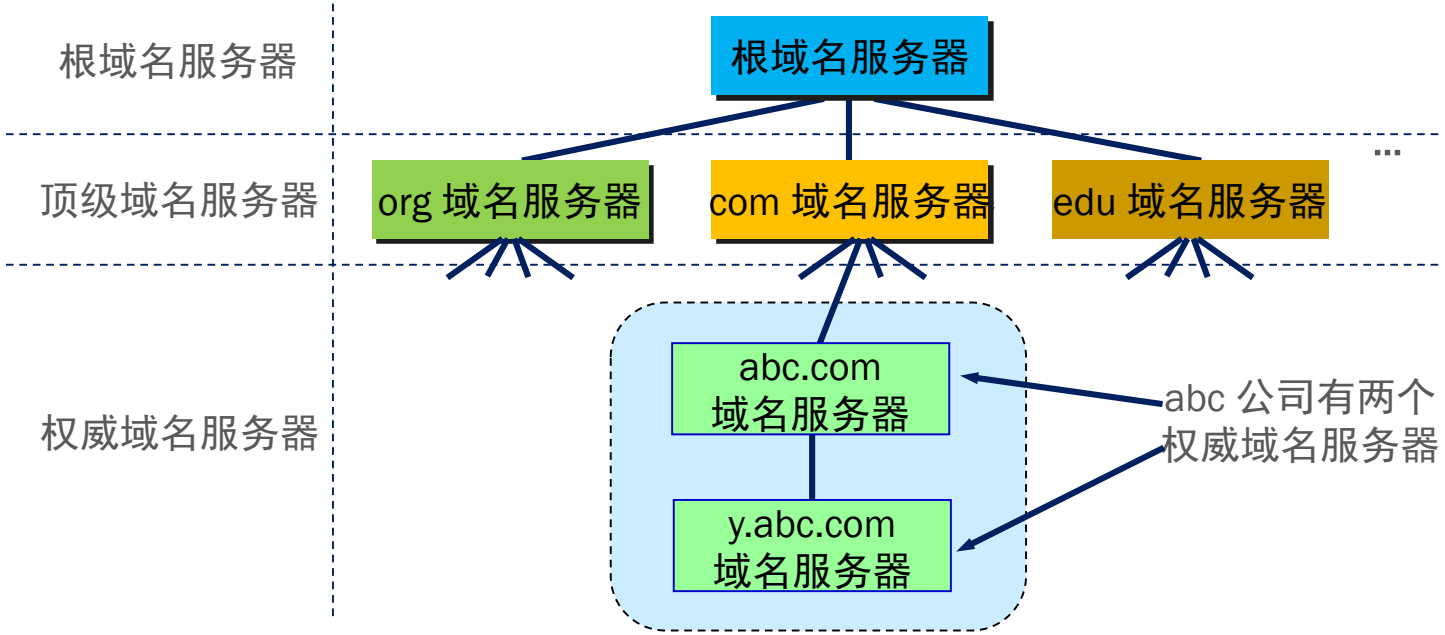
6.2.3 域名服务器

- 域名服务器实现域名和IP地址之间映射
- 一个服务器所负责管辖的（或有权限的）范围叫做区(zone)。
- 每一个区设置相应的权威域名服务器，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。

权威域名服务器和管辖区



DNS域名服务器的等级结构



域名服务器有以下四种类型

01 根域名服务器

02 顶级域名服务器

03 权威域名服务器

04 本地域名服务器

根域名服务器

——最高层次的域名服务器——

- 这是最高层次的域名服务器。
- 根域名服务器并不直接管辖某个区的域名信息，但每个根域名服务器都知道所有的顶级域名服务器的域名及其IP地址。
- 在因特网上共有13个不同IP地址的根域名服务器。尽管我们将这13个根域名服务器中的每一个都视为单个的服务器，但每台“服务器”实际上是由许多分布在世界各地的计算机构成的服务器群集。
- 当本地域名服务器向根域名服务器发出查询请求时，路由器就把查询请求报文转发到离这个DNS客户最近的一个根域名服务器。
- 根域名服务器通常并不直接对域名进行解析，而是返回该域名所属顶级域名的顶级域名服务器的IP地址。

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）。

权威域名服务器

- 负责管理某个区的域名服务器。
- 每一个主机的域名都必须在某个权威域名服务器处注册登记。
- 因此权威域名服务器知道其管辖的域名与IP地址的映射关系。
- 另外，权威域名服务器还知道其下级域名服务器的地址。

本地域名服务器

- 当一个主机发出DNS查询报文时，这个查询报文就首先被送往该主机的本地域名服务器。本地域名服务器起着DNS代理的作用。
- 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**。

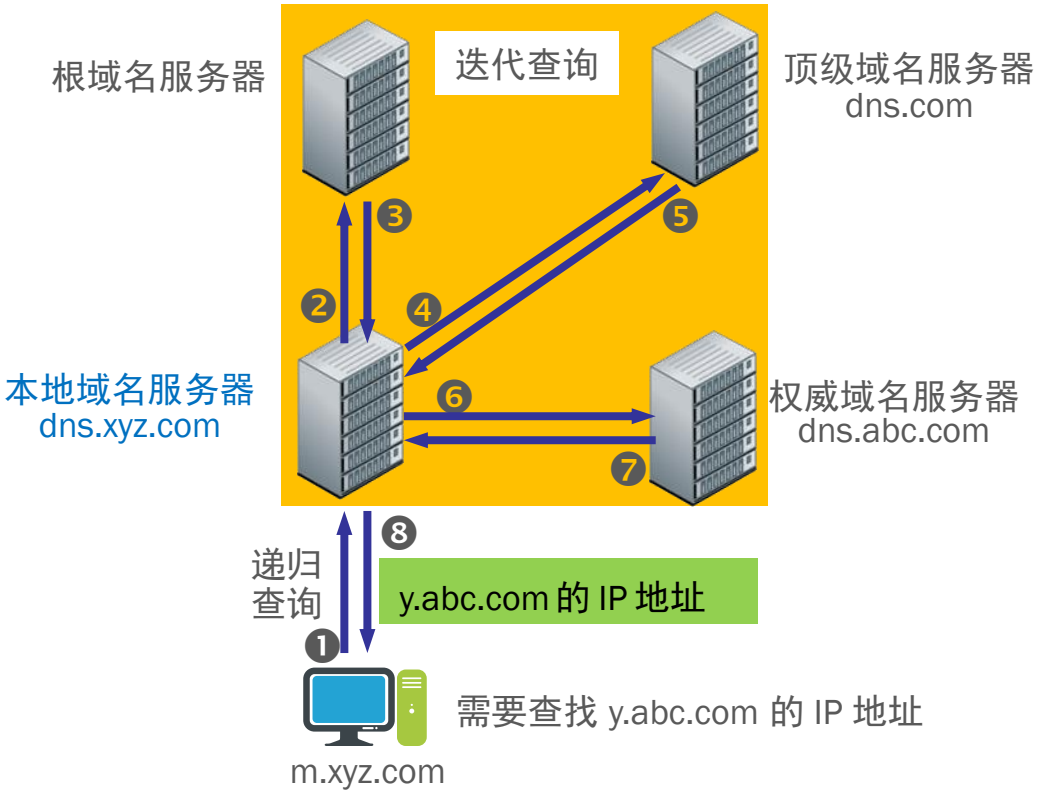
提高域名服务器的可靠性

- 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。

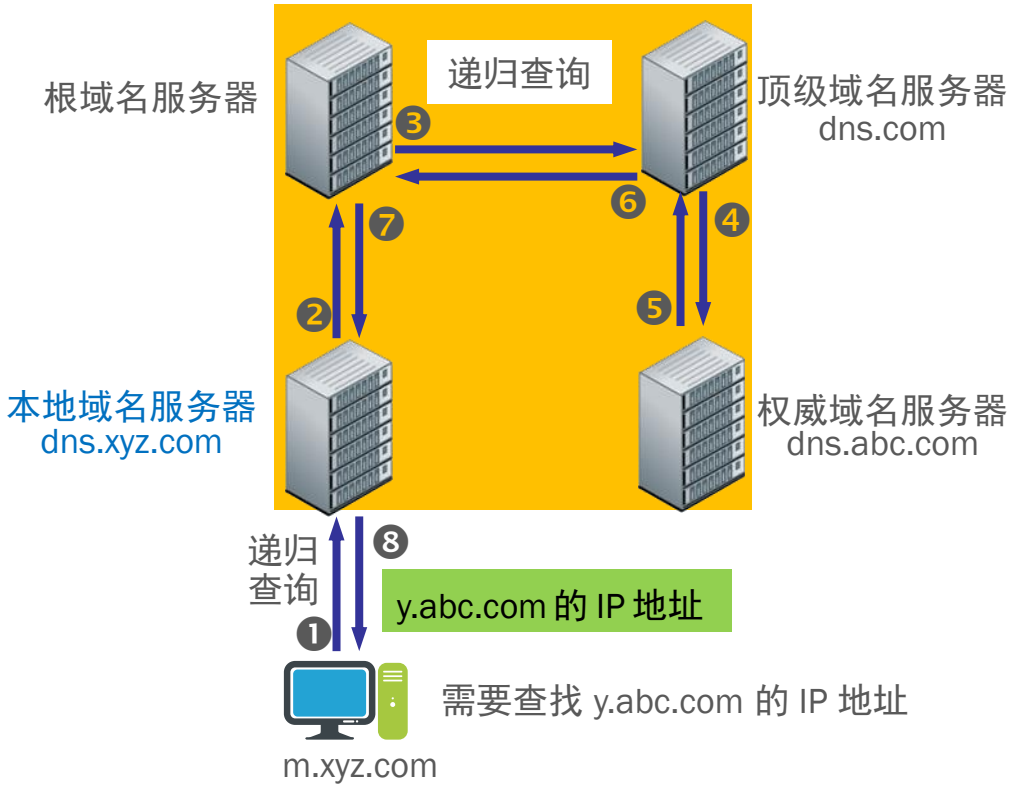
6.2.4 域名的解析过程

- 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



本地域名服务器采用递归查询（比较少用）



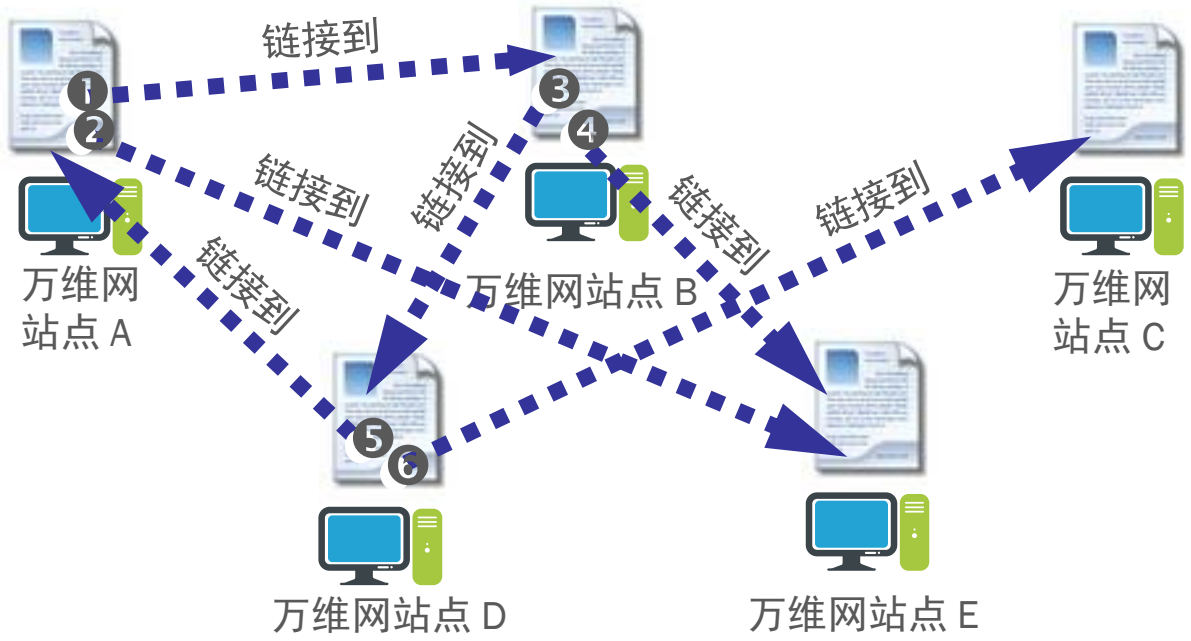
名字的高速缓存

- 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。
- 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）。
- 当权威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。

6.3.1 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


链接到
链接到
链接到
链接到
链接到
链接到
1
2
3
4
5
6
万维网
站点 A
万维网站点 B
万维网
站点 C
万维网站点 D
万维网站点 E

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)。

万维网必须解决的问题

(1) 怎样标志分布在整个因特网上的万维网文档？

- 使用统一资源定位符 URL (Uniform Resource Locator)来标志万维网上的各种文档。
- 使每一个文档在整个因特网的范围内具有唯一的标识符 URL。

(2) 用何协议实现Web页面的传输？

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

(3) 如何编写万维网文档使它们能在因特网上的各种计算机上显示出来，如何在文档中嵌入超链？

- 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）。

6.3.2 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 给资源的位置提供一种抽象的识别方法，并用这种方法给资源定位。
- 只要能够对资源定位，系统就可以对资源进行各种操作，如存取、更新、替换和查找其属性。
- URL 相当于一个文件名在网络范围的扩展。因此 URL 是与因特网相连的机器上的任何可访问对象的一个指针。

URL 的一般形式

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

URL 的一般形式是：

<协议>://<主机>:<端口>/<路径>

- 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

2. 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

2. 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

2. 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

2. 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

2. 使用 HTTP 的 URL

使用 HTTP 的 URL 的一般形式

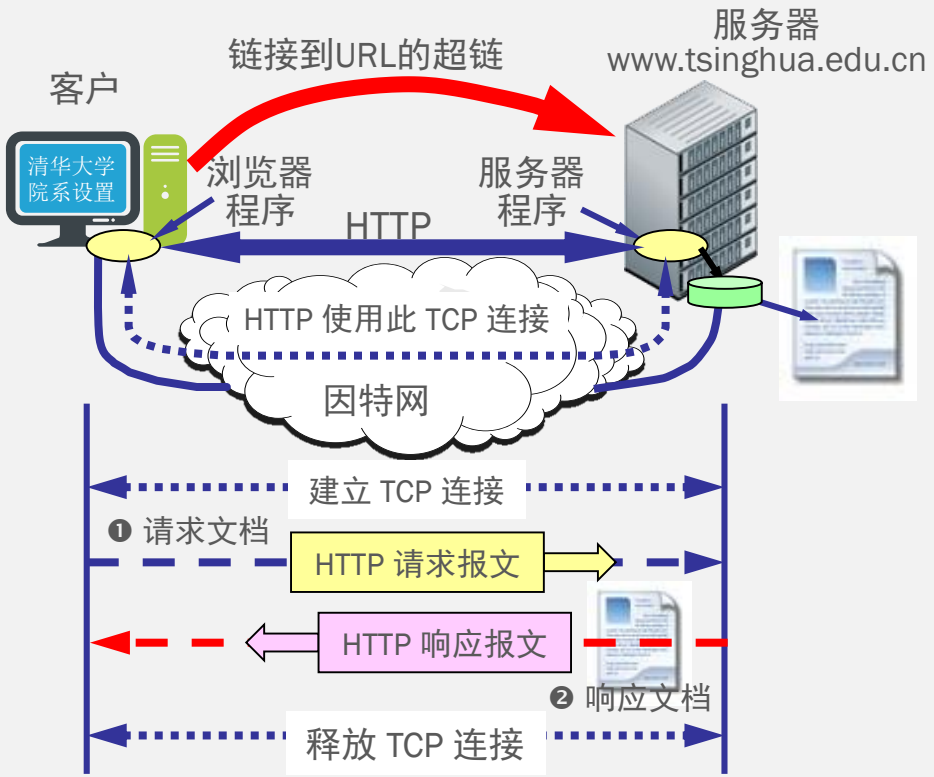
- http://<主机>:<端口>/<路径>

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)。

6.3.3 超文本传送协议 HTTP

- **1. HTTP 的操作过程**
  - **HTTP协议定义了浏览器（即万维网客户进程）怎样向万维网服务器请求万维网文档，以及万维网服务器怎样把万维网文档传送给浏览器。**
  - **HTTP使用的运输层协议是TCP，默认端口号是80。**

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.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/yxsz/index.htm。

(6) • 服务器给出响应，把文件 index.htm 发给浏览器。

(7) • 释放TCP连接。

(8) • 浏览器显示“清华大学院系设置”文件 index.htm 中的所有文本。

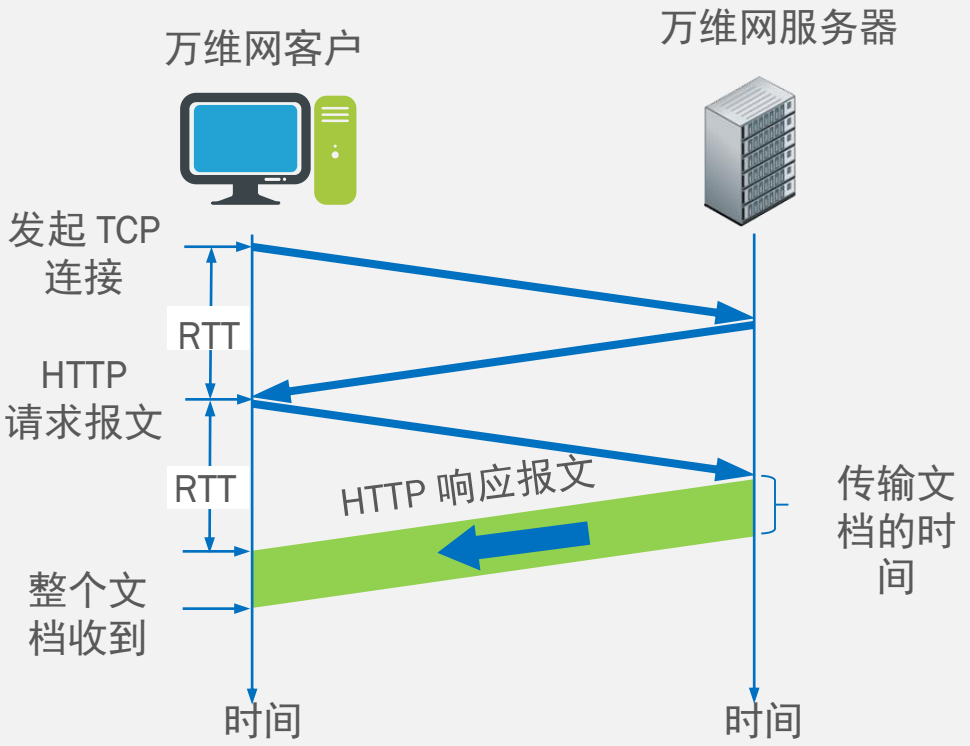
HTTP 的主要特点

- HTTP 1.0 协议是无状态的(stateless)。HTTP不要求服务器保留客户的任何状态信息。
- 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。

2. 非持续连接与持续连接

- HTTP/1.0 协议使用非持续连接。
- 客户（浏览器）每发送一个请求，Web服务器在发送响应后就关闭这条连接
- 要向同服务器发送下一个请求需再建立TCP连接
- 一个Web网页除了一个基本的HTML文档外，可能还包括多个要在页面上显示或表现的引用对象（图片、声音等）
- 请求一个网页可能要建立多次连接，效率太低

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

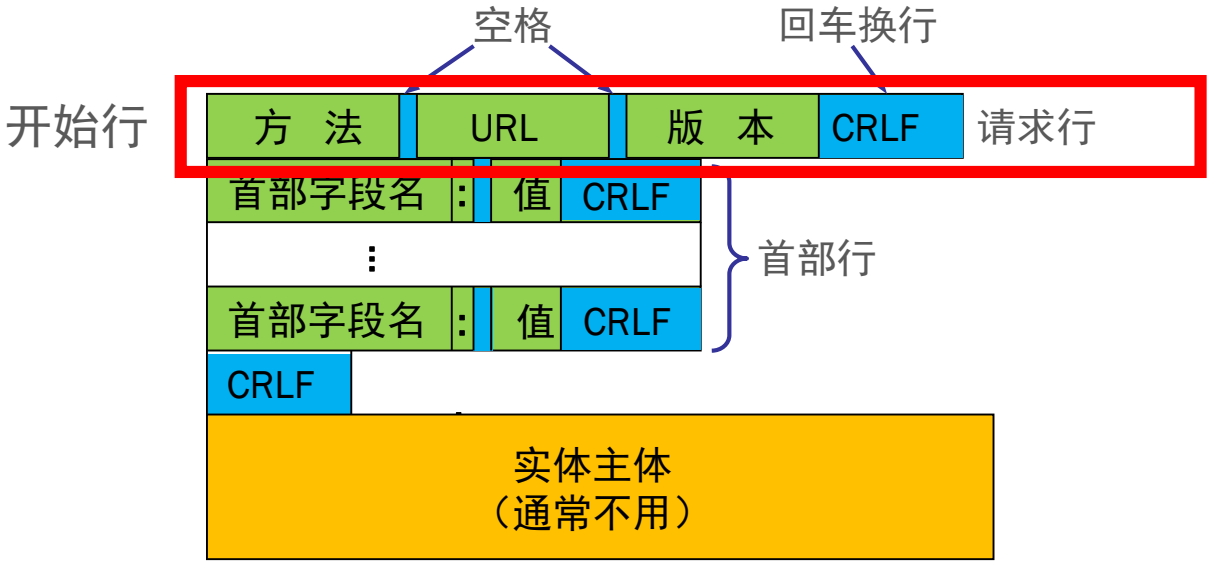
非流水线方式：客户在收到前一个响应后才能发出下一个请求。这比非持续连接的两倍 RTT 的开销节省了建立 TCP 连接所需的一个 RTT 时间。但服务器在发送完一个对象后，其 TCP 连接就处于空闲状态，浪费了服务器资源。

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。

3. HTTP 的报文结构

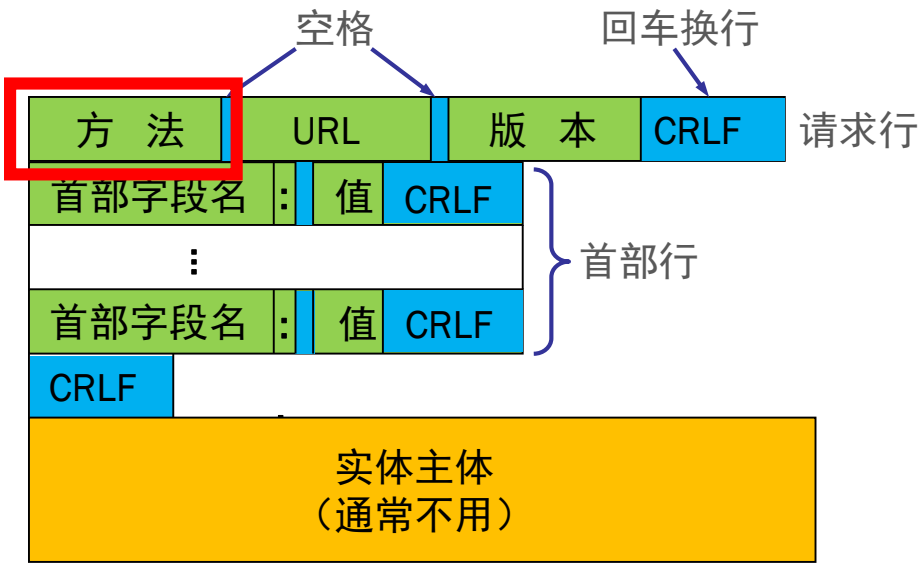
- HTTP 有两类报文：
  - 请求报文——从客户向服务器发送请求报文。
  - 响应报文——从服务器到客户的回答。
- 由于 HTTP 是面向正文的(text-oriented)，因此在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的。

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）

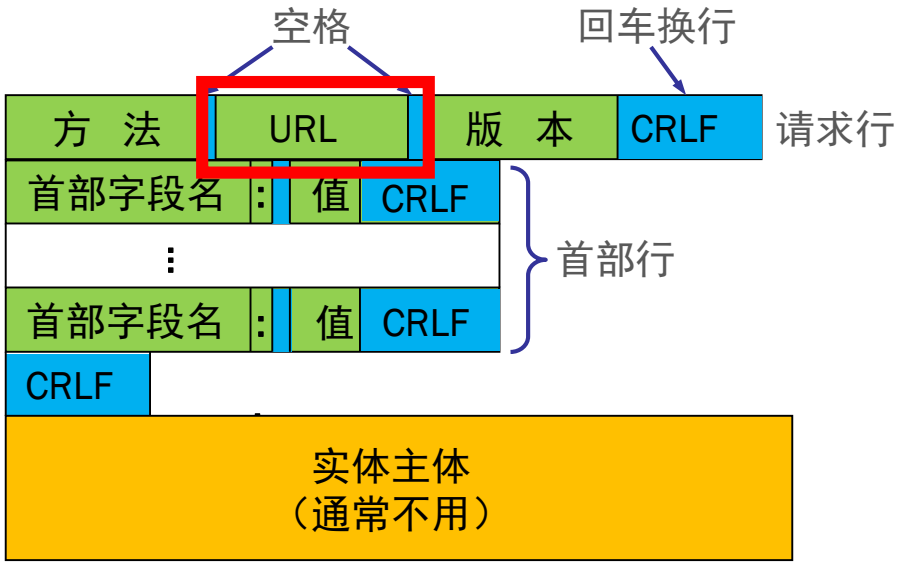


“方法”是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的。

HTTP 请求报文的一些方法

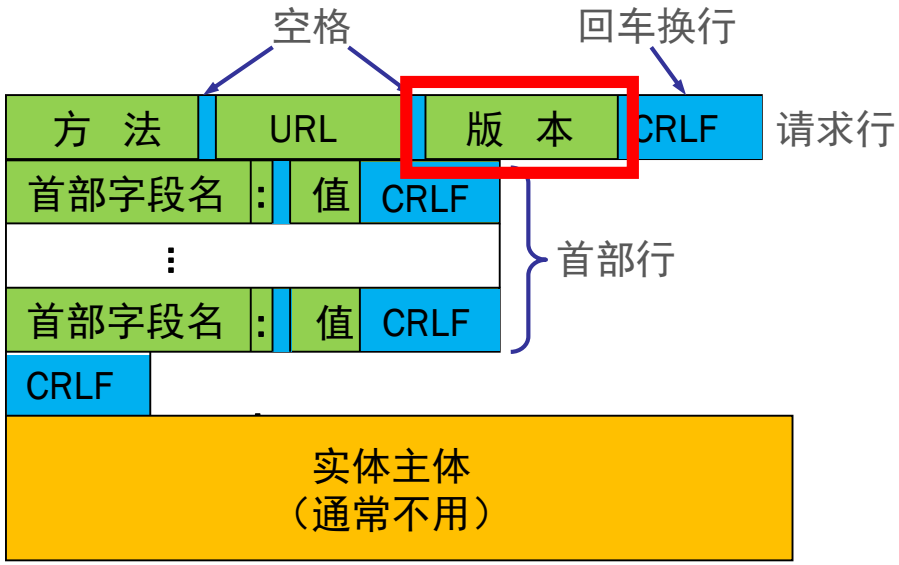
方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL所标志的信息
HEAD	请求读取由 URL所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL下存储一个文档
DELETE	删除指明的 URL所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

HTTP 的报文结构（请求报文）



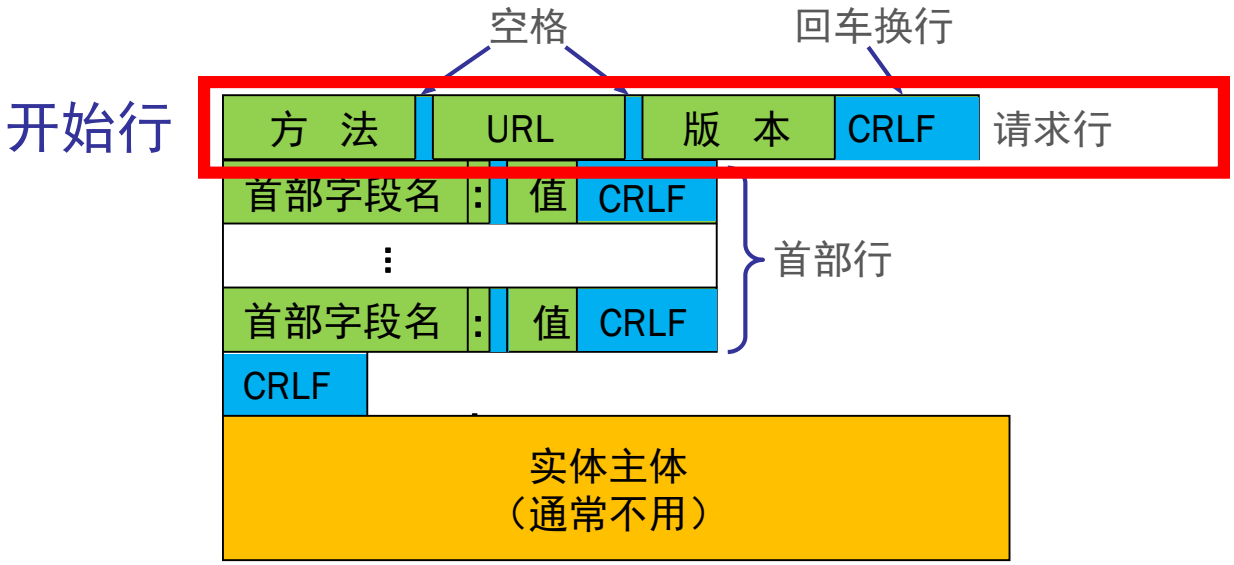
“URL”是所请求的资源的 URL。

HTTP 的报文结构（请求报文）



“版本”是 HTTP 的版本。

HTTP 的报文结构（响应报文）



响应报文的开始行是状态行。
状态行包括三项内容，即 HTTP 的版本，状态码，以及解释状态码的简单短语。

状态码都是三位数字

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

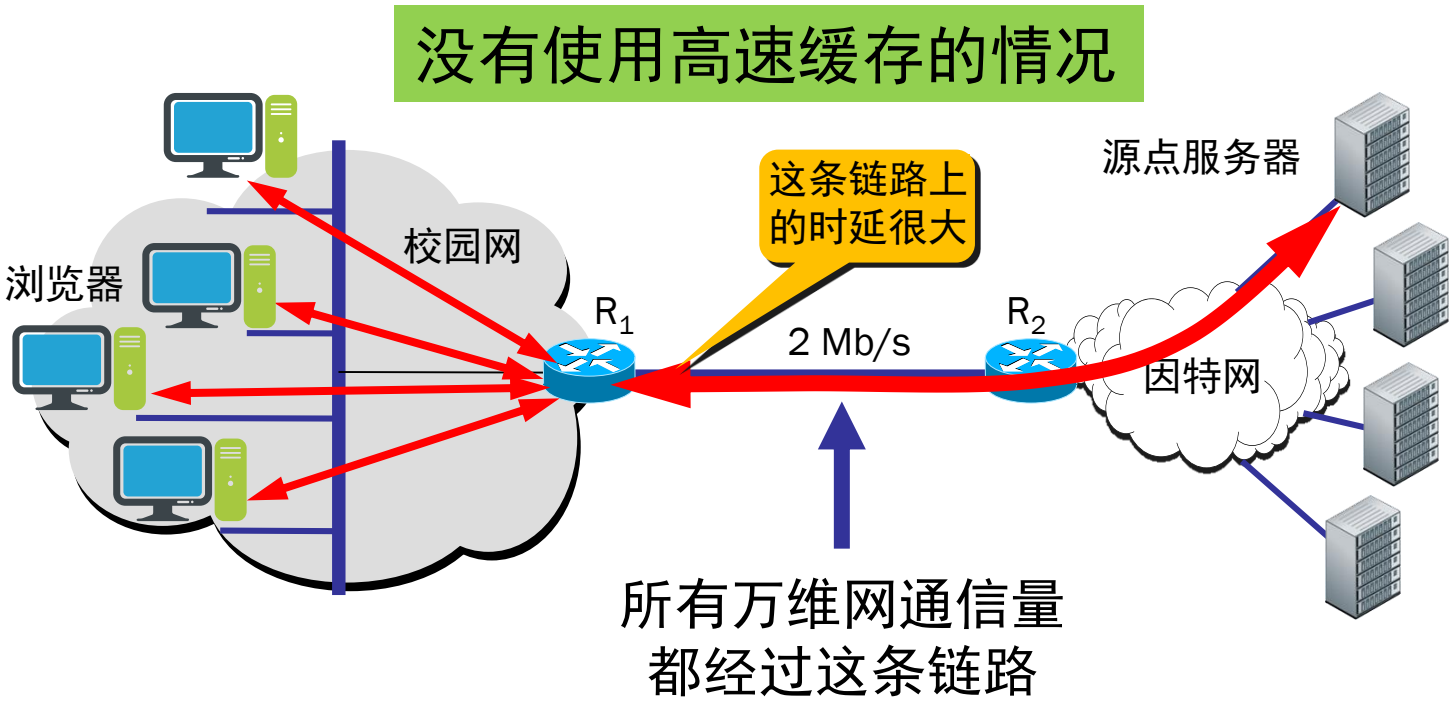
4. 在服务器上存放用户的信息

- 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。
- 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。

5. 代理服务器(proxy server)

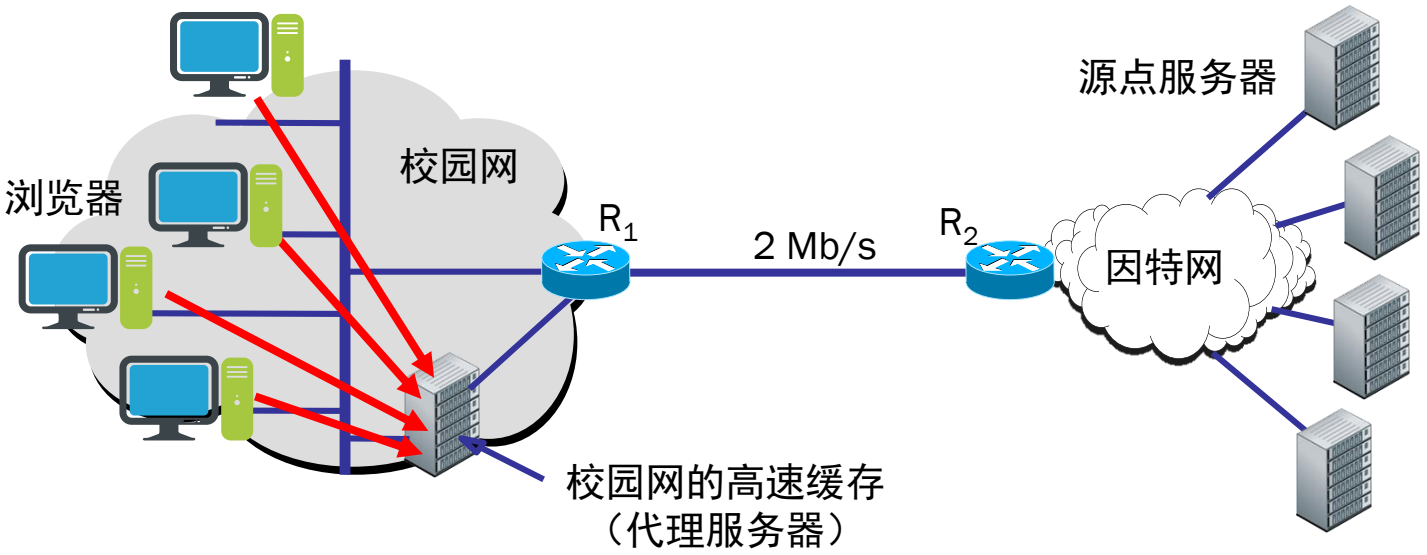
- **代理服务器**(proxy server)又称为万维网高速缓存(Web cache)，它代表浏览器发出 HTTP 请求。
- 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求相同的新请求到达时，万维网高速缓存就把暂存的响应发送出去，而不需要按 URL 的地址再去因特网访问该资源。

使用高速缓存可减少访问因特网服务器的时延



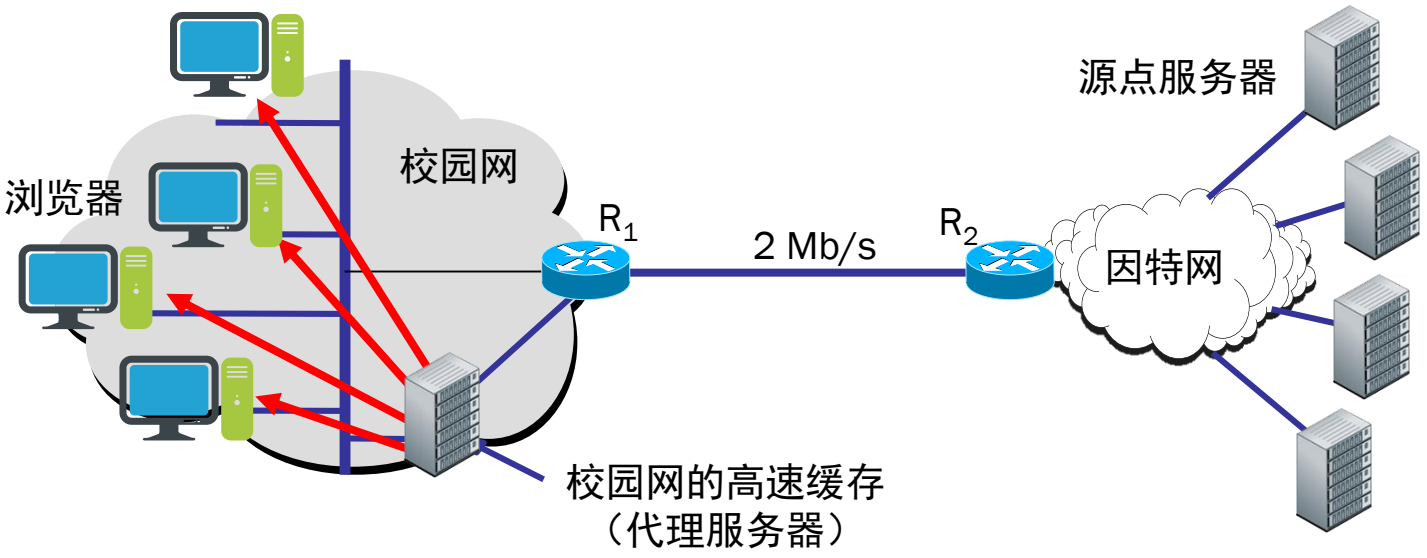
使用高速缓存的情况

- **(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**



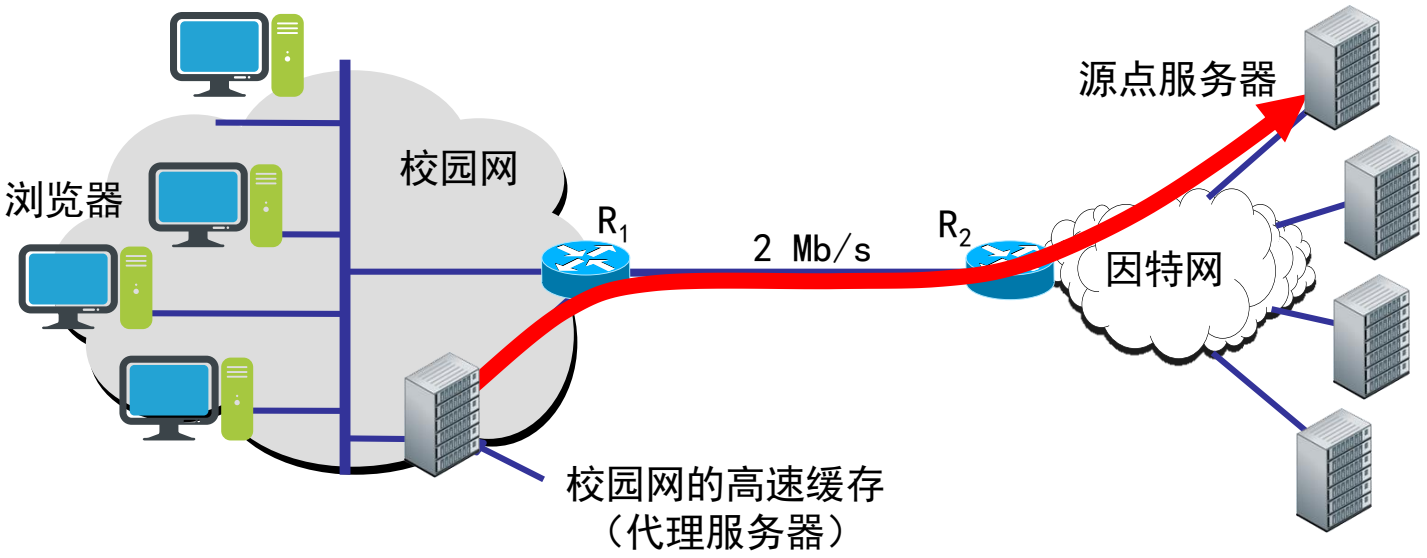
使用高速缓存的情况

- (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。



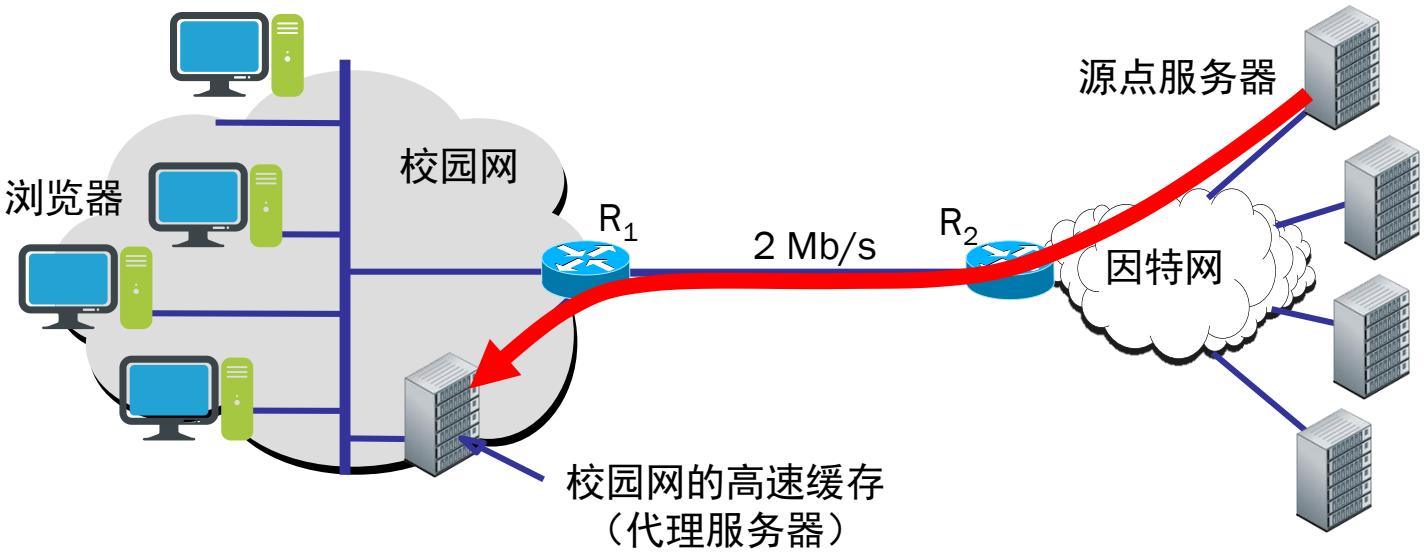
使用高速缓存的情况

- (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。



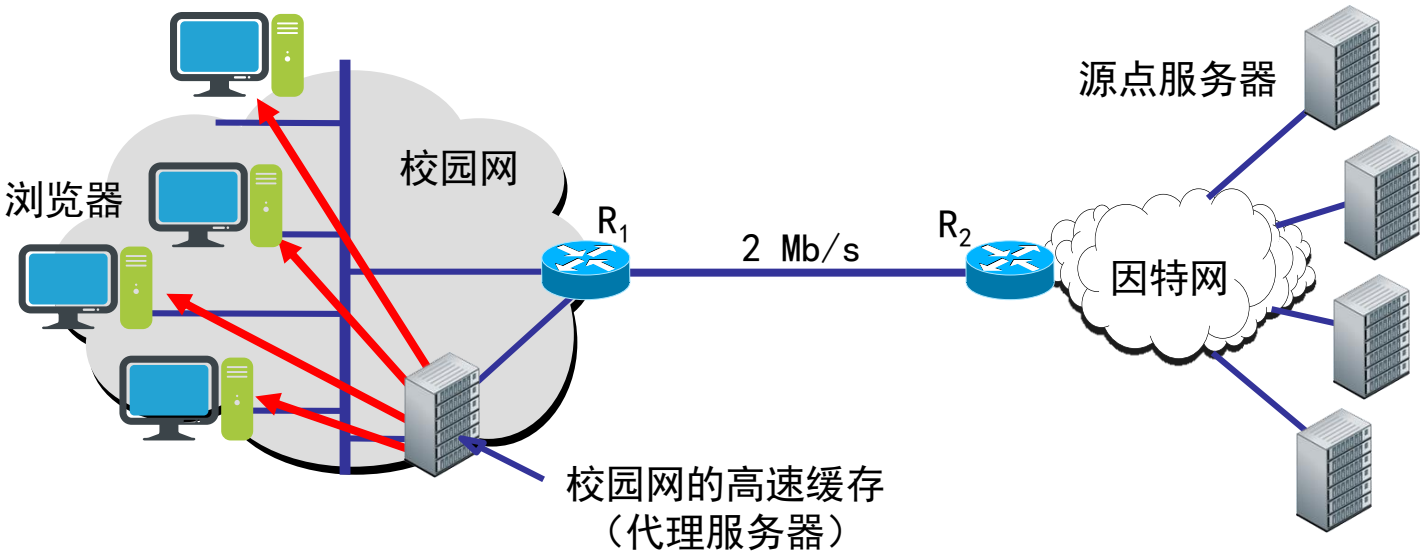
使用高速缓存的情况

- (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。



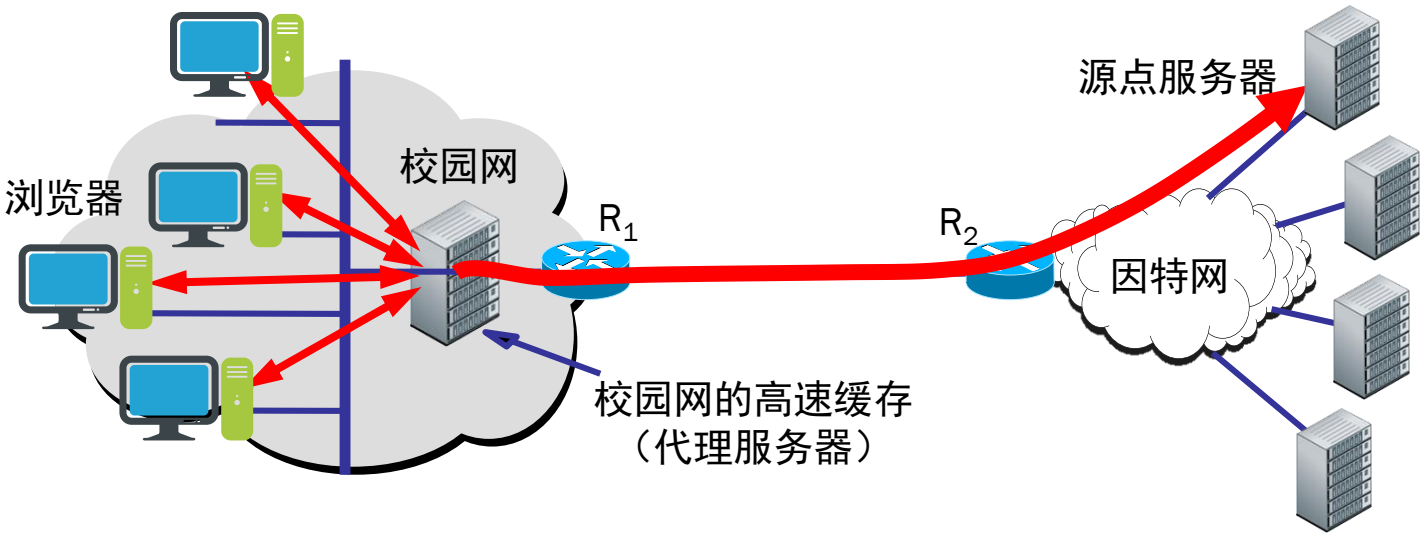
使用高速缓存的情况

- (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。



使用高速缓存的情况

- (6) 代理服务器的另一个作用就是可以用来隔离内外网络。



6.3.4 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。**

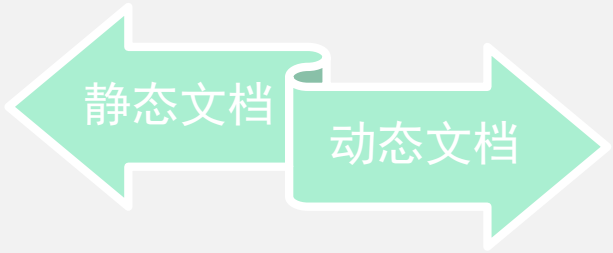
- HTML允许在万维网页面中插入图像。
- HTML还规定了链接的设置方法。
- 链接的终点可以是其他网站上的页面。这种链接方式叫做远程链接。
- 有时链接可以指向本计算机中的某一个文件或本文件中的某处。这叫做本地链接。

HTML 文档

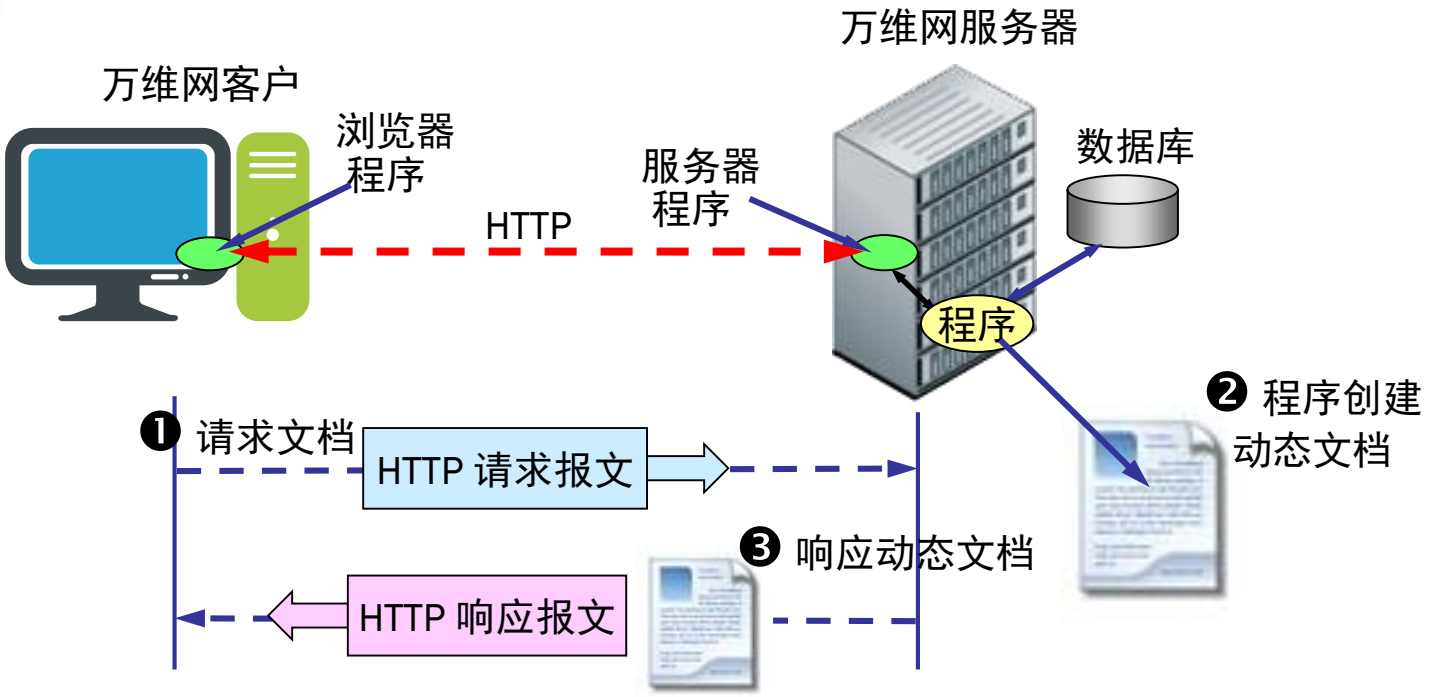
- 虽然完全可以用任何文本编辑器来编辑HTML文档，但使用“所见即所得”的万维网页面开发工具能很方便地制作各种美观的页面。
- 目前较为流行的网页制作工具有FrontPage, DreamWeaver等。

2. 动态文档

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



动态文档


万维网服务器
万维网客户
浏览器
程序
服务器
程序
数据库
HTTP
程序
❷ 程序创建
动态文档
❶ 请求文档
HTTP 请求报文
❸ 响应动态文档
HTTP 响应报文

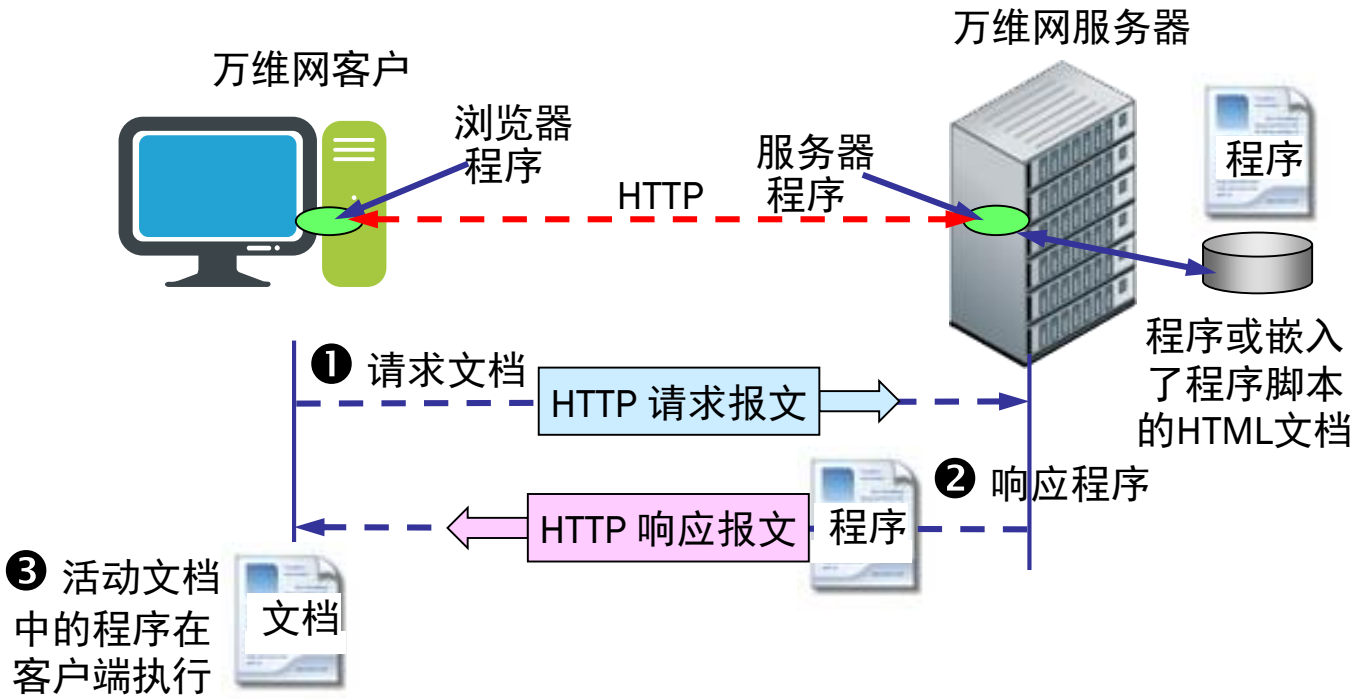
动态文档技术

- 通用网关接口(Common Gateway Interface, CGI)
- 超文本预处理器(Hypertext Preprocessor, PHP)，使用Perl语言
- Java服务器网页(Java Server Pages, JSP)，使用Java语言
- 活动服务器网页(Active Server Pages, ASP)，使用VBScript, JScript等语言
- ASP.NET，使用C#, VB.net等语言

3. 活动文档

- **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


万维网客户
浏览器
程序
HTTP
服务器
程序
万维网服务器
程序
程序或嵌入
了程序脚本
的HTML文档
❶ 请求文档
HTTP 请求报文
❷ 响应程序
HTTP 响应报文
程序
❸ 活动文档
中的程序在
客户端执行
文档

活动文档技术

Java applet 01

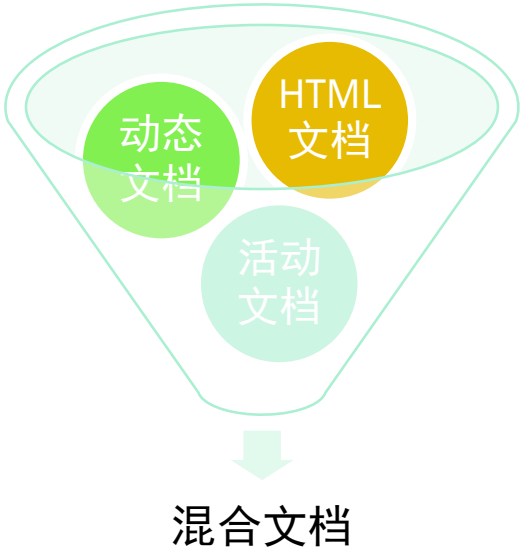
02 JavaScript

ActionScript 03

04 等等

混合文档

- 实际上，现在万维网上的很多文档都是这三种文档的混合体。
- 在这样的万维网页面中有一部分是用HTML编写的静态部分，一部分是用程序在服务器端动态生成的，还有一部分是可以在浏览器端运行的程序或程序脚本。



4. B/S应用程序结构

- 浏览器/服务器(Browser/Server)方式，一种特殊的C/S方式
- 利用动态和活动网页，通过通用的浏览器为用户提供人机交互的界面
- 优点是用户不需要安装单独的应用程序，简化了应用的开发、维护和使用
- 越来越多的网络应用采用B/S结构，例如购物网站、电子邮件、搜索引擎、博客等等。

6.3.5 移动Web

- **早期采用新的协议栈：无线应用协议WAP (Wireless Application Protocol)，但随着网络带宽和设备计算能力的提高，目前更多的方法是：**

1. 移动版本网页。现在越来越多的Web网站针对移动电话用户开发和设计移动友好(mobile-friendly)的Web页面内容。

2. 内容转换技术。利用设置在移动电话和Web服务器之间的转码服务器，将页面内容转换成移动友好的内容再发送给用户。

3. 移动浏览器。为手持设备的小型屏幕显示网页做了各种优化，通常与转码器技术配合使用，以减少产生的流量。例如全球著名的Opera，国内的UC浏览器。

6.3.6 万维网搜索引擎

- 搜索引擎实际上就是一个基于B/S结构的网络应用软件系统
- 从网络用户角度来看，它根据用户提交的类自然语言查询词或者短语，返回一系列很可能与该查询相关的网页信息，供用户进一步判断和选取。
- 搜索引擎要尽量提高**响应时间**、**查全率**、**查准率**和**用户满意度**四个指标，即用尽可能少的时间返回尽可能相关的网页信息列表，并将最可能满足用户需求的信息排在最前面。

（1）网页搜集

- 大规模搜索引擎都是事先通过网页搜集软件在万维网上自动搜集大量网页并下载存储在本地存储系统中供以后进行查询。
- 通过网页之间的超链关系，网页搜集软件按照先深或先广遍历算法在万维网上从一个网页查找到另一个网页，就好像蜘蛛在蜘蛛网上爬行一样，因此网页收集软件往往被称为“蜘蛛”或“网络爬虫”。

（2）建立索引

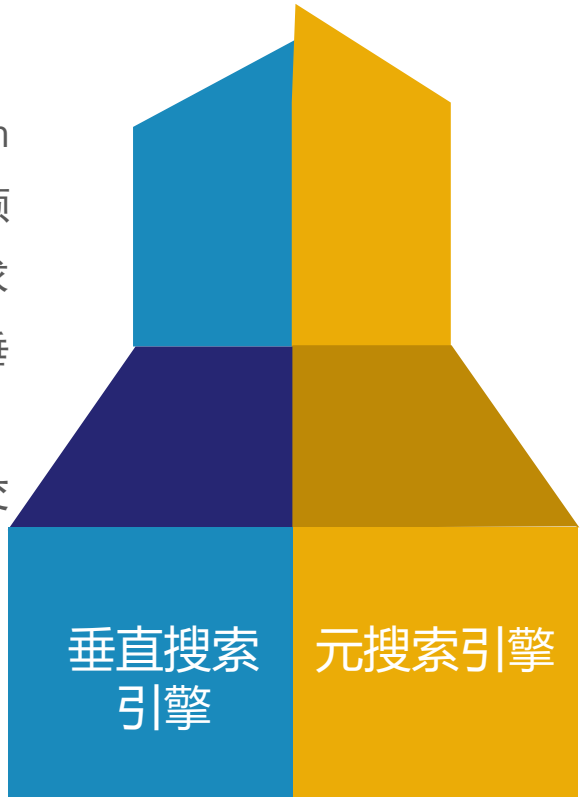
- 针对每个查询请求直接到这些海量网页中去全文检索太慢。
- 更好的方法是事先遍历每个网页并记录每个网页包含的各种词汇及其位置，然后根据词汇反过来建立索引项记录包含该词汇的网页及位置。以后根据关键词检索网页就非常迅速了。
- 将文档以关键词作为索引的数据结构被称为**倒排表**，是进行快速全文检索的关键。

（3）检索排序

- 搜索引擎需要将与用户查询条件相关性最高的网页条目放在最前面。
- 最简单的方法就是根据查询关键词在网页上出现的频率进行排序。
- Google综合考虑了网页的重要性和相关性，认为被链接得越多的网页越重要，被越重要的网页链接的网页也越重要，其PangeRank算法可以快速地计算每个网页的重要性排名。Google因此成为了最优秀的搜索引擎。

2. 垂直搜索引擎和元搜索引擎

垂直搜索引擎(Vertical Search Engine)，它们针对某一特定领域、特定人群或某一特定需求提供搜索服务。目前热门的垂直搜索领域有：购物、旅游、汽车、论坛、房产、求职、交友、图片等等。



元搜索引擎(Meta Search Engine)是搜索引擎之上的搜索引擎，它们自己并不在万维网上搜集网页，而是在接受用户查询请求时，同时在其他多个搜索引擎上进行搜索，并将检索的结果进行综合处理后，以统一的格式返回给用户。

6.3.7 博客与微博

1. 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。**
- **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，网民不仅的因特网上内容的消费者，而且还是因特网上内容的生产者。**
- **现在从一些著名的门户网站的主页上都可以很容易地进入到博客的页面，这让用户查看或发表自己的博客都是非常方便的。**

博客与个人网站的区别

- 建立个人网站不仅的成本较高，需要租用个人空间、域名等，同时对建立网站的个人需要懂得HTML语言和网页制作等相关技术
- 博客在这方面是不需要什么投资的，所需的技术仅仅是会上网和会用键盘或书写板输入汉字即可。
- 因此网民用较短的时间就能够把自己写的博客发表在网上，而不像制作个人网站那样花费较多的时间。

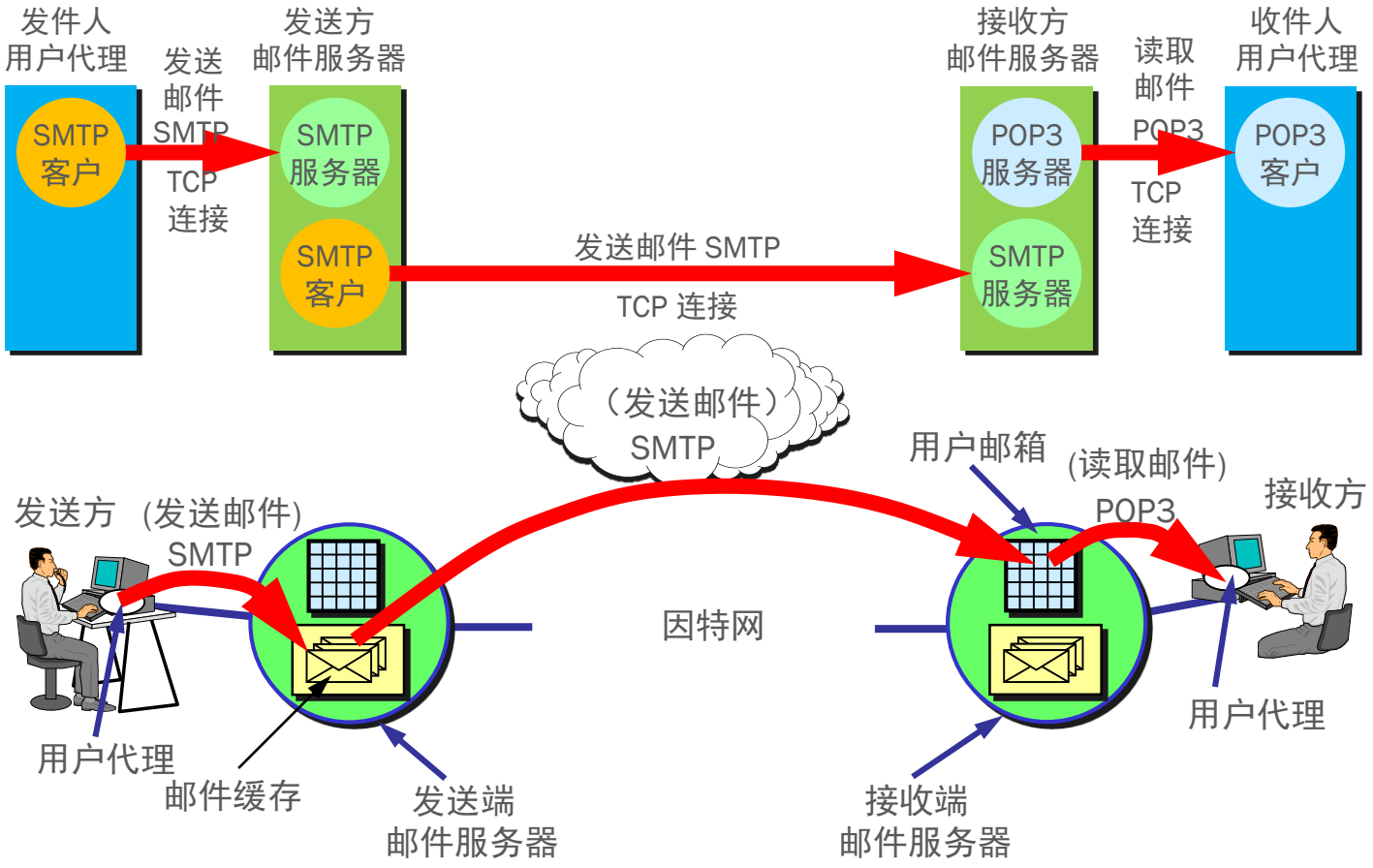
2. 微博

- 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。
- 根据新浪微博白皮书，从2010年3月到2010年6月，新浪微博月覆盖人数从2510.9万增长到4435.8万。
- 博客或微博里的朋友，常称为“博友”。微博也被人戏称为“围脖”，因此现在也有人把博友戏称为“脖友”。

6.4.1 电子邮件系统的组成

- **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。

发送和接收电子邮件的几个重要步骤

01 OPTION 发件人调用 PC 机中的用户代理撰写和编辑要发送的邮件。

02 OPTION 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，

03 OPTION SMTP 服务器把邮件临时存放在邮件缓存队列中，等待发送。

04 OPTION 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。

05 OPTION 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。

06 OPTION 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　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

6.4.2 简单邮件传送协议 SMTP

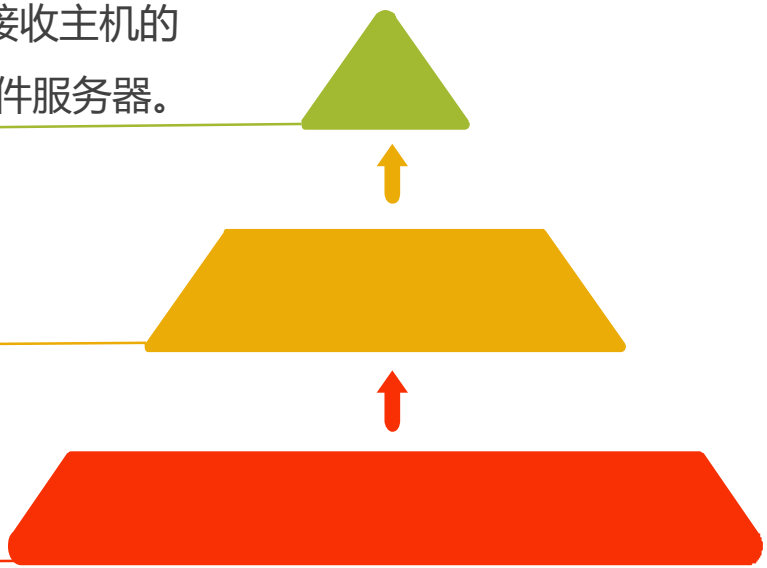
- SMTP 所规定的就是在两个相互通信的 SMTP 进程之间应如何交换信息。
- 由于 SMTP 使用客户服务器方式，因此负责发送邮件的 SMTP 进程就是 SMTP 客户，而负责接收邮件的 SMTP 进程就是 SMTP 服务器。
- SMTP 规定了 14 条命令和 21 种应答信息。每条命令用 4 个字母组成，而每一种应答信息一般只有一行信息，由一个 3 位数字的代码开始，后面附上（也可不附上）很简单的文字说明。

SMTP 通信的三个阶段

1. 连接建立：连接是在发送主机的 SMTP 客户和接收主机的 SMTP 服务器之间建立的。SMTP不使用中间的邮件服务器。

2. 邮件传送

3. 连接释放：邮件发送完毕后，SMTP 应释放 TCP 连接。



SMTP交互实例

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

6.4.3 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。

6.4.4 邮件读取协议POP3 和 IMAP

- SMTP用于发送邮件，是“推“协议
  - 客户端向服务器端推送邮件
- 而邮件读取协议，是“拉”协议
  - 客户端向服务器端拉取邮件
  - POP3 (Post Office Protocol)
  - IMAP (Internet Message Access Protocol)

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。
- POP3有两种工作方式：**下载并删除方式**和**下载并保留方式**。

IMAP 协议 (Internet Message Access Protocol)

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

IMAP 的特点

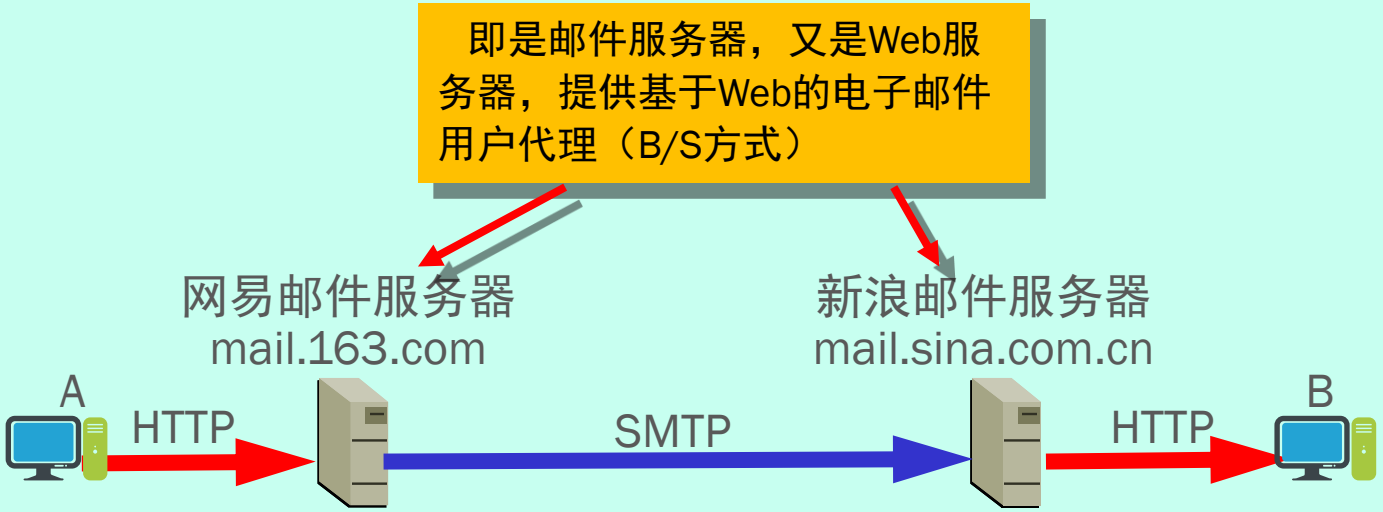
- IMAP最大的好处就是用户可以在不同的地方使用不同的计算机随时上网阅读和处理自己的邮件。
- IMAP 还允许收件人只读取邮件中的某一个部分。例如，收到了一个带有视像附件（此文件可能很大）的邮件。为了节省时间，可以先下载邮件的正文部分，待以后有时间再读取或下载这个很长的附件。
- IMAP 的缺点是如果用户没有将邮件复制到自己的 PC 机上，则邮件一直是存放在 IMAP 服务器上。因此用户需要经常与 IMAP 服务器建立连接。

必须注意

- 不要将邮件读取协议 POP 或 IMAP 与邮件传送协议 SMTP 弄混。
- 发信人的用户代理向源邮件服务器发送邮件，以及源邮件服务器向目的邮件服务器发送邮件，都是使用 SMTP 协议。
- 而 POP 协议或 IMAP 协议则是用户从目的邮件服务器上读取邮件所使用的协议。

6.4.5 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。



6.4.6 通用因特网邮件扩充 MIME

1. 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。

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[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 和 SMTP 的关系

MIME 主要包括三个部分

- 5 个新的邮件首部字段，它们可包含在[RFC 822]首部中。这些字段提供了有关邮件主体的信息。
- 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 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。

2. 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。

3. 内容类型

- MIME着标准规定 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。

6.5 文件传送协议FTP

- **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。

6.5 文件传送协议FTP

- 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

01 计算机存储数据的格式不同。

02 文件的目录结构和文件命名的规定不同。

03 对于相同的文件存取功能，操作系统使用的命令不同。

04 访问控制方法不同。

6.5 文件传送协议FTP

- 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。

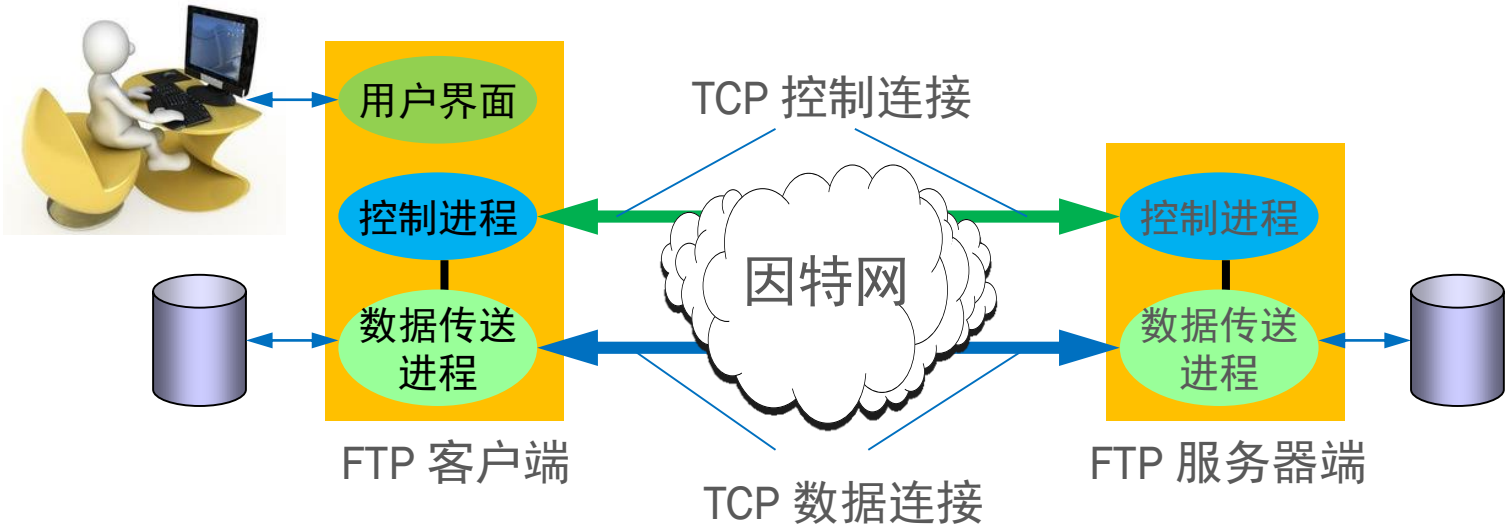
主进程的工作步骤如下

- 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。从属程对客户进程的请求处理完毕后即终止，但从属程在运行期间根据需要还可能创建其他一些子进。
- 回到等待状态，继续接受其他客户进程发来的请。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 使用的两个 TCP 连接



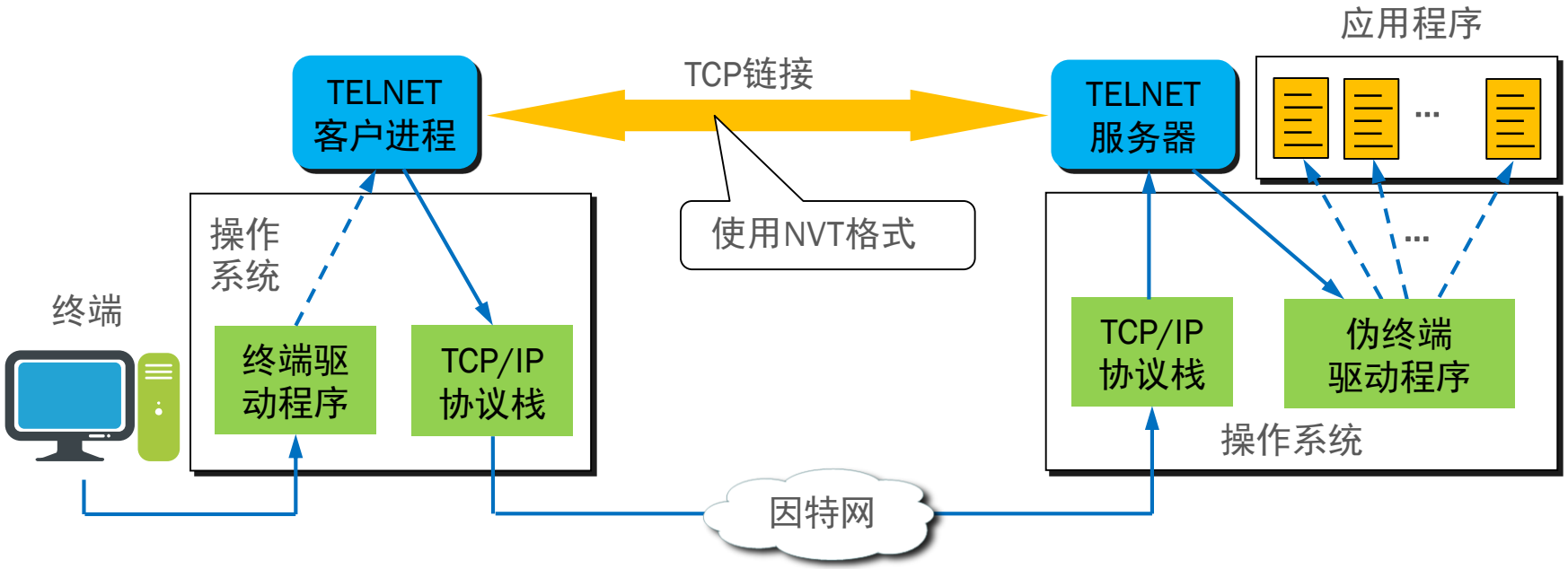
FTP是有状态的

- FTP服务器必须在整个会话期间保留用户的状态信息。特别是，服务器必须把特定的用户账户与控制连接联系起来，服务器必须追踪用户在远程文件目录树上的当前位置。

6.6 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。
- TELNET以前应用得很多，但现在使用它的主要是网络工程技术人员而不是普通用户。
- 专业人士使用它登录到远程设备，如服务器、路由器、交换机等，进行一些调试、管理和配置工作。
- TELNET也使用客户/服务器方式。在本地系统运行TELNET客户进程，而在远程主机则运行TELNET服务器进程。

通过网络登录到远程主机



网络虚拟终端 NVT 格式

- 客户软件把用户的击键和命令转换成 NVT 格式，并送交服务器。
- 服务器软件把收到的数据和命令，从 NVT 格式转换成远地系统所需的格式。
- 向用户返回数据时，服务器把远地系统的格式转换为 NVT 格式，本地客户再从 NVT 格式转换到本地系统所需的格式。

6.7 动态主机配置协议 DHCP

- 一台主机要连接到网上需要配置的项目
  - 01 IP 地址
  - 02 子网掩码
  - 03 默认路由器的 IP 地址
  - 04 域名服务器的 IP 地址
- 这些信息通常存储在一个配置文件中，计算机在引导过程中可以对这个文件进行存取。
- DHCP允许一台计算机加入新的网络和获取IP地址而不用手工参与。

动态主机配置协议 DHCP (Dynamic Host Configuration Protocol)

- **动态主机配置协议 DHCP 提供了即插即用连网(plug-and-play networking)的机制。**
- **这种机制允许一台计算机加入新的网络和获取IP地址而不用手工参与。**

DHCP 使用客户服务器方式。

- 由于不知道 DHCP 服务器的地址，需要 IP 地址的主机在启动时用UDP广播发送一个DHCP**发现报文**（DHCPDISCOVER），这时该主机就成为 DHCP 客户。
- 本地网络上所有主机都能收到此广播报文，但只有 DHCP 服务器才回答此广播报文。由于此时DHCP客户还没有分配到IP地址，DHCP服务器通过UPD广播向DHCP客户应答一个DHCP**提供报文**（DHCPOFFER），包含可以“提供”的IP地址等配置信息。
- DHCP 服务器先在其数据库中查找该计算机的配置信息。若找到，则返回找到的信息。若找不到，则从服务器的 IP 地址池(address pool)中取一个地址分配给该计算机。

动态主机配置协议 DHCP (Dynamic Host Configuration Protocol)

- **动态主机配置协议 DHCP 提供了即插即用连网(plug-and-play networking)的机制。**
- **这种机制允许一台计算机加入新的网络和获取IP地址而不用手工参与。**

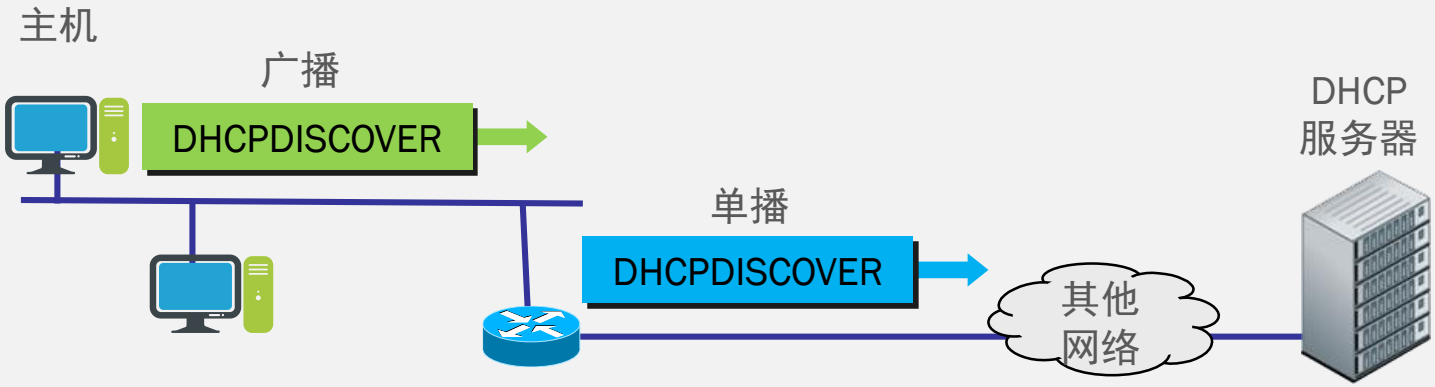
DHCP 使用客户服务器方式。

- DHCP客户可能会收到来自多个服务器的提供报文，需要选择其中的一个，并广播一个DHCP **请求报文**（DHCPREQUEST）来正式请求该提供报文中提供的配置信息。
- 提供该配置信息的服务器会对该请求报文用DHCP**确认报文**（DHCPACK）进行确认，而其他DHCP服务器收到该请求报文后会释放预分配的资源。
- 由于DHCP客户收到确认报文后才能使用提供报文中的配置信息，该确认报文也需要使用广播。

DHCP 中继代理(relay agent)

- 并不是每个网络上都有 DHCP 服务器，这样会使 DHCP 服务器的数量太多。现在是每一个网络至少有一个 DHCP **中继代理**，它配置了 DHCP 服务器的 IP 地址信息。
- 当 DHCP 中继代理收到主机发送的发现报文后，就以单播方式向 DHCP 服务器转发此报文，并等待其回答。收到 DHCP 服务器回答的提供报文后，DHCP 中继代理再将此提供报文发回给主机。

DHCP 使用客户服务器方式。



注意：DHCP 报文只是 UDP 用户数据报中的数据。

租用期(lease period)

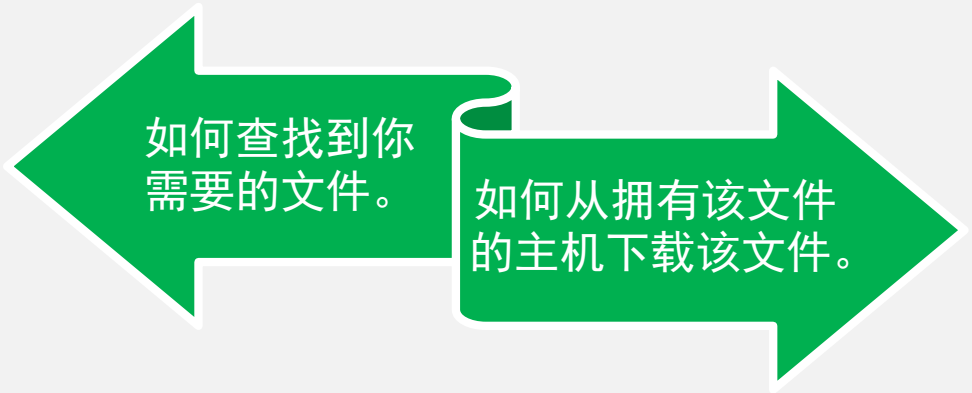
- DHCP 服务器分配给 DHCP 客户的 IP 地址的临时的，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。DHCP 协议称这段时间为**租用期**。
- 租用期的数值应由 DHCP 服务器自己决定。
- DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。

6.8 P2P文件共享

- 基于客户/服务器体系结构的应用要求有总是在运行着的基础设施服务器，例如：DNS服务器、万维网服务器、邮件服务器等等。
- 与这些应用不同，基于P2P体系结构的应用是对等方之间直接进行通信，而且对等方主要运行于间断连接的主机上，如个人电脑上。
- 目前在因特网上流行的P2P应用主要包括P2P文件共享、即时通信、P2P流媒体、分布式存储等。

文件共享的两个基本问题

- **对于文件共享应用实际上有两个基本的问题要解决：**



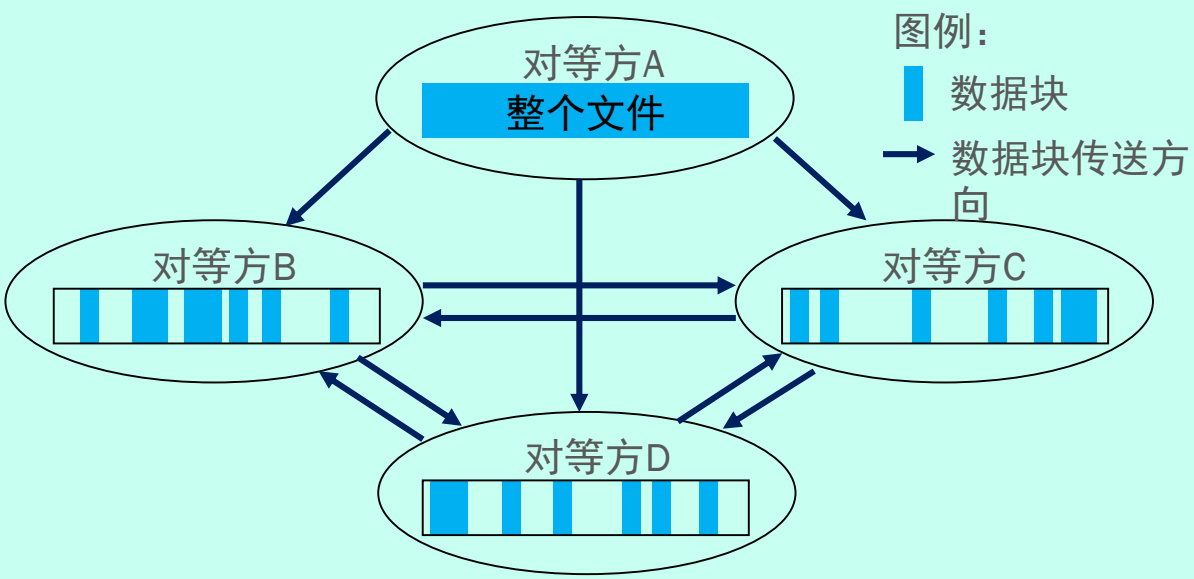
6.8.1 P2P文件分发

P2P文件分发例子

- 将主机H1中的一个大小为f的大文件分发给其余7台主机，假设文件传输的瓶颈是各主机的上行速率b。
- 对于客户/服务器方式，主机H1为服务器，而其它主机为客户，显然主机H1要依次将文件发送给所有其它主机，需要7f/b时间。可以证明采用客户/服务器方式，文件分发时间随客户数量呈线性增长。
- 采用P2P方式，每个对等方都能在收到文件后再将该文件分发给其余对等方，从而协助主机H1进行分发，这样大大缩短了文件分发的时间。
- 例如可以在3f/b时间内将文件分发给所有7台主机：
  - 第1个f/b时间，H1发给H2；
  - 第2个f/b时间，H1发给H3，H2发给H4；
  - 第3个f/b时间，H1发给H5，H2发给H6，H3发给H7，H4发给H8。

对等方互相交换文件数据块

通过分片，即将文件划分为很多等长的小数据块进行分发，可以进一步加快文件分发的速度。



6.8.2 在P2P对等方中搜索对象

- **如何找到你所感兴趣的对象，这里的对象可以是：**

01 文件共享系统中的文件或文件的索引

02 即时讯息系统中的某个好友

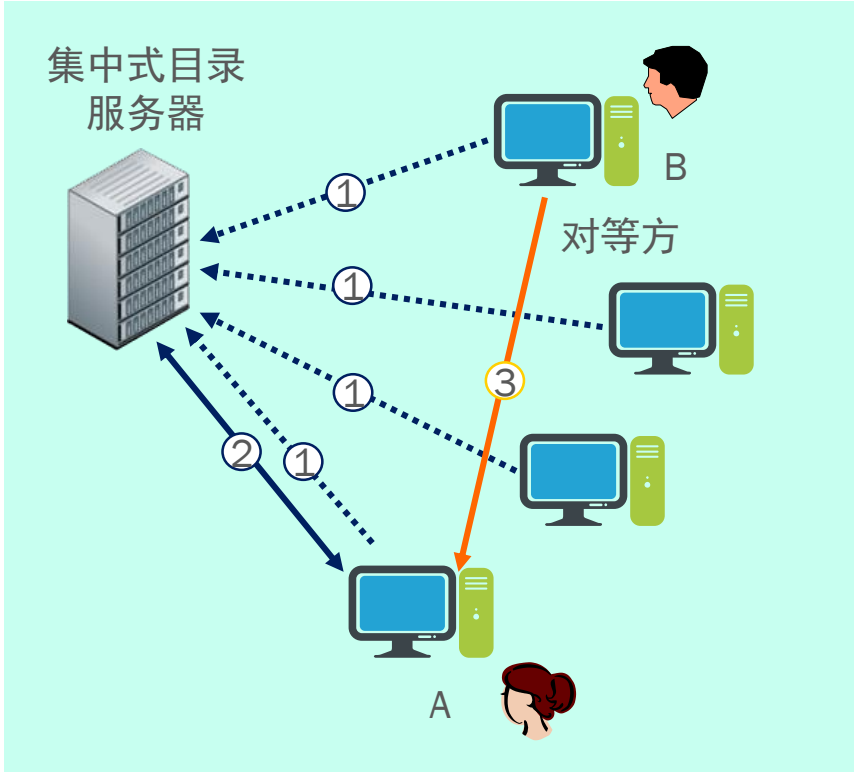
03 或者某个特殊资源

04 等等…

1. 集中式目录

Napster:

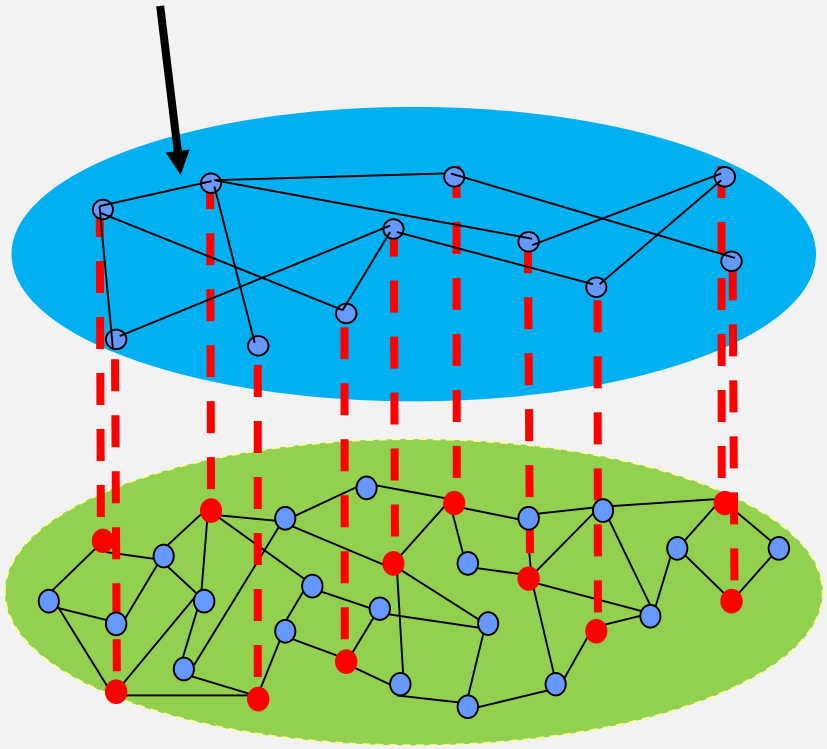
- **1) 当对等方启动时或内容发生更新，通知中心目录服务器:**
  - **IP地址**
  - **可共享的对象名称**
- **2) A向中心目录服务器查询歌曲M**
- **3) A向B请求歌曲M**



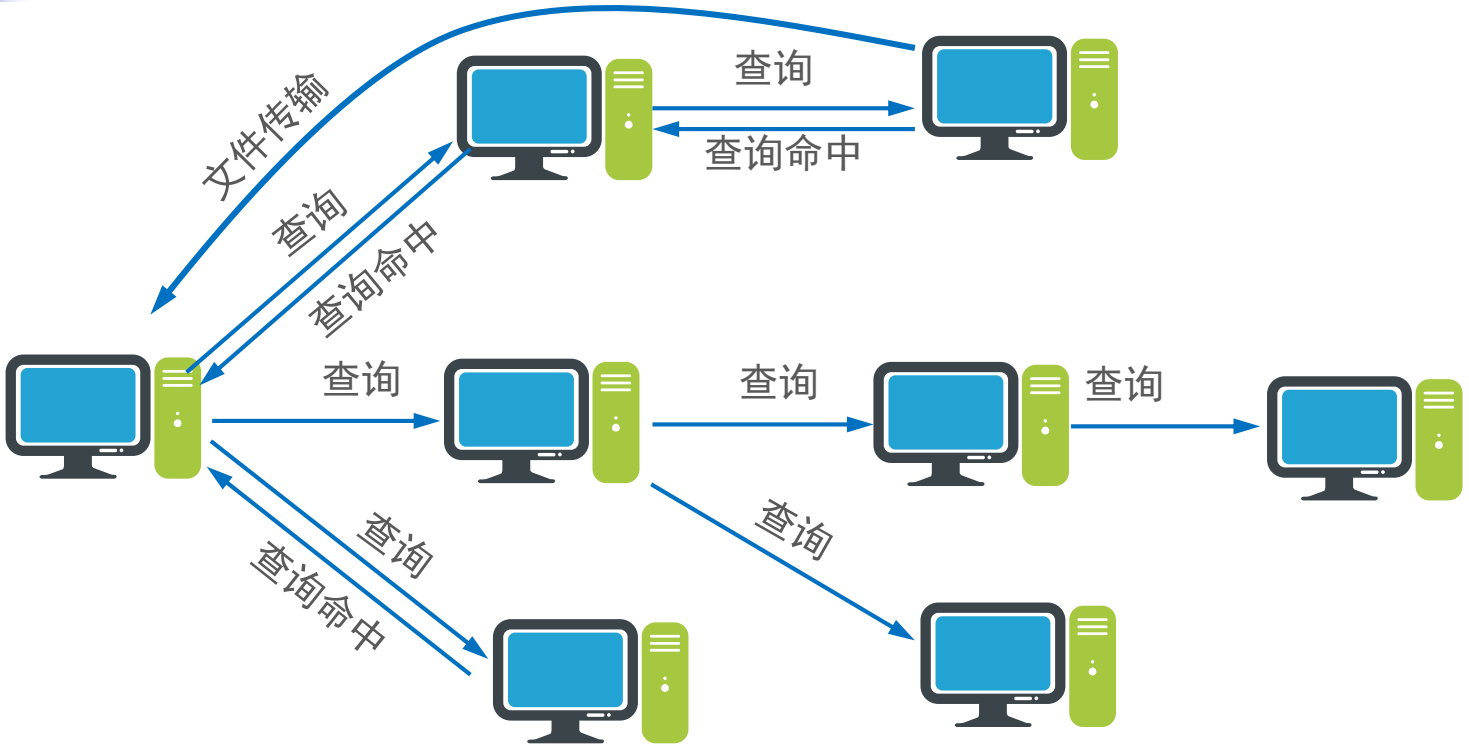
2. 查询洪泛

- 使用集中式目录定位内容虽然非常简单，但存在客户/服务器方式所固有的缺点，即服务器成为整个系统的性能瓶颈和故障点。
- 一些P2P文件共享软件，例如Gnutella，没有使用集中式服务器来定位文件，而是在应用层把所有对等方组织的**覆盖网络**(overlay network)上通过洪泛法进行查询。

覆盖网络对等方组成的逻辑网络



查询洪泛过程



范围受限的查询洪泛

- 在一个大的覆盖网络上进行查询洪泛，会在网络中产生大量的流量。为解决该问题，可使用范围受限的查询洪泛。
- 当对等方发送初始查询报文时，在报文的对等方计数字段中设置一个特定值（例如7）。每个对等方在转发查询报文时先把该字段减1，当对等方收到对等方计数字段降为0的查询报文时，就停止转发该查询。
- 由于不能搜索所有对等方，可能你所需要的文件存在于覆盖网络中，却不一定能找到它。

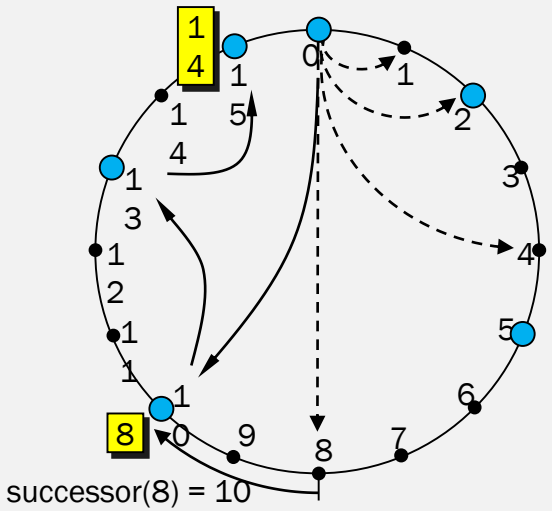
3. 分布式散列表

- 因为结点之间的边是随机选择的，Gnutella网络是一种**非结构化覆盖网络**，可扩展性差。
- 利用**分布式散列表DHT**（distributed hash table)技术可以把要定位的对象可靠地映射到网络中的特定结点（为该对象提供服务的结点），并且能有效路由到该结点。
- 这种覆盖网络要求相邻结点之间有某种数学关系，因此被称为**结构化覆盖网络**。

3. 分布式散列表

- 使用散列函数可以很容易地将名字映射到地址：hash(x)→n。
- 结构化覆盖网络将对象和结点一起均匀地散列到一个大的ID空间中（例如一个128位的ID空间）：
  - hash(object_name) → objid
  - hash(IP_addr) → nodeid

经典DHT算法：Chord



结点0索引表

1	2
2	2
4	5
8	10

结点10索引表

11	13
12	13
14	15
2	2

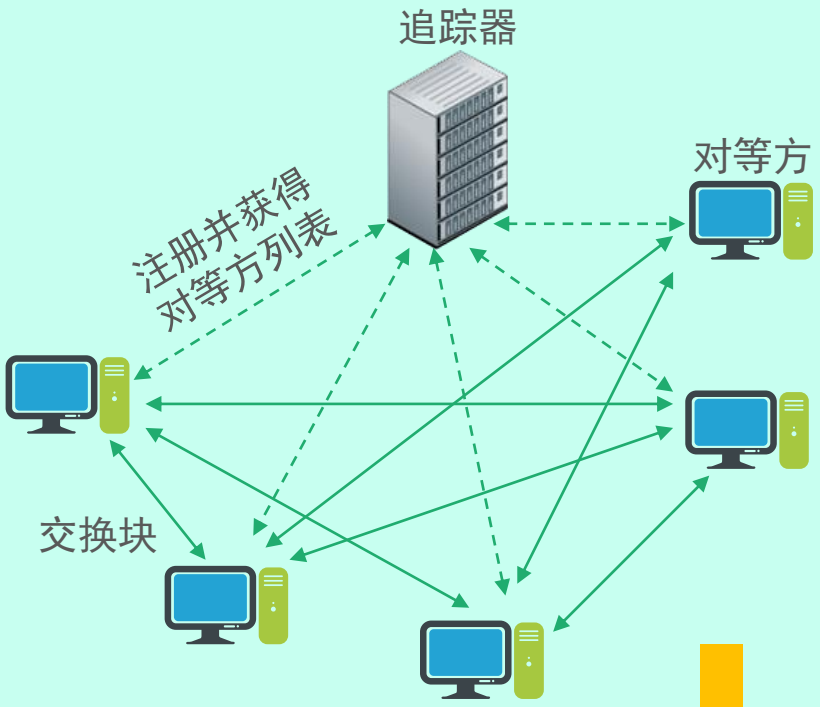
结点13索引表

14	15
15	15
1	2
5	5

图例：
结点
8 对象8

案例： BitTorrent

- “.torrent”种子文件中包含追踪器(tracket)（因特网上有很多追踪器）的地址
- 追踪器负责维护参与一个特定文件分发的所有对等方的信息。



6.9 多媒体网络应用

- 多媒体网络应用往往数据量巨大，要求更高的网络带宽，并且与传统的弹性应用（如电子邮件、文件传输、网页浏览等）不同的是，对端到端时延和时延抖动高度敏感，但却可容忍少量的数据丢失。

- 三类多媒体应用：
  - 流式存储音频/视频
  - 流式实况音频/视频
  - 实时交互音频/视频

6.9.1 实时多媒体数据传输中的问题

1. 音/视频压缩

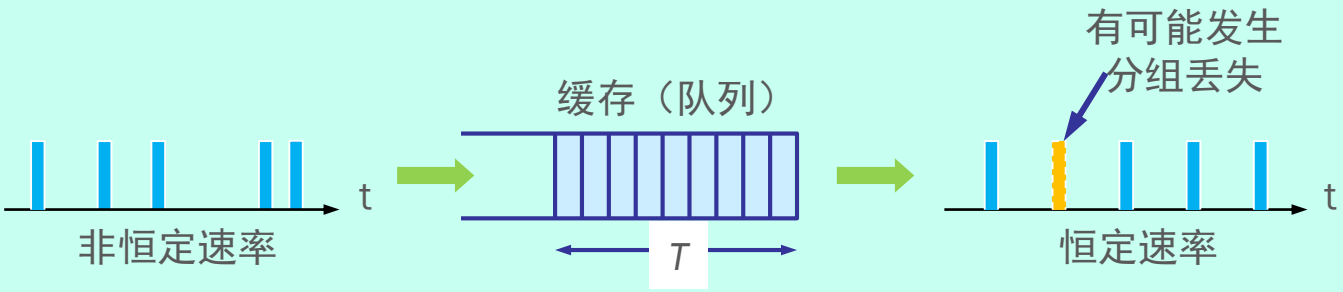
- 含有音频或视频的多媒体信息往往信息量巨大，会消耗大量的存储空间和网络带宽，导致很大的传输时延。因此在网上传送多媒体信息都无例外地采用各种信息压缩技术。
  - 话音压缩技术
  - 立体声音乐的压缩技术
  - 视频压缩技术

2. 时延抖动消除

- 实时音频/视频源以恒定速率产生并发送分组，因而这些分组是等时(isochronous)的。
- 但由于端到端的时延抖动，通过因特网到达接收方的分组则是非等时的

2. 时延抖动消除

- 接收端需设置适当大小的缓存。当缓存中的分组数达到一定的数量后再以恒定速率按顺序把分组读出进行还原播放（需要打上**时间戳**）。
- 缓存实际上就是一个先进先出的队列。图中标明的 *T* 叫做**播放时延**。



3. 丢失分组恢复

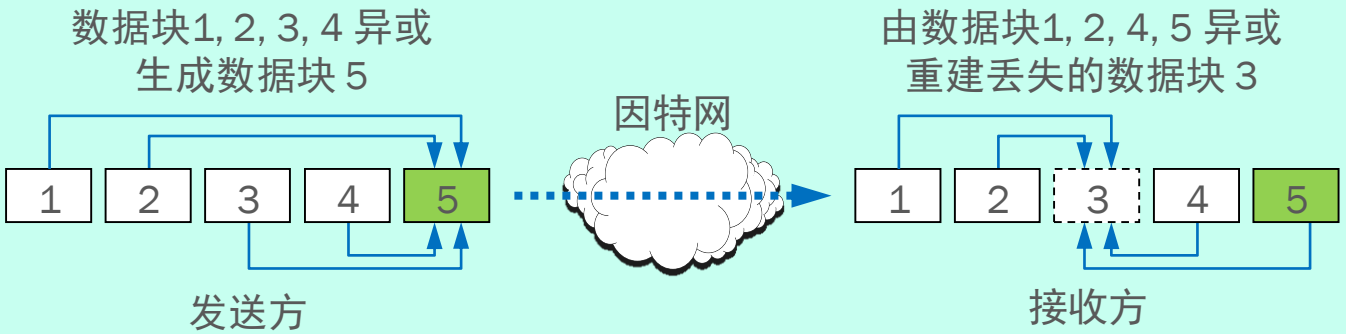
- **数据丢失会直接影响多媒体的播放质量。虽然TCP可以有效解决分组丢失问题，但会导致比UDP大很多的时延抖动。事实上，重传一个已经错过播放时间的分组是毫无意义的。**

- 在实时多媒体应用中倾向于使用前向纠错或数据恢复等技术来重建丢失的分组，或采用交织技术来减少分组丢失对对媒体流质量的影响。

前向纠错

前向纠错的基本思想是在原始分组流中添加冗余信息。对于少量的丢失分组，能够用这些冗余信息重建丢失数据。

一个例子：



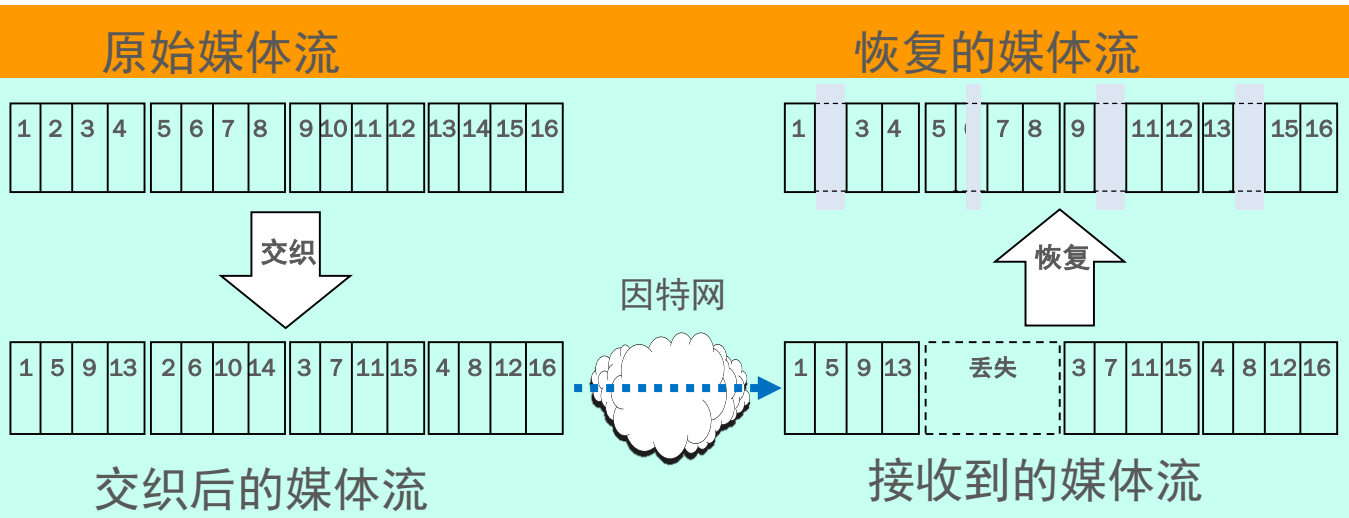
接收方数据恢复

- **利用音视频流的短期自相似特性，当少量数据丢失时，可以用相邻数据来估算丢失数据的近似值，从而减少丢失数据对音视频播放质量的影响。**
- **最简单的方法：用丢失分组的前一个分组来代替丢失分组。**

- 效果更好但计算量更大的方法：使用内插法，根据丢失分组的前后数据来估计它们之间的数据。

交织

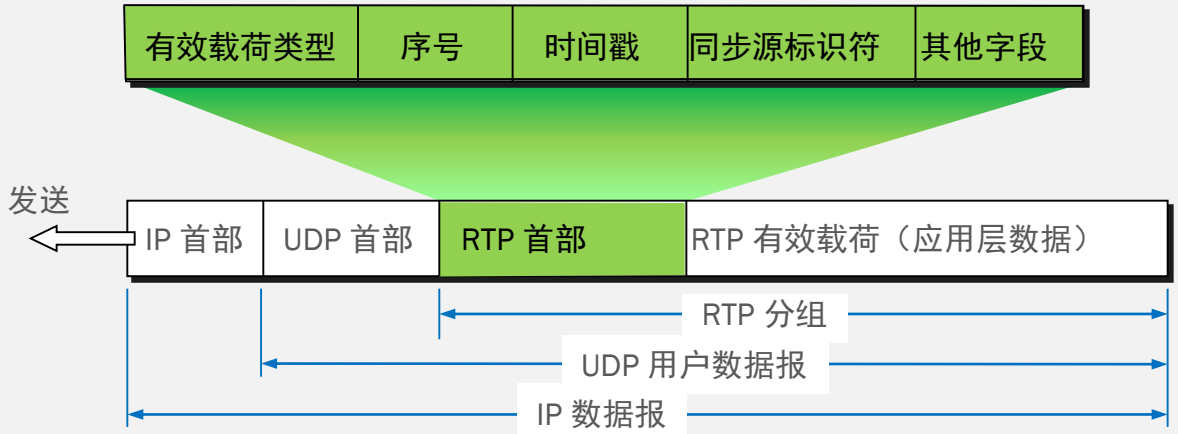
- 一个大间隙的数据丢失对音频/视频流质量影响较大，而多个小间隙的数据丢失对音频/视频流质量影响较小。
- 交织技术的基本思想是打乱原始流中数据单元的顺序，把原来连续的数据单元分散到不同的分组中去，当单个分组丢失时，仅导致重建流中多个小的间隔，而不是一个大的间隔。



6.9.2 实时传输协议RTP

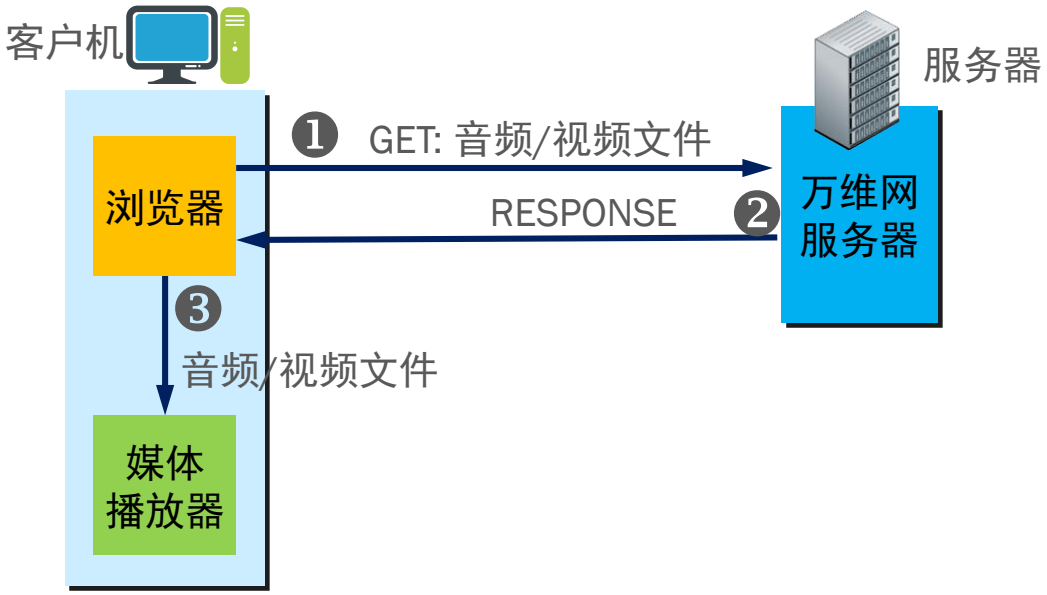
- TCP并不适合传输实时多媒体数据。相比而言，UDP更加适合于实时多媒体通信，但是UDP缺少实时多媒体网络应用所需的序号、时间戳等机制。
- **实时传输协议RTP** (Real-time Transport Protocol) 在UDP之上为实时多媒体网络应用提供端到端的传输服务。

经典DHT算法：Chord

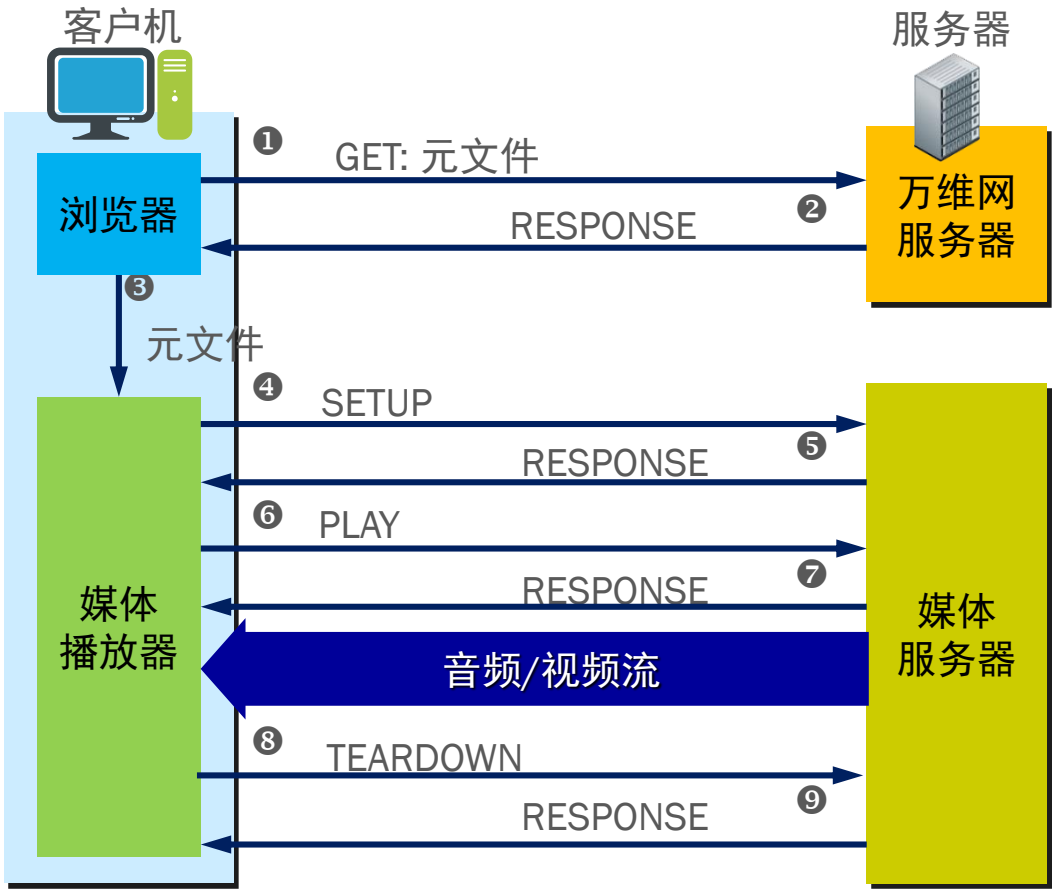


6.9.3 流式存储音频/视频

1. 从万维网服务器下载后播放

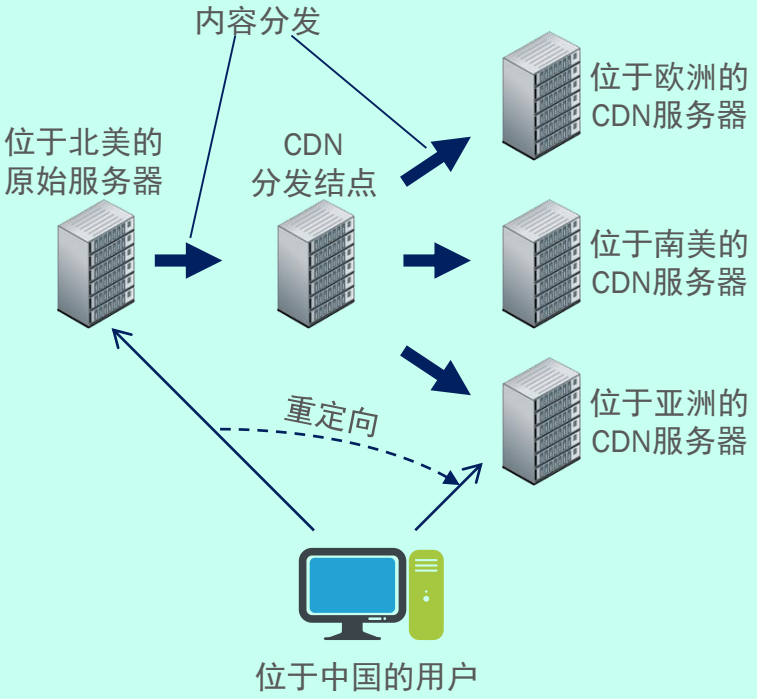


2. 使用媒体服务器边下载边播放



3. 内容分发网络CDN (Content Distribution Network)

- 将大量流式存储视频流按需传送到世界各地的大量用户是个巨大的挑战：
  - 用户可能远离服务器，会产生较大的时延和丢包率
  - 对于热播视频，大量用户的重复下载势必会消耗大量的带宽，造成服务器周边网络的严重拥塞
- CDN提前将多媒体数据直接推送到靠近用户的多个冗余服务器上，使用户能从最靠近自己的服务器上获取数据，避免大量重复数据的远程传输，大大减小了整个系统的传输时延和网络流量。

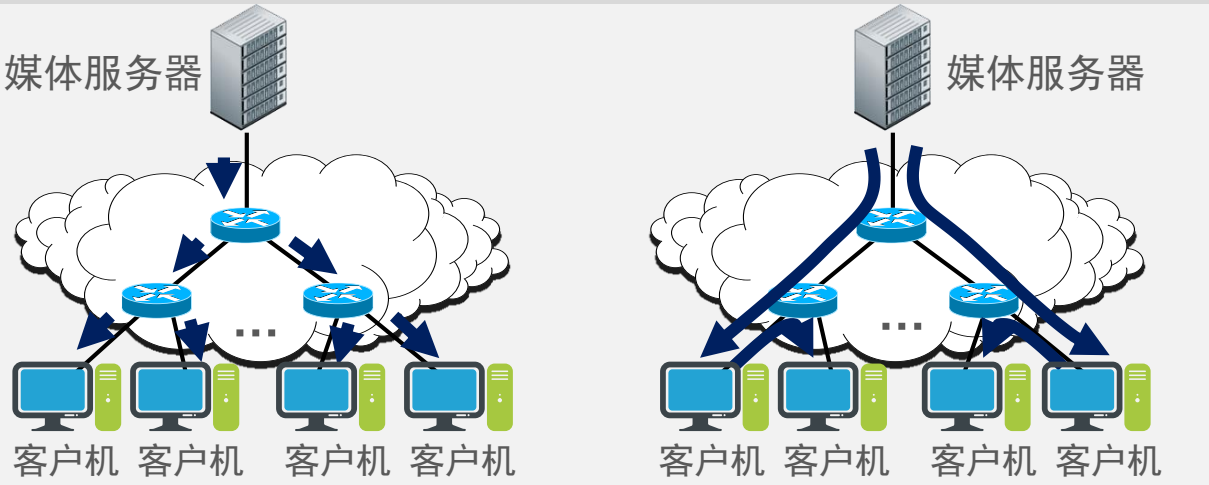


利用DNS实现用户请求重定向

6.9.4 流式实况音频/视频

- **个实况直播节目可能有大量用户在同时收听或收看，因此特别适合使用多播技术来实现流式实况音频/视频。**
- **由于IP多播还没有得到大规模的应用，今天的实况音频/视频的分发，通常是通过应用层多播（P2P应用层多播或内容分发网CDN）或多个独立的媒体服务器到客户机的单播来实现的。**

P2P应用层多播



(a) 基于 IP 多播的流媒体直播　　(b) 基于 P2P 应用层多播的流媒体直播

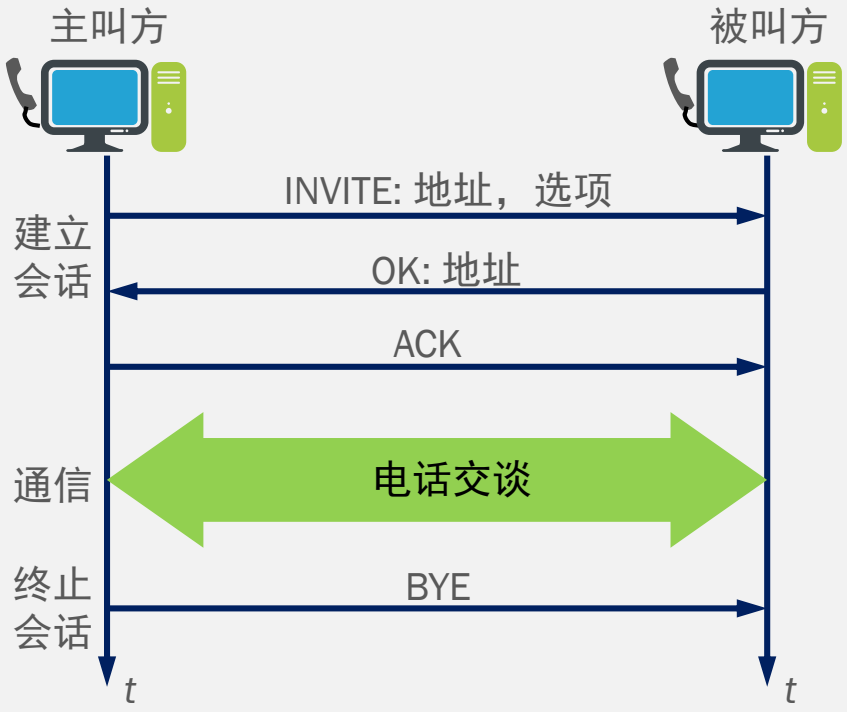
P2P应用层多播

- 采用P2P应用层多播技术，每个对等方既是服务的请求者，也是服务的提供者，因而请求服务的用户越多，每个用户获得的媒体服务质量反而越高。
- PPLive是当前最流行的因特网视频直播软件之一，其采用的技术就是P2P应用层多播。加入PPLive系统的用户越多，播放节目就越流畅。

6.9.5 实时交互音频/视频

- 典型的实时交互应用包括因特网电话和视频会议。在这方面IETF和ITU制定了很多标准[RFC 3261-3266]。
- **会话发起协议SIP** (Session Initiation Protocol)用于因特网电话，是一个由IETF制定的一套较为简单且实用的实时交互协议，能够用来定位用户、建立、管理和终止多媒体会话（呼叫），支持双方、多方或多播会话。

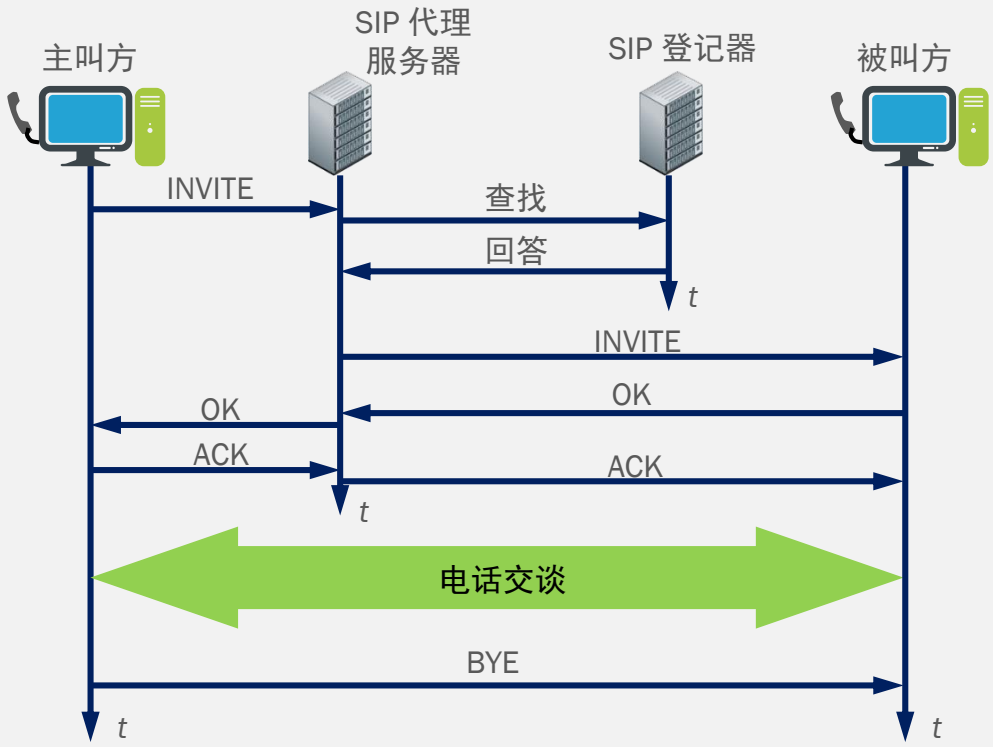
一个简单的 SIP 会话



SIP 的地址十分灵活

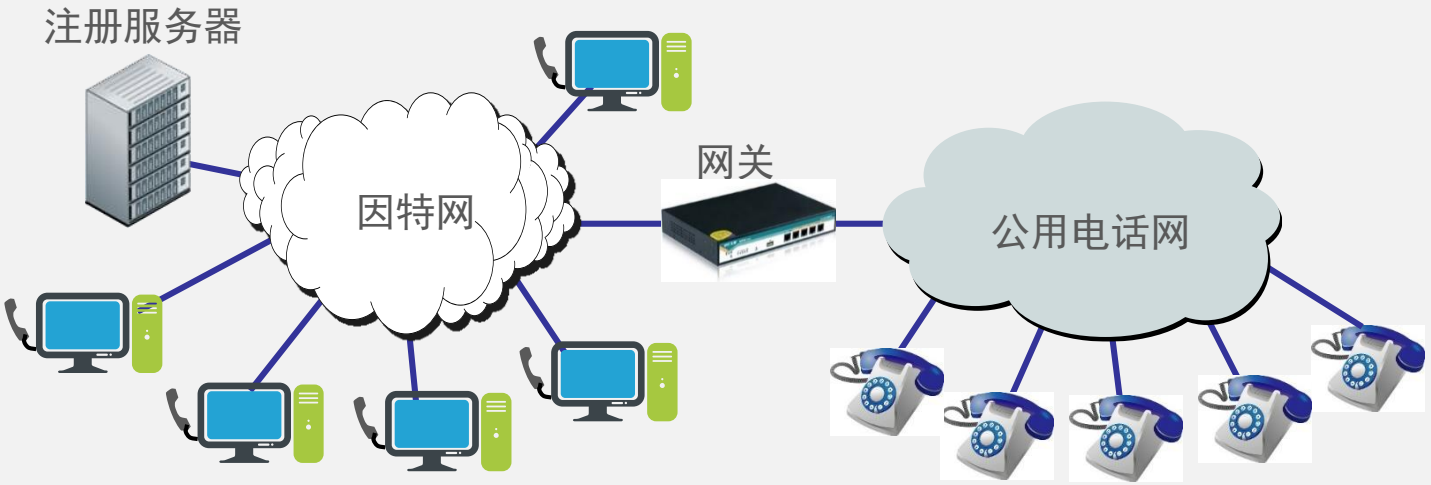
- 可以是电话号码，也可以是电子邮件地址、IP 地址或其他类型的地址。但一定要使用 SIP 的地址格式，例如：
- **电话号码**
  - sip:zhangsan@8625-87654321
- **IPv4 地址**
  - sip:zhangsan@201.12.34.56
- **电子邮件地址**
  - sip:zhangsan@public1.ptt.js.cn

SIP的用户定位



通过网关实现因特网端系统和公用电话的互通

- 在基于分组交换的因特网和基于电路交换的电话网之间部署若干网关，实现两个完全不同技术网络间的协议转换



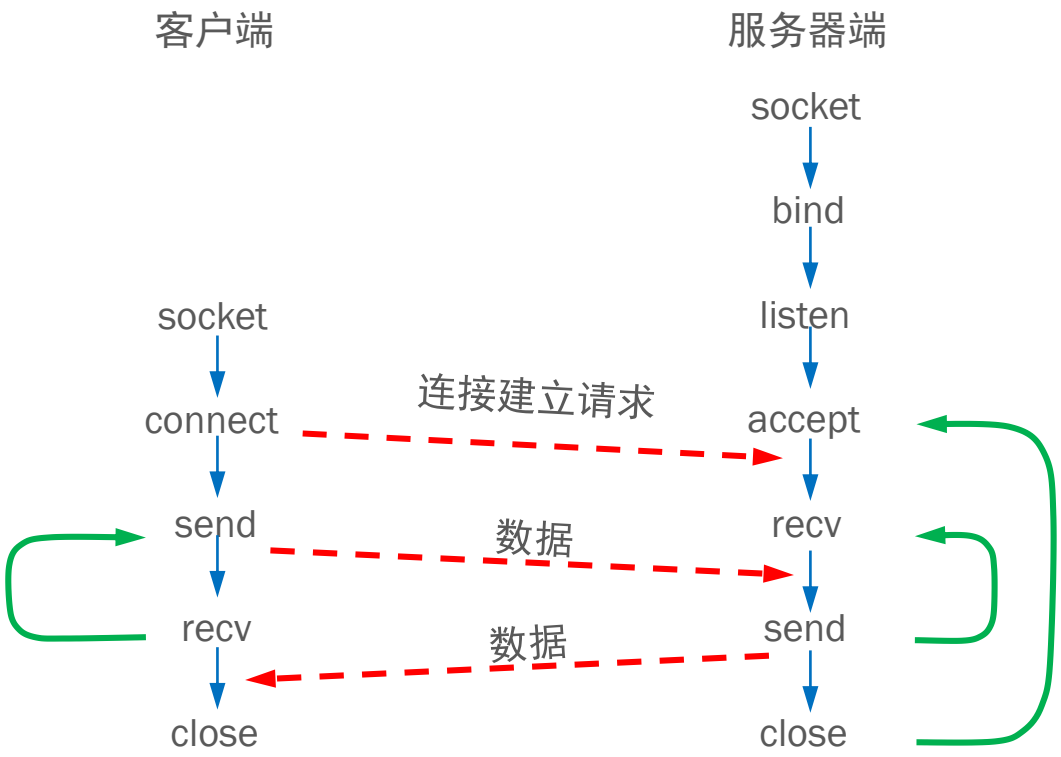
关于 Skype

- Skype是目前极为流行的P2P应用程序。
- Skype除了能提供PC到PC的因特网电话服务外，还提供PC到固定电话、固定电话到PC以及PC到PC的视频会议服务。
- Skype的用户定位采用了P2P技术，没有类似SIP的专门的注册服务器和代理服务器。
- Skype使用自己专用的协议，而且所有话音和控制分组都进行了加密。

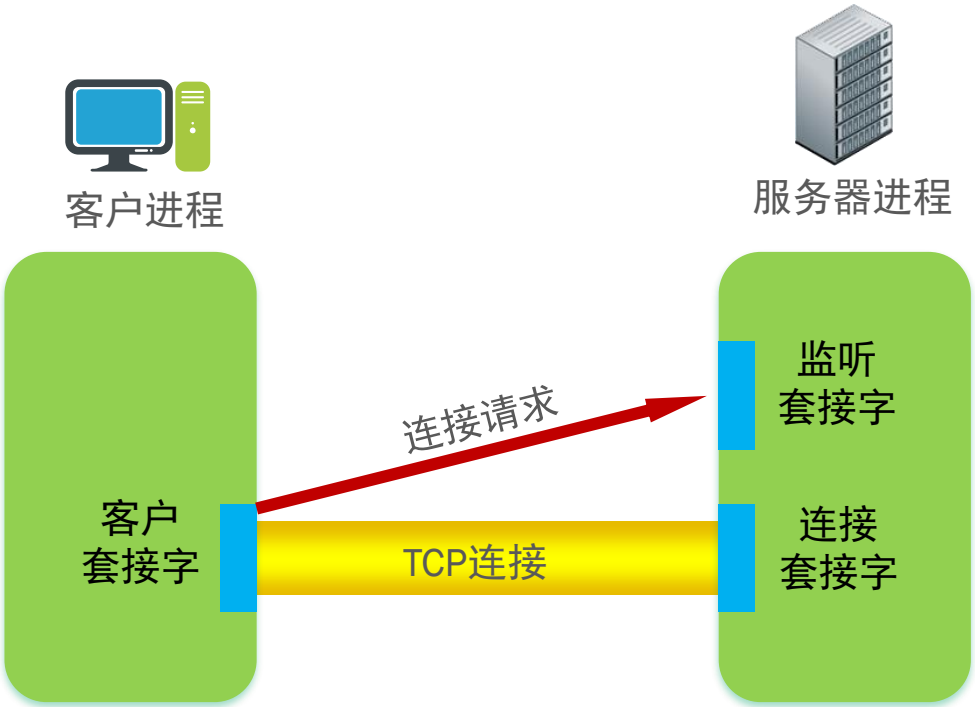
6.10 网络应用编程接口

- 应用程序要想执行网络操作必须通过操作系统为应用程序操作网络所提供的接口，这个接口通常称为网络应用编程接口API (Application Programming Interface)。
- 最著名的是最初由加州大学伯克利分校的UNIX小组开发的套接字(socket) API，现在几乎所有流行的操作系统都支持它。在此基础上，微软联合其它公司制定了Windows下的网络应用编程接口，即Windows Socket规范。

6.10.1 TCP套接字编程



客户机套接字、监听套接字和连接套接字



6.10.2 一个简单的代码实例

- 一个非常简单的TCP应用的代码实例：“Hello World!”
- 使用的是Windows操作系统的套接字API：Windows Socket 2.0

```
3       #include <stdio.h>
4       #include <winsock2.h>
5       #pragma comment(lib, "ws2_32.lib")
```

服务器代码实例

```
7           int main()
8           {
9               WSADATA wsaData;
10              WSAStartup(0x202, &wsaData); //加载WinSock动态链接库
11              // 创建监听套接字
12              SOCKET s;
13              s = socket(AF_INET, SOCK_STREAM, 0);
14              // 设置服务器端地址
15              struct sockaddr_in serveraddr;
16              memset((void *)&serveraddr, 0, sizeof(serveraddr));
17              serveraddr.sin_family = AF_INET;
18              serveraddr.sin_addr.s_addr = htonl(INADDR_ANY); //选择服务器
的任一IP地址
19              serveraddr.sin_port = htons(8888);
20              // 将服务器端地址与监听套接字绑定
```

服务器代码实例

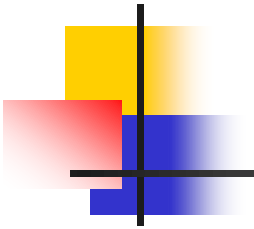
```
21          bind(s, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
22          listen(s, 10);
23          // 接受连接请求并获得连接套接字
24          SOCKET ss;
25          ss = accept(s, NULL, NULL);
26          closesocket(s);// 关闭监听套接字
27          // 显示接收的字符串
28          char buf[13];
29          recv(ss, buf, 13, 0);
30          printf("%s\n", buf);
31          // 关闭连接套接字
32          closesocket(ss);
33          WSACleanup(); //注销并释放所有套接字资源
34          return 0;
35      }
```

客户端代码实例

```
7           int main()
8           {
9               WSADATA wsaData;
10              WSAStartup(0x202, &wsaData); //加载WinSock动态链接库
11              // 创建客户端套接字
12              SOCKET s;
13              s = socket(AF_INET, SOCK_STREAM, 0);
14              // 设置服务器端地址
15              struct sockaddr_in serveraddr;
16              memset((void *)&serveraddr, 0, sizeof(serveraddr));
17              serveraddr.sin_family = AF_INET;
18              serveraddr.sin_addr.s_addr = inet_addr("127.0.0.1"); //环回测
试地址
19              serveraddr.sin_port = htons(8888);
```

客户端代码实例

```
20          // 连接服务器
21          connect(s, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
22          // 发送字符串
23          send(s, "Hello World!", 13, 0);
24          // 关闭客户端套接字
25          closesocket(ss);
26          WSACleanup(); //注销并释放所有套接字资源
27          return 0;
28      }
```



谢谢！