

# 第 6 章 应用层



# 第 6 章 应用层

---



- 6.1 域名系统 DNS
- 6.2 文件传送协议
- 6.3 远程终端协议 TELNET
- 6.4 万维网 WWW
- 6.5 电子邮件

# 应用层协议的特点



- 每个应用层协议都是为了解决某一类应用问题，而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。
- 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) 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。客户是服务请求方，服务器是服务提供方。

# 6.1 域名系统 DNS

---



- 6.1.1 域名系统概述
- 6.1.2 互联网的域名结构
- 6.1.3 域名服务器

## 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。
- 互联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 互联网的域名结构



- 互联网采用了层次树状结构的命名方法。
- 任何一个连接在互联网上的主机或路由器，都有一个**唯一**的层次结构的名称，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。

# 互联网的域名空间

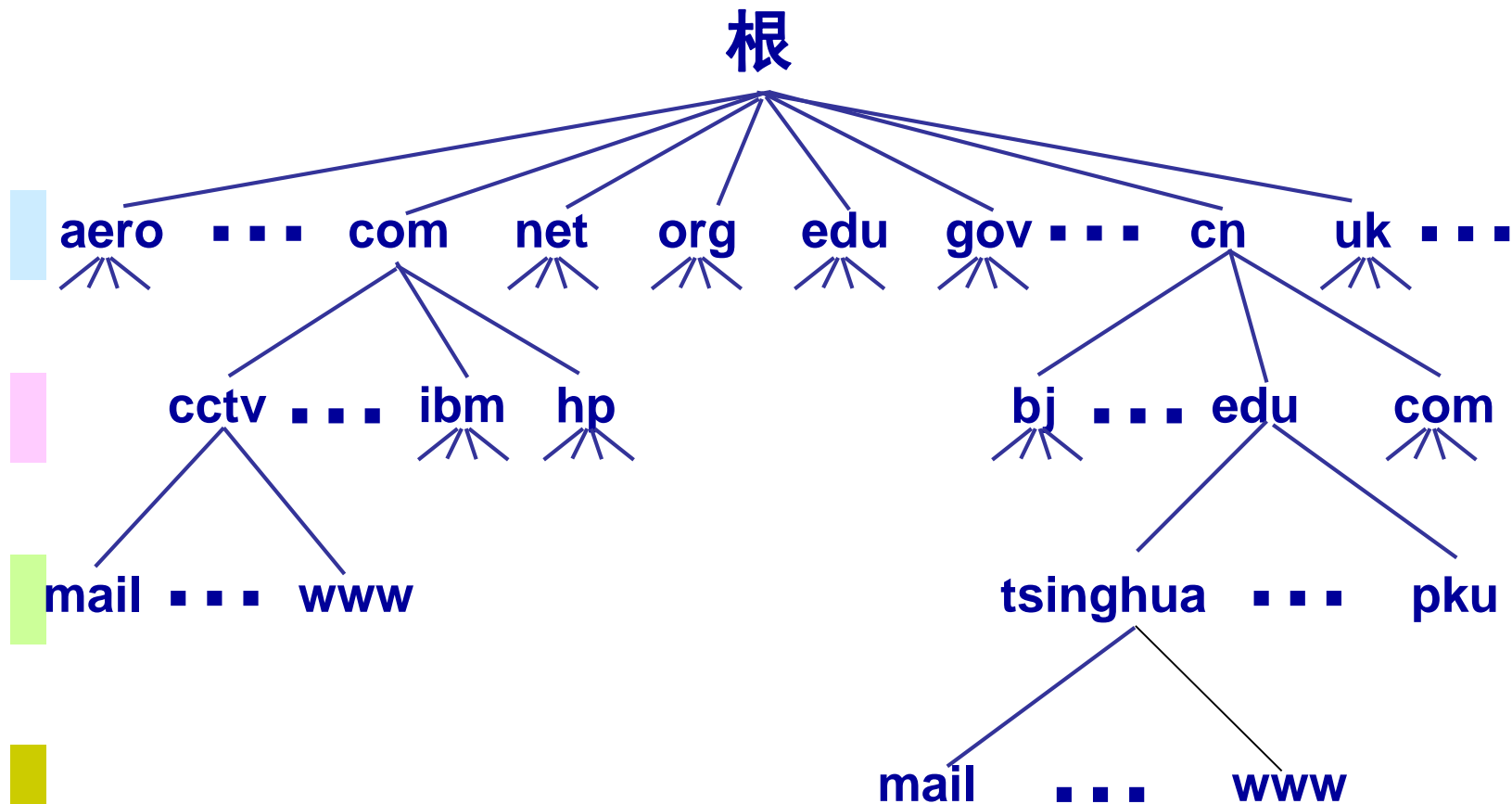


顶级域名

二级域名

三级域名

四级域名



# 顶级域名 TLD (Top Level Domain)



## ■ (1) 国家顶级域名 nTLD

.cn 表示中国,

.us 表示美国,

.uk 表示英国, 等等。



# 顶级域名 TLD (Top Level Domain)



## ■ (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）

# 新增加了下列的通用顶级域名



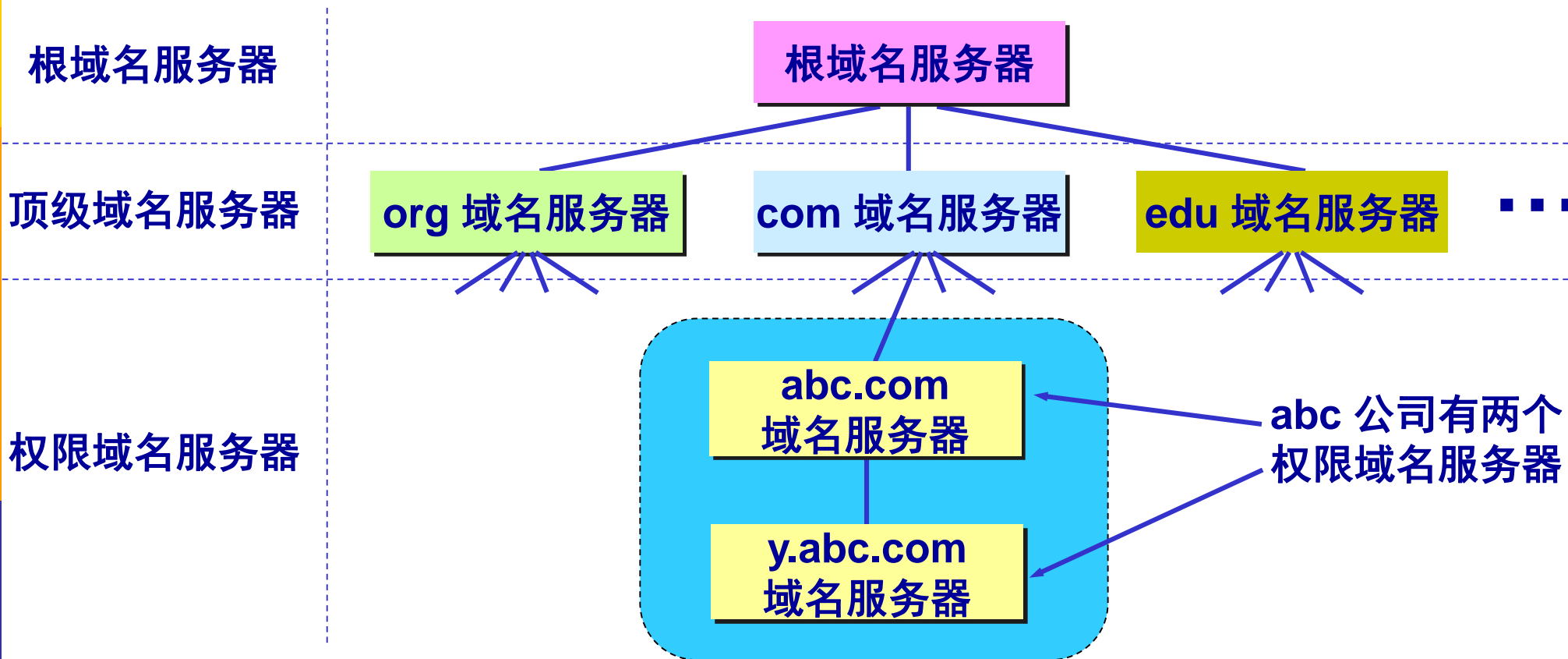
- .aero** (航空运输企业)
- .biz** (公司和企业)
- .cat** (加泰隆人的语言和文化团体)
- .coop** (合作团体)
- .info** (各种情况)
- .jobs** (人力资源管理者)
- .mobi** (移动产品与服务的用户和提供者)
- .museum** (博物馆)
- .name** (个人)
- .pro** (有证书的专业人员)
- .travel** (旅游业)

## 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**，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。

# 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



- 根域名服务器是**最高层次**的域名服务器，也是**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，若要对互联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。
- 在互联网上共有 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



- 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2016年2月，全世界已经在 588 个地点安装了根域名服务器，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 L 的地点分布图



根域名服务器 L 分布在世界 150 个地点

- 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



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

# 权限域名服务器



- 负责一个区的域名服务器。
- 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



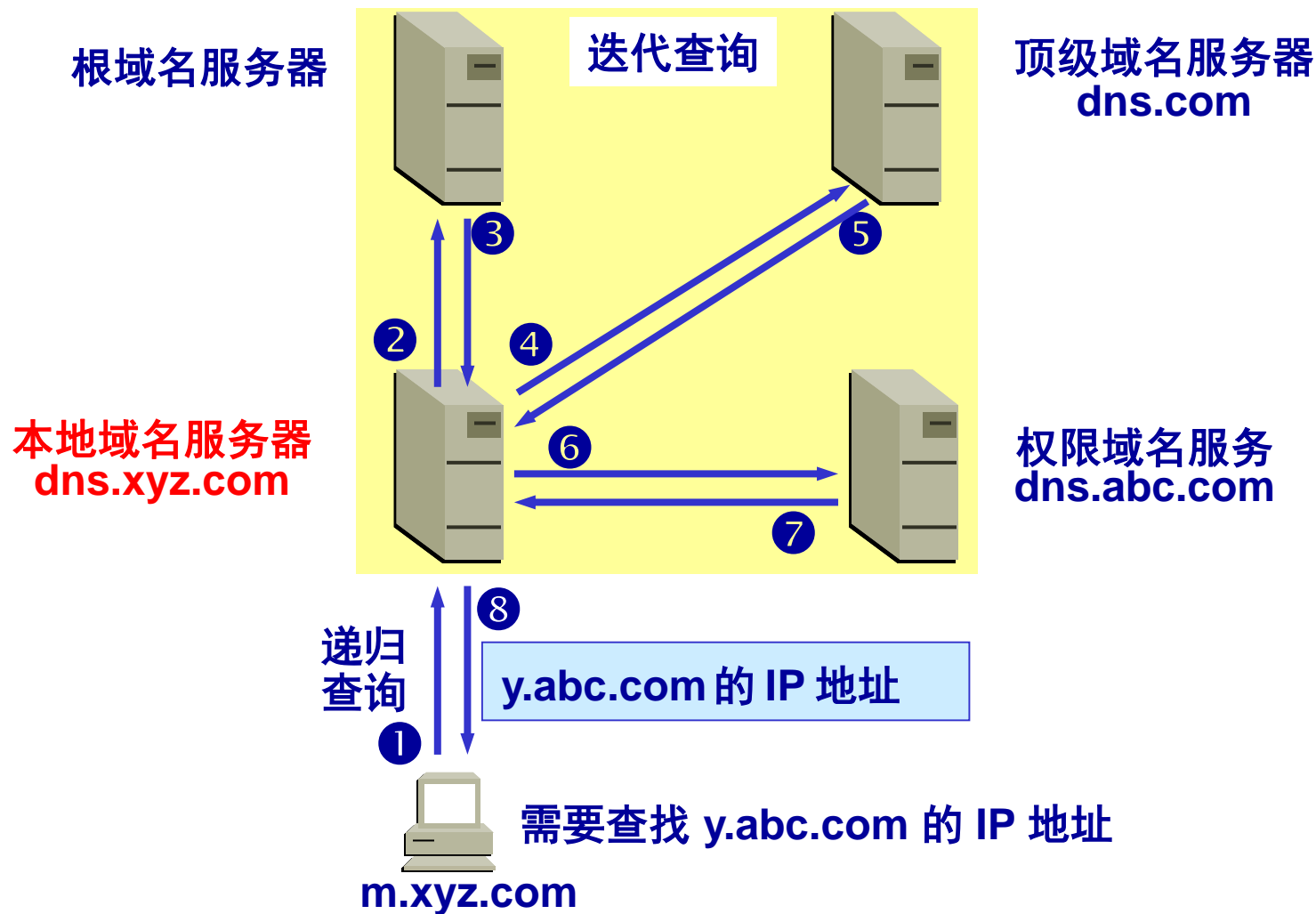
- 本地域名服务器对域名系统非常重要。
- 当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器。
- 每一个互联网服务提供者 ISP，或一个大学，甚至一个大学里的系，都可以拥有一个本地域名服务器，
- 这种域名服务器有时也称为**默认域名服务器**。

# 域名的解析过程

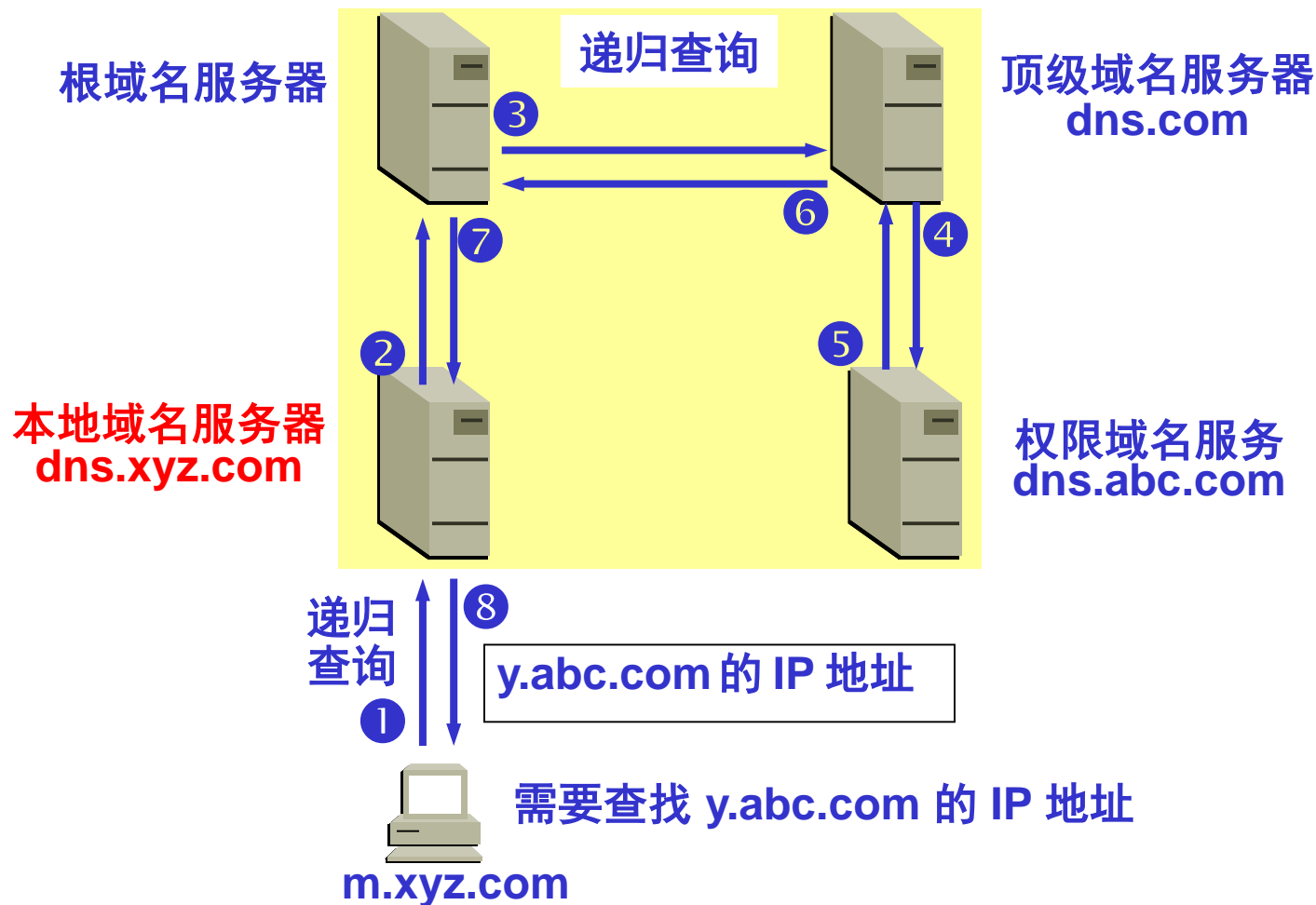


- **主机向本地域名服务器的查询一般都是采用递归查询。**如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。
- **本地域名服务器向根域名服务器的查询通常是采用迭代查询。**当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。

# 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，使互联网上的 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。

## 6.2 文件传送协议

---



- 6.2.1 FTP 概述
- 6.2.2 FTP 的基本工作原理
- 6.2.3 简单文件传送协议 TFTP



## 6.2.1 FTP 概述



- **文件传送协议** FTP (File Transfer Protocol) 是互联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 很早就成为了互联网的正式标准。

# 1. 文件传输协议概述



- 文件传输协议(File Transfer Protocol, FTP)是用于在网络上进行文件传输的一套标准协议。
- 文件传送协议能把多种类型的文件或文档从不同厂家生产的计算机、手机等智能终端通过有线或无线的计算机网络发送（复制）到远程的另一台计算机、手机等智能设备中。

# 文件传送并非很简单的问题



- FTP是网络环境中的一项基本应用。
- 初看起来，在两个主机之间传送文件似乎是很简单的事情。
- 其实不然，原因是众多的 计算机厂商研制出的文件系统种类繁多，且差别非常大。

# 网络环境下复制文件的复杂性



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

## 2. FTP协议的特点



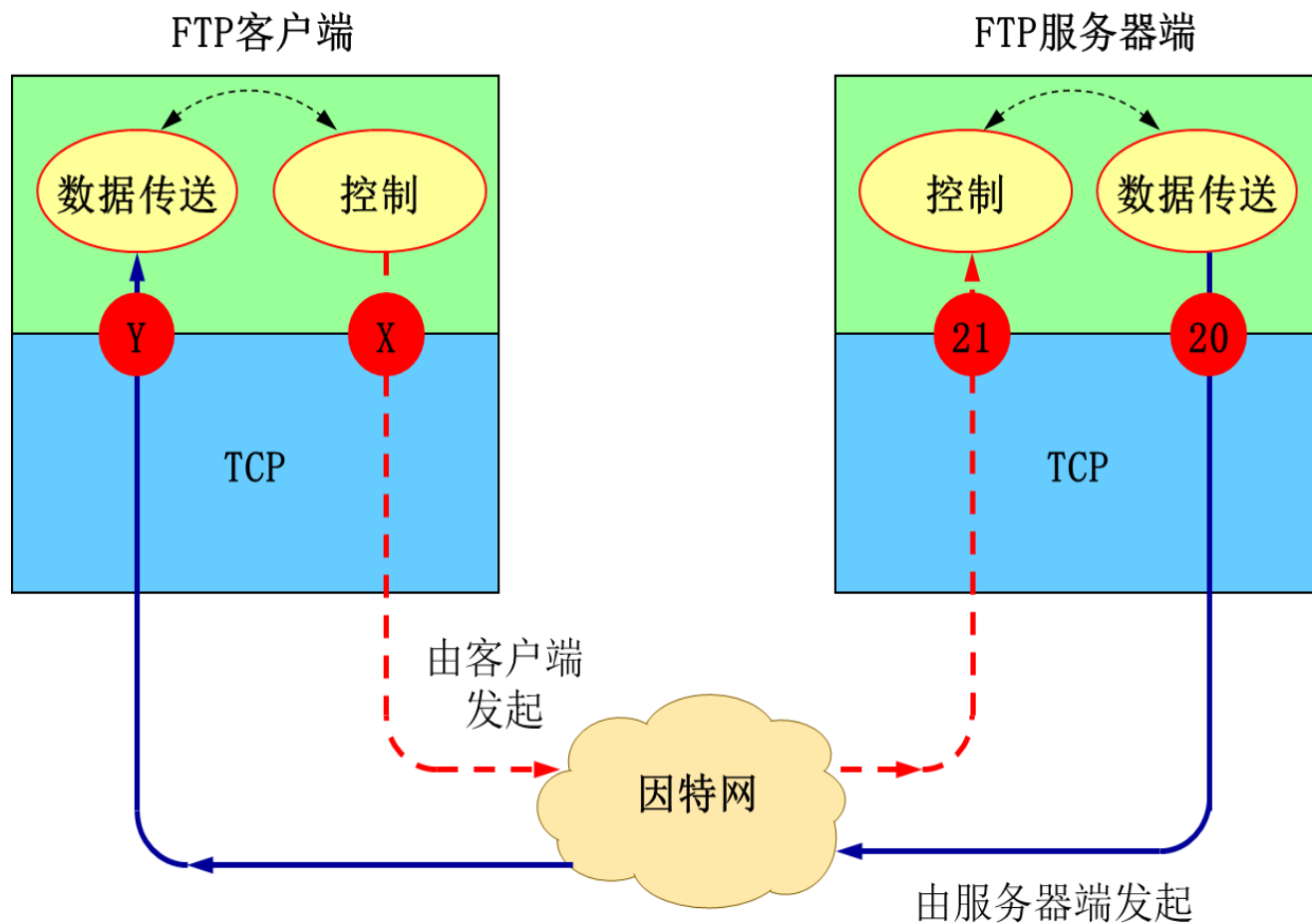
- (1) FTP只提供文件传送的一些基本的服务，它使用TCP可靠的传输服务。
- (2) FTP的主要功能是减少或消除在不同操作系统下处理文件的不兼容性。
- (3) FTP使用客户 / 服务器方式。一个FTP服务器进程可同时为多个客户进程提供服务。FTP的服务器进程由两大部分组成：主进程负责接受新的请求，其他的若干个从属进程负责处理单个请求。

### 3.主进程的工作步骤



- (1) 打开熟知端口（端口号为21），使客户进程能够连接上。
- (2) 等待客户进程发出连接请求。
- (3) 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。
- (4) 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的。

# 4.FTP 使用的两个 TCP 连接



# 两个不同的端口号



- 当客户进程向服务器进程发出建立连接请求时，要寻找连接服务器进程的**熟知端口(21)**，同时还要告诉服务器进程自己的另一个端口号码，用于建立数据传送连接。
- 接着，服务器进程用自己传送数据的**熟知端口(20)**与客户进程所提供的端口号码建立数据传送连接。
- 由于 **FTP** 使用了两个不同的端口号，所以数据连接与控制连接不会发生混乱。



# 使用两个不同端口号的好处

---



- 使协议更加简单和更容易实现。
- 在传输文件时还可以利用控制连接（例如，客户发送请求终止传输）。

# FTP的文件传输过程



**ftp Client** **ftp Server**

TCP:21 控制连接初始化, 控制端口

SYN

Port xxxx -----> Port 21

SYN, ACK

Port xxxx <----- Port 21

ACK

Port xxxx -----> Port 21

控制操作: 用户列目录或传输文件

Port, IP, Port yyyy

Port xxxx -----> Port 21

Port Successful

Port xxxx <----- Port 21

List, Retr or Stor

Port xxxx -----> Port 21

TCP:20 数据连接初始化, 数据端口

SYN

Port yyyy <----- Port 20

SYN, ACK

Port yyyy -----> Port 20

ACK

Port yyyy <----- Port 20

数据操作: 数据传输

Data

Port yyyy <-----> Port 20

.....



- 先建立控制连接（客户端发起）；
- 接下来建立数据连接；
- 最后是文件传输。
- 当数据传输完成后，先断开数据连接；
- FTP会话结束后，断开控制连接。

## 6.4 万维网 WWW



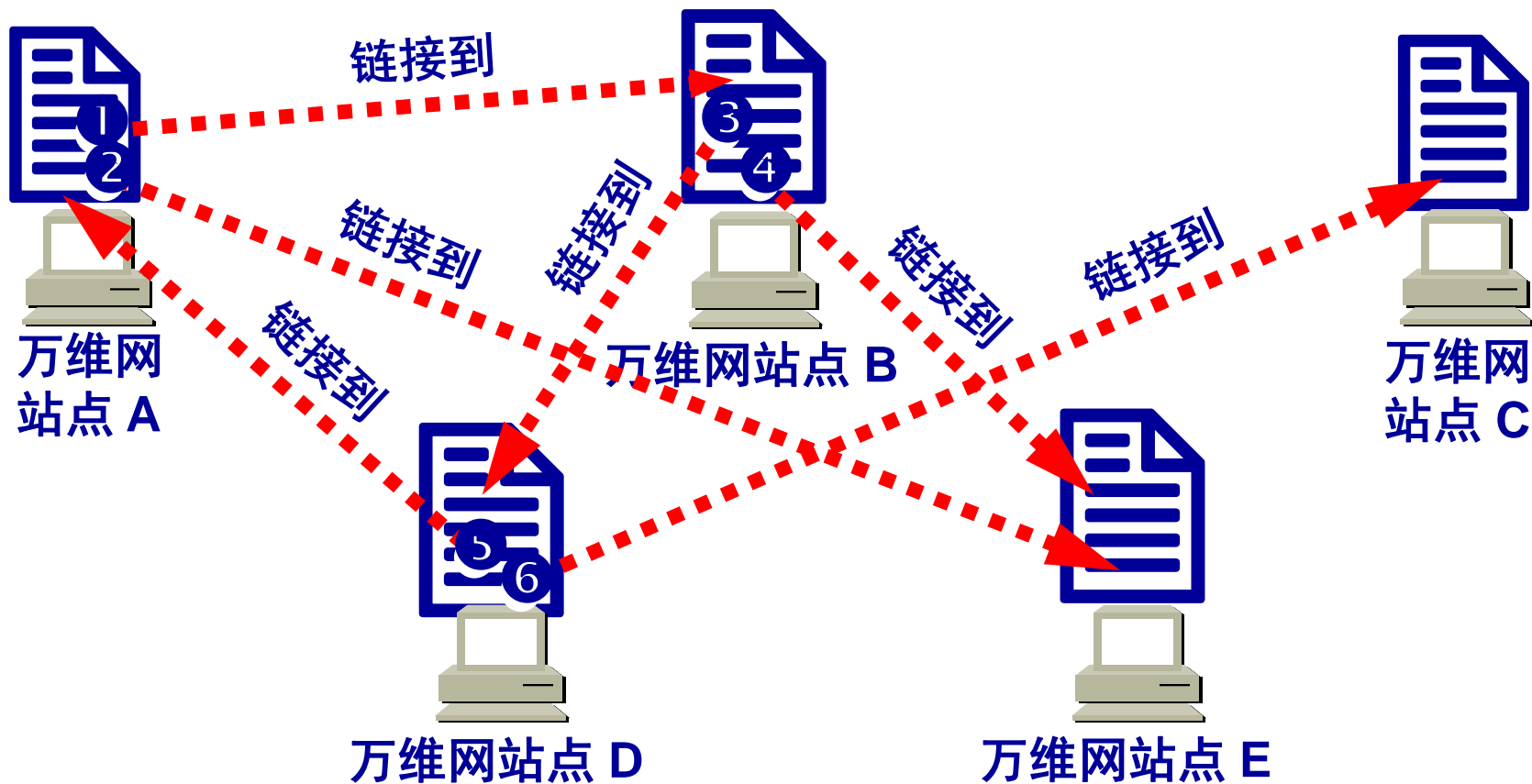
- 6.4.1 万维网概述
- 6.4.2 统一资源定位符 URL
- 6.4.3 超文本传送协议 HTTP
- 6.4.4 万维网的文档
- 6.4.5 万维网的信息检索系统
- 6.4.6 博客和微博
- 6.4.7 社交网络

## 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互联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互联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互联网上的万维网文档？
- 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互联网的范围具有唯一的标识符 URL。

# 万维网必须解决的问题



(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，它使用 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互联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互联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。

# 万维网必须解决的问题



(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**）。

## 6.4.2 统一资源定位符 URL



### 1. URL 的格式

- 资源定位符 URL 是对可以从互联网上得到的资源的位置和访问方法的一种简洁表示。
- 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 是与互联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



- 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>

规定的格式

# 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  
在互联网中的域名**



# 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

# URL 的一般形式（续）



- 现在有些浏览器为了方便用户，在输入 URL 时，可以把最前面的“http://”甚至把主机名最前面的“www”省略，然后浏览器替用户把省略的字符添上。
- 例如，用户只要键入 ctrip.com，浏览器就自动把未键入的字符补齐，变成 `http://www.ctrip.com`。

## 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 就指到互联网上的某个主页 (home page)。

更复杂一些的路径是指向层次结构的从属页面。例如：

`http://www.tsinghua.edu.cn/publish/newthu/newthu_cnt/faculties/index.html`

主机域名

路径名

## 6.4.3 超文本传送协议 HTTP

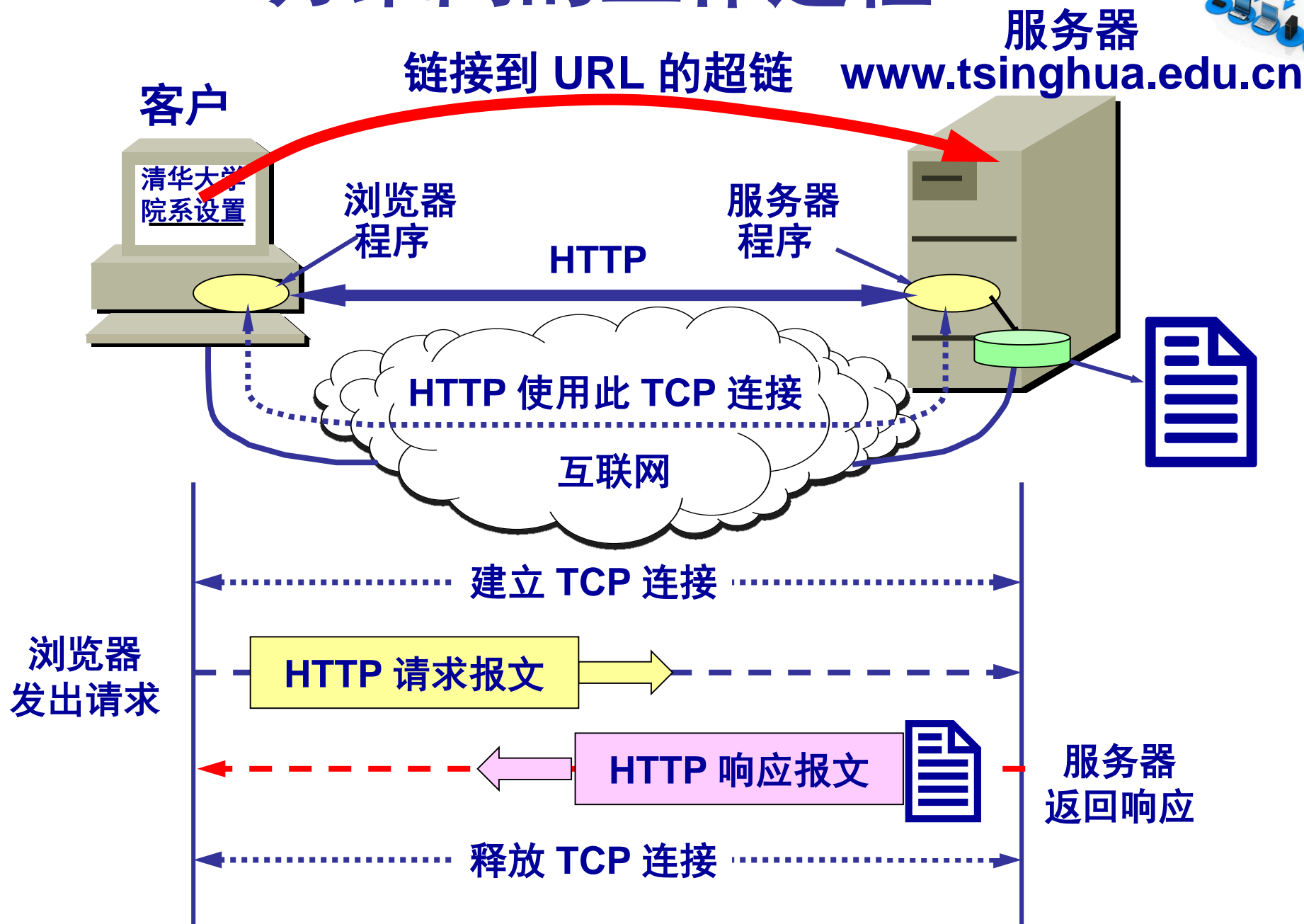


### 1. HTTP 的操作过程

- 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。
- 从层次的角度看，HTTP 是**面向事务的** (transaction-oriented) 应用层协议，它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



# 万维网的工作过程（续）



- 每个万维网网点都有一个服务器进程，它不断地**监听 TCP 的端口 80**，以便发现是否有浏览器向它发出连接建立请求。
- 一旦监听到连接建立请求并建立了 TCP 连接之后，浏览器就向万维网服务器发出浏览某个页面的请求，服务器接着就返回所请求的页面作为响应。
- 最后，TCP 连接就被释放了。

# 万维网的工作过程（续）



- 在浏览器和服务端之间的**请求和响应**的交互，必须按照规定的格式和遵循一定的规则。**这些格式和规则就是超文本传送协议 HTTP。**
- HTTP 规定在 HTTP 客户与 HTTP 服务器之间的每次交互，都由一个 ASCII 码串构成的请求和一个类似的通用互联网扩充，即“类 MIME (MIME-like)”的响应组成。
- **HTTP 报文通常都使用 TCP 连接传送。**

# 用户浏览页面的两种方法



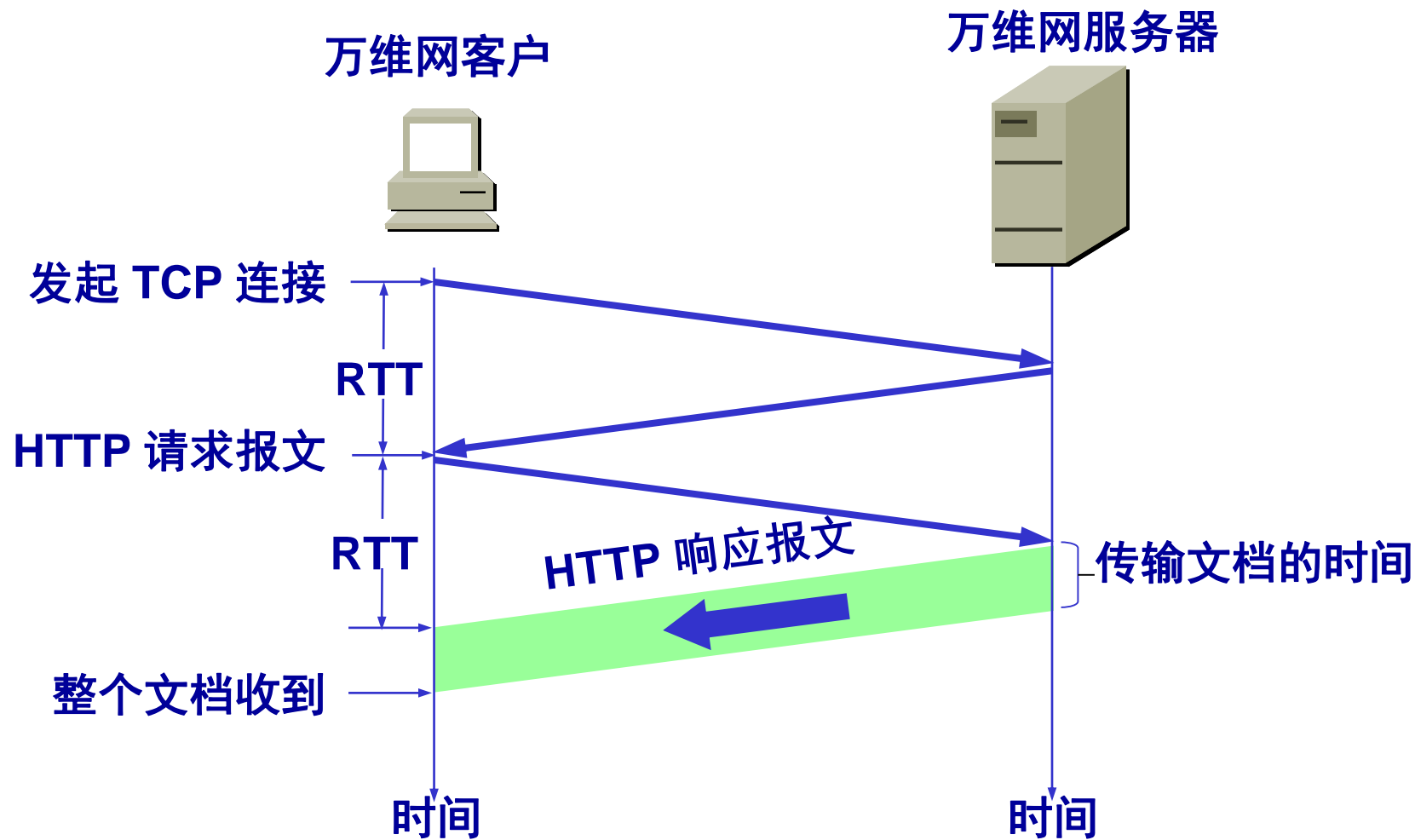
- 1, 在浏览器的地址窗口中键入所要找的页面的URL。
- 2, 在某一个页面中用鼠标点击一个可选部分, 这时浏览器会自动在互联网上找到所要链接的页面。

# HTTP 的主要特点



- HTTP 使用了面向连接的 TCP 作为运输层协议，保证了数据的可靠传输。
- 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。
- 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)。

# 请求一个万维网文档所需的时间



# 持续连接



- **HTTP/1.1 协议使用持续连接 (persistent connection)。**
- 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。



## 2. 代理服务器

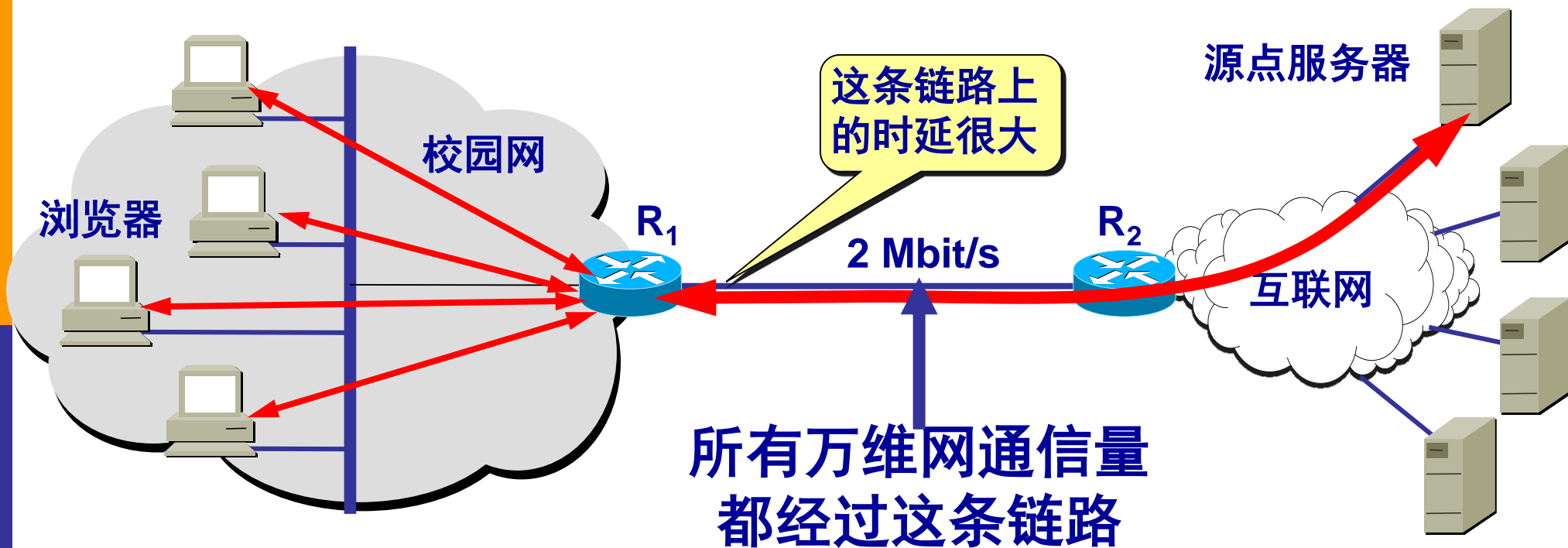


- **代理服务器** (proxy server) 又称为**万维网高速缓存** (Web cache), 它代表浏览器发出 HTTP 请求。
- 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求相同的新请求到达时, 万维网高速缓存就把暂存的响应发送出去, 而不需要按 URL 的地址再去互联网访问该资源。

# 使用高速缓存可**减少** 访问互联网服务器的时延



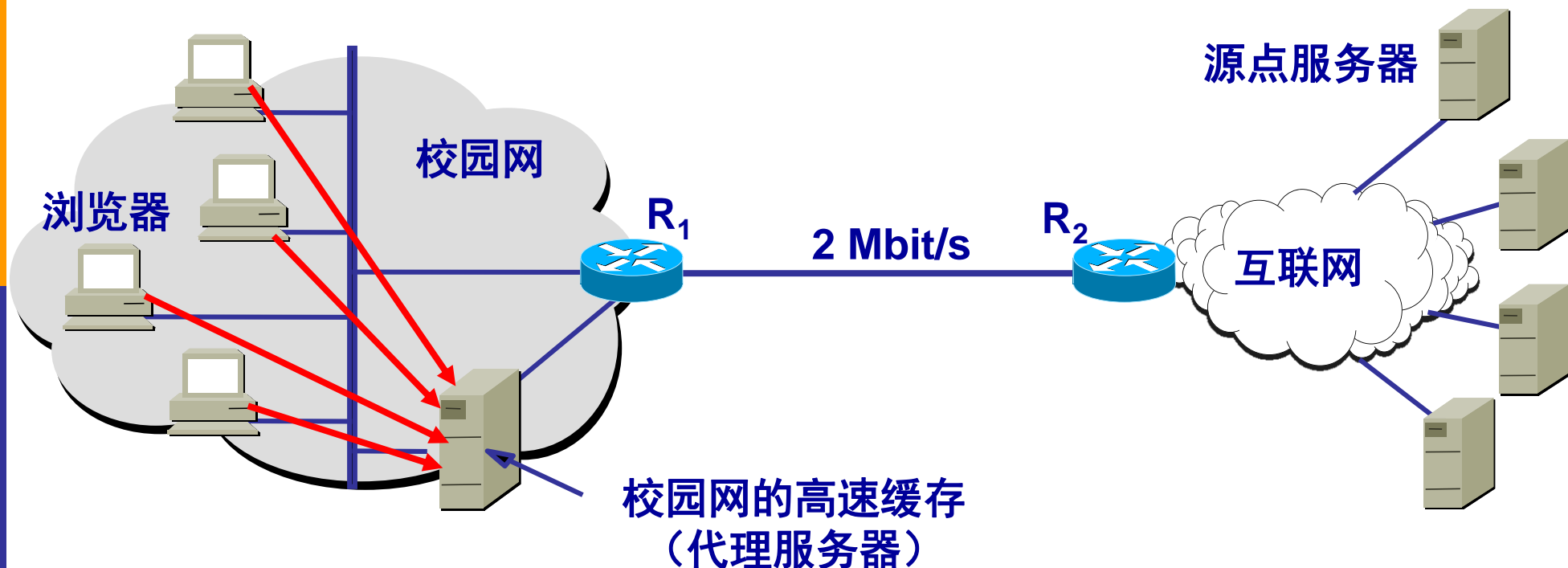
没有使用高速缓存的情况



# 使用高速缓存的情况



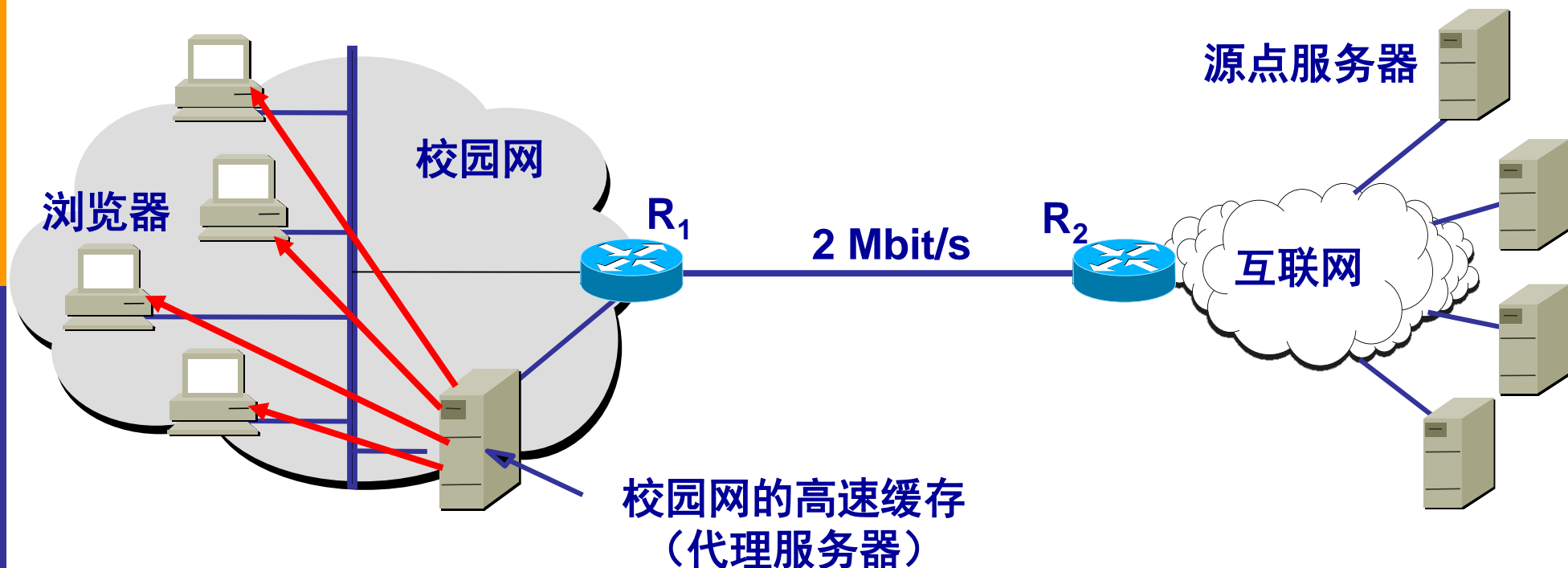
(1) 浏览器访问互联网的服务器时，要先与校园网的高速缓存建立 TCP 连接，并向高速缓存发出 HTTP 请求报文。



# 使用高速缓存的情况



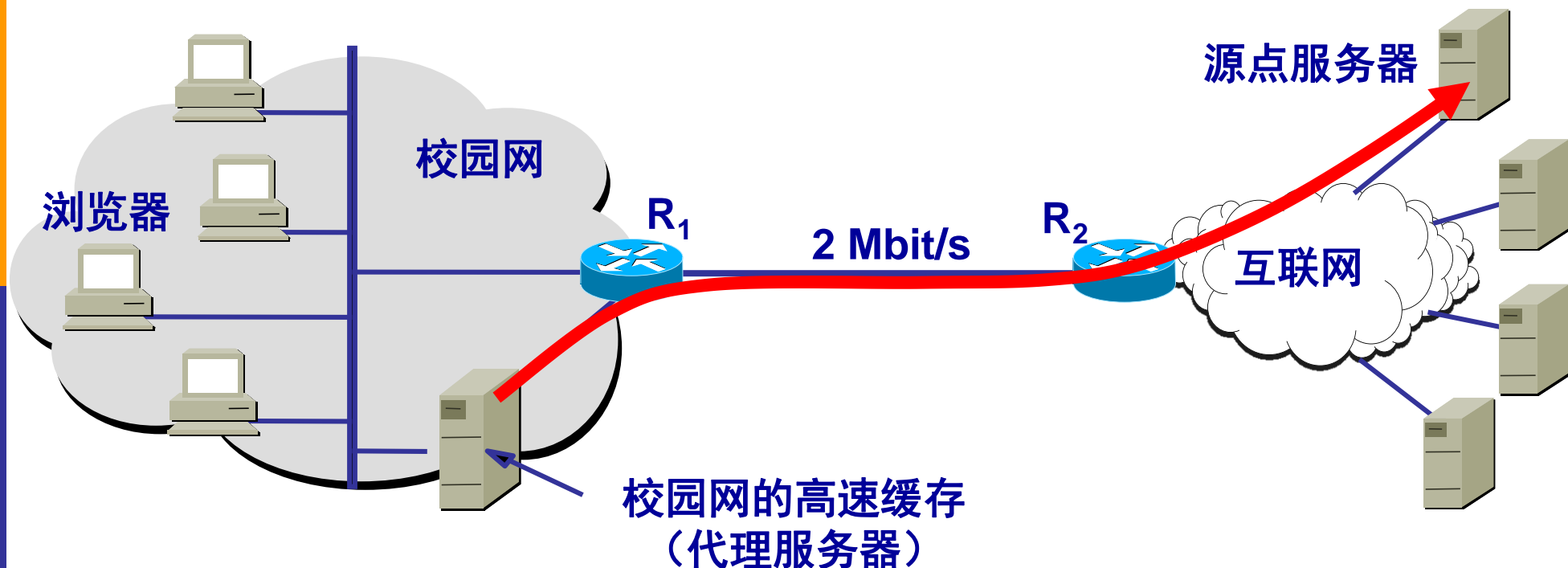
(2) 若高速缓存已经存放了所请求的对象，则将此对象放入 HTTP 响应报文中返回给浏览器。



# 使用高速缓存的情况



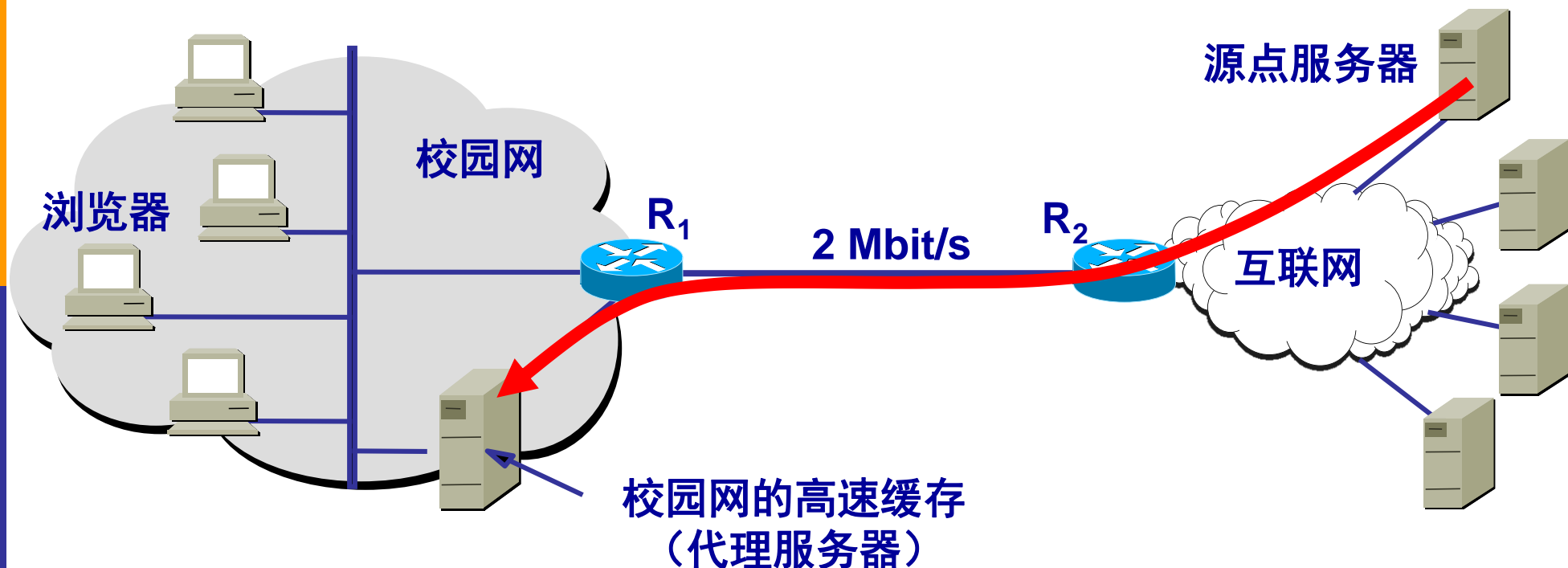
(3) 否则，高速缓存就代表发出请求的用户浏览器，与互联网上的源点服务器建立 TCP 连接，并发送 HTTP 请求报文。



# 使用高速缓存的情况



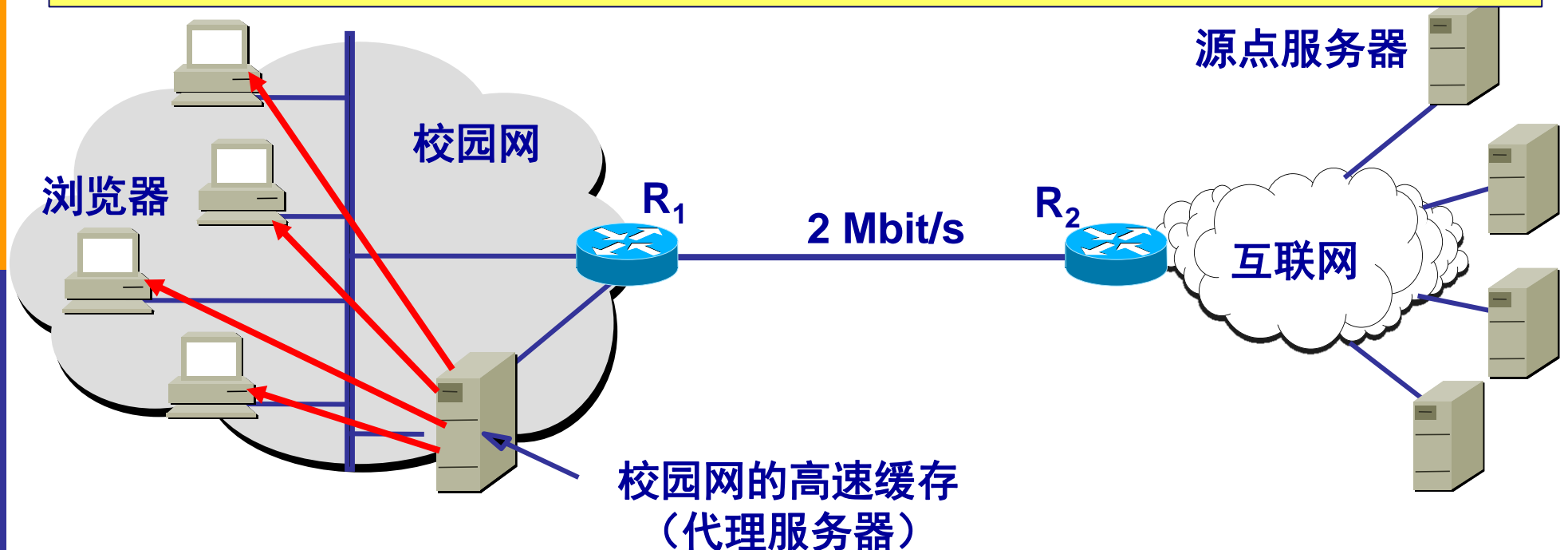
(4) 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存。



# 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

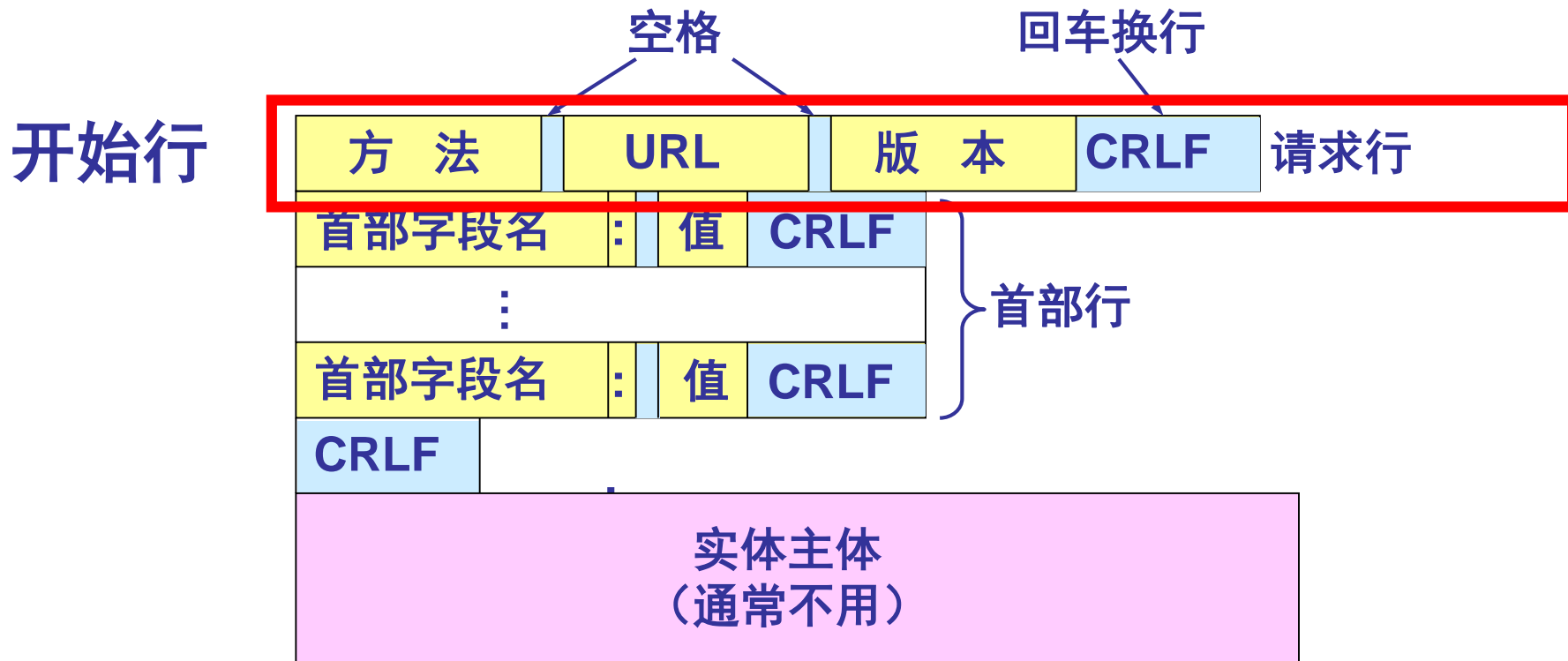


HTTP 有两类报文：

- **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。

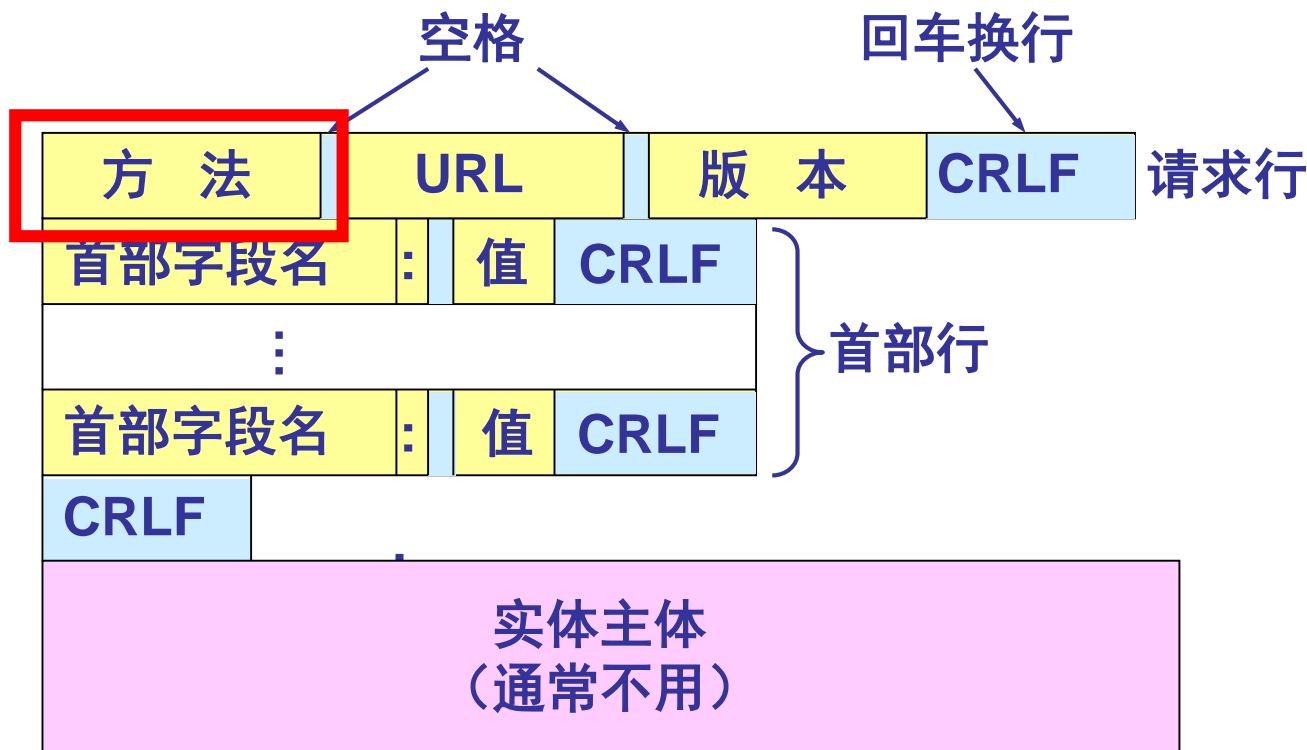


# HTTP 的报文结构（请求报文）



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）



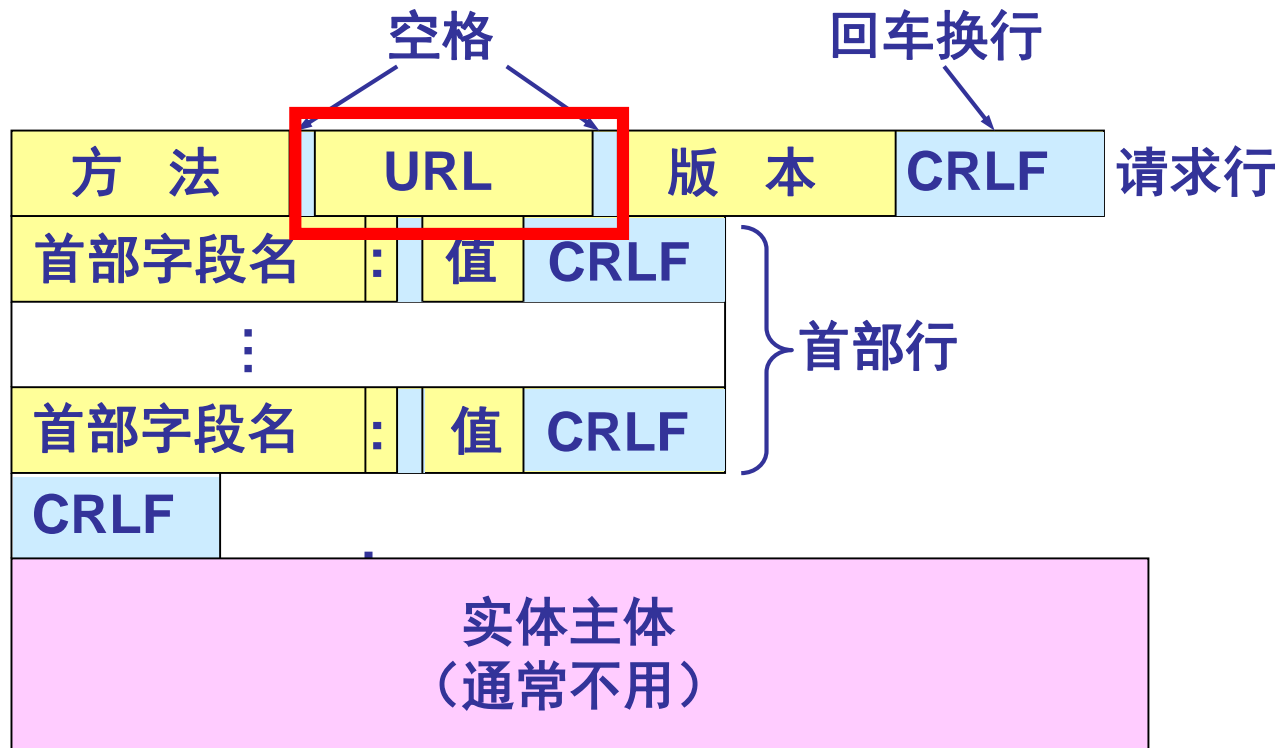
**“方法”**是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的。

# HTTP 请求报文的一些方法



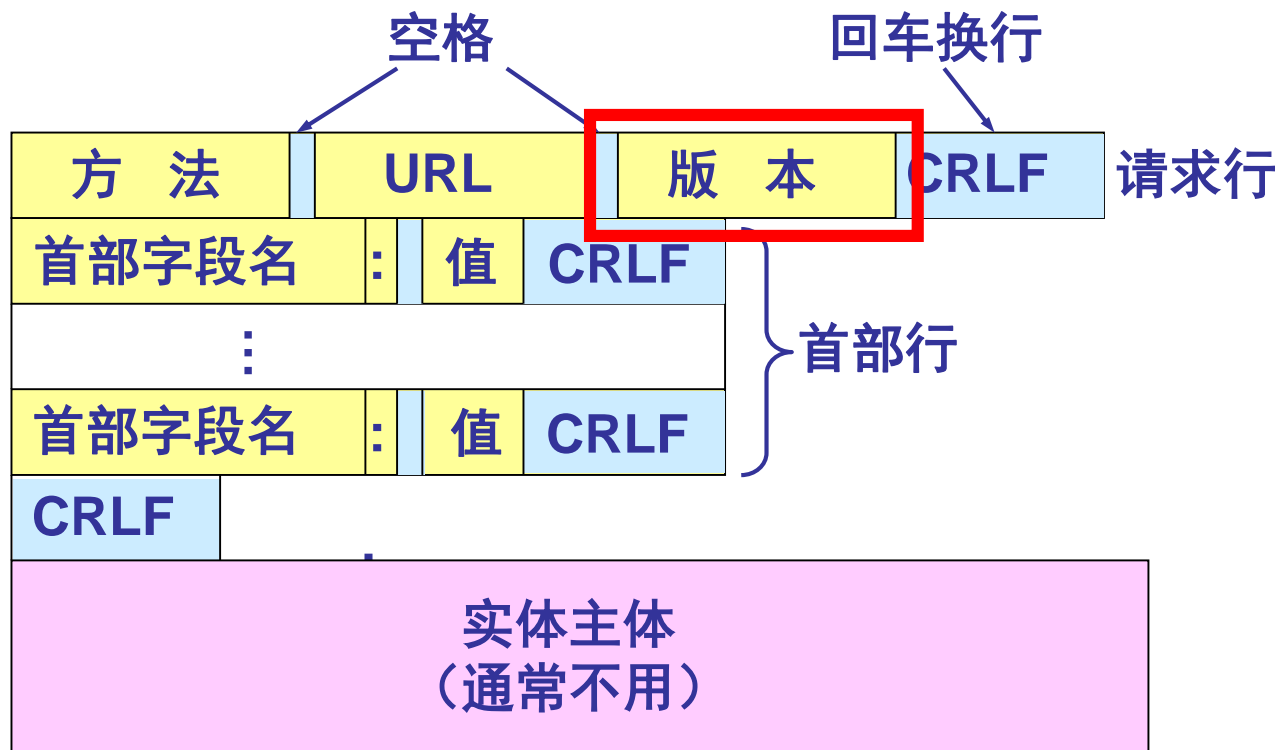
方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL 所标志的信息
HEAD	请求读取由 URL 所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL 下存储一个文档
DELETE	删除指明的 URL 所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

# HTTP 的报文结构（请求报文）



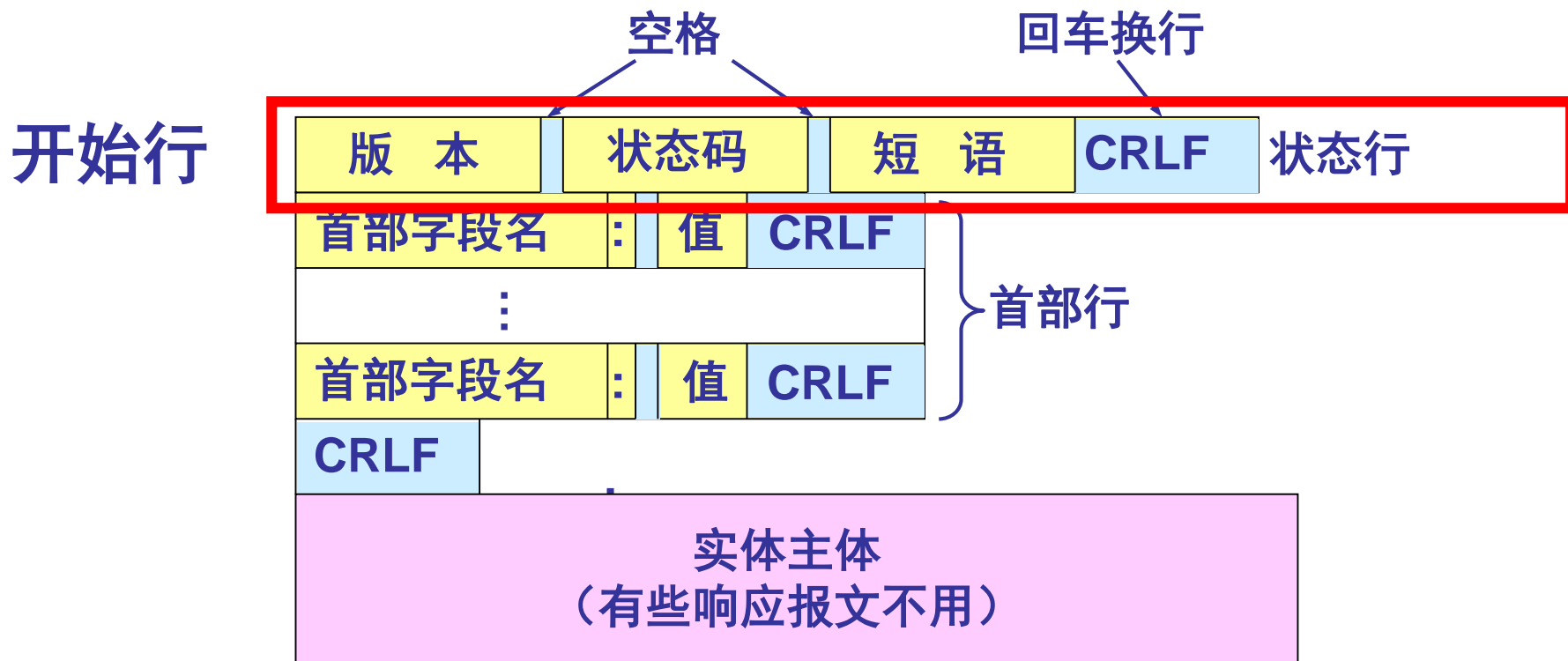
“URL” 是所请求的资源的 URL。

# 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。



## 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>

1 级标题

<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>

<P>这是第一个段落。虽然很短，但它仍是一个段落。</P>

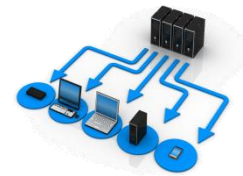
<P>这是第二个段落。</P>

</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

# 两种不同的链接



- HTML还规定了链接的设置方法。每个链接都有一个起点和终点。
- **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。

# XML



- XML (Extensible Markup Language) 是可扩展标记语言，它和 HTML 很相似。
- 但 XML 的设计宗旨是**传输数据**，而不是显示数据（HTML 是为了在浏览器上显示数据）。
- XML 不是要替换 HTML，而是对 HTML 的补充。

# CSS



- **CSS (Cascading Style Sheets) 是层叠样式表，它是一种样式表语言，用于为 HTML 文档定义布局。**
- **CSS 与 HTML 的区别就是：HTML 用于结构化内容，而 CSS 则用于格式化结构化的内容。**

# 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))



# 6.5 电子邮件



- 6.5.1 电子邮件概述
- 6.5.2 简单邮件传送协议 SMTP
- 6.5.3 电子邮件的信息格式
- 6.5.4 邮件读取协议 POP3 和 IMAP
- 6.5.5 基于万维网的电子邮件
- 6.5.6 通用互联网邮件扩充 MIME

## 6.5.1 电子邮件概述



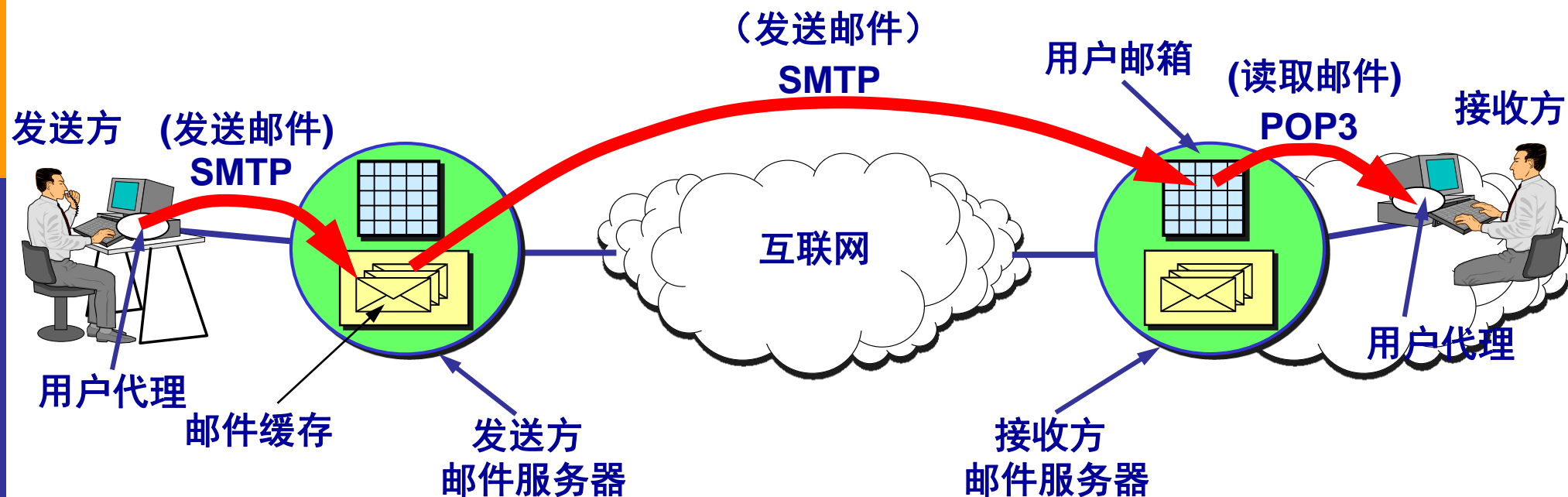
- **电子邮件 (e-mail)** 是互联网上使用得最多的和最受用户欢迎的一种应用。
- 电子邮件把邮件发送到收件人使用的邮件服务器，并放在其中的收件人邮箱中，收件人可随时上网到自己使用的邮件服务器进行读取。
- 电子邮件不仅使用方便，而且还具有传递迅速和费用低廉的优点。
- 现在电子邮件不仅可传送文字信息，而且还可附上声音和图像。

# 电子邮件的一些标准



- 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。

# 应当注意



- 一个邮件服务器既可以作为客户，也可以作为服务器。
- 例如，当邮件服务器 A 向另一个邮件服务器 B 发送邮件时，邮件服务器 A 就作为 SMTP 客户，而 B 是 SMTP 服务器。
- 当邮件服务器 A 从另一个邮件服务器 B 接收邮件时，邮件服务器 A 就作为 SMTP 服务器，而 B 是 SMTP 客户。

# 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



- 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

## 6.5.2 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。
- 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。

## 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。
- **“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



- IMAP (Internet Message Access Protocol) 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。**

# 必须注意



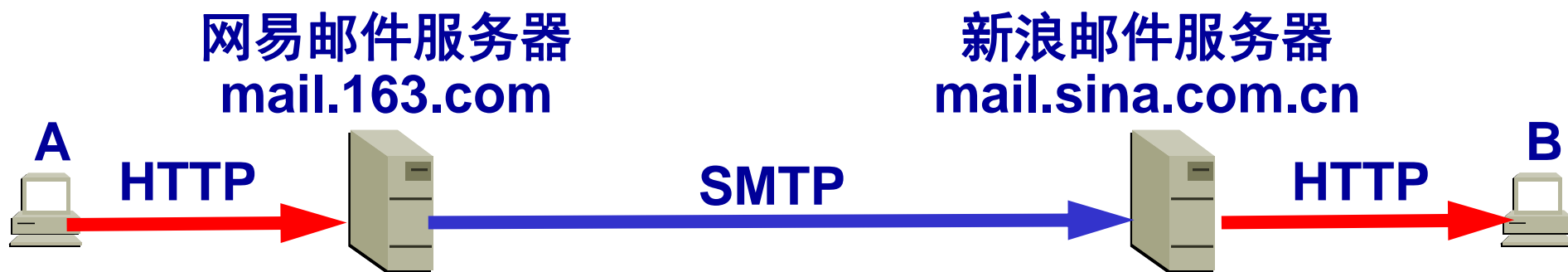
- 不要将邮件读取协议 POP 或 IMAP 与邮件传送协议 SMTP 弄混。
- 发信人的用户代理向源邮件服务器发送邮件，以及源邮件服务器向目的邮件服务器发送邮件，都是使用 SMTP 协议。
- 而 POP 协议或 IMAP 协议则是用户从目的邮件服务器上读取邮件所使用的协议。

## 6.5.5 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。

**万维网电子邮件的好处：**只要能够找到上网的计算机，打开任何一种浏览器就可以非常方便地收发电子邮件。



## 6.5.6 通用互联网邮件扩充 MIME

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。

# 1. MIME 概述



- **通用互联网邮件扩充 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 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。

# MIME Content-Type 说明中的类型及子类型



内容类型	子类型举例	说明
<b>text (文本)</b>	<b>plain, html, xml, css</b>	<b>不同格式的文本</b>
<b>image (图像)</b>	<b>gif, jpeg, tiff</b>	<b>不同格式的静止图像</b>
<b>audio (音频)</b>	<b>basic, mpeg, mp4</b>	<b>可听见的声音</b>
<b>video (视频)</b>	<b>mpeg, mp4, quicktime</b>	<b>不同格式的影片</b>
<b>model (模型)</b>	<b>vrml</b>	<b>3D模型</b>
<b>application (应用)</b>	<b>octet-stream, pdf, javascript, zip</b>	<b>不同应用程序产生的数据</b>
<b>message (报文)</b>	<b>http, rfc822</b>	<b>封装的报文</b>
<b>multipart (多部分)</b>	<b>mixed, alternative, parallel, digest</b>	<b>多种类型的组合</b>

## 6.6 动态主机配置协议 DHCP



- 为了将软件协议做成通用的和便于移植，协议软件的编写者把协议软件参数化。这就使得在很多台计算机上使用同一个经过编译的二进制代码成为可能。
- 一台计算机和另一台计算机的区别，都可通过一些不同的参数来体现。
- 在软件协议运行之前，必须给每一个参数赋值。

# 协议配置



- 在协议软件中给这些参数赋值的动作叫做**协议配置**。
- 一个软件协议在使用之前必须是已正确配置的。
- 具体的配置信息有哪些则取决于协议栈。

# 协议配置（续）



- 需要配置的项目

- (1) IP 地址

- (2) 子网掩码

- (3) 默认路由器的 IP 地址

- (4) 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。

# 动态主机配置协议 DHCP



- 互联网广泛使用的**动态主机配置协议** DHCP (Dynamic Host Configuration Protocol) 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



- 需要 IP 地址的主机在启动时就向 DHCP 服务器**广播发送**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。
- 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。DHCP 服务器的回答报文叫做提供报文（DHCPOFFER）。



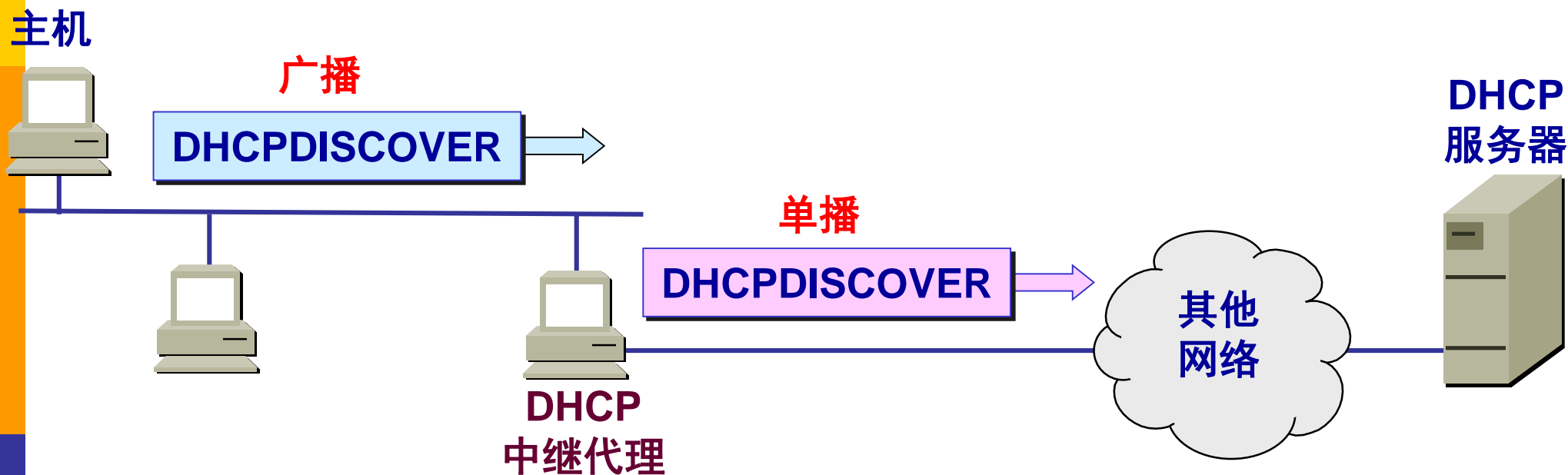
# DHCP 中继代理(relay agent)



- 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 中继代理

## 以单播方式转发发现报文



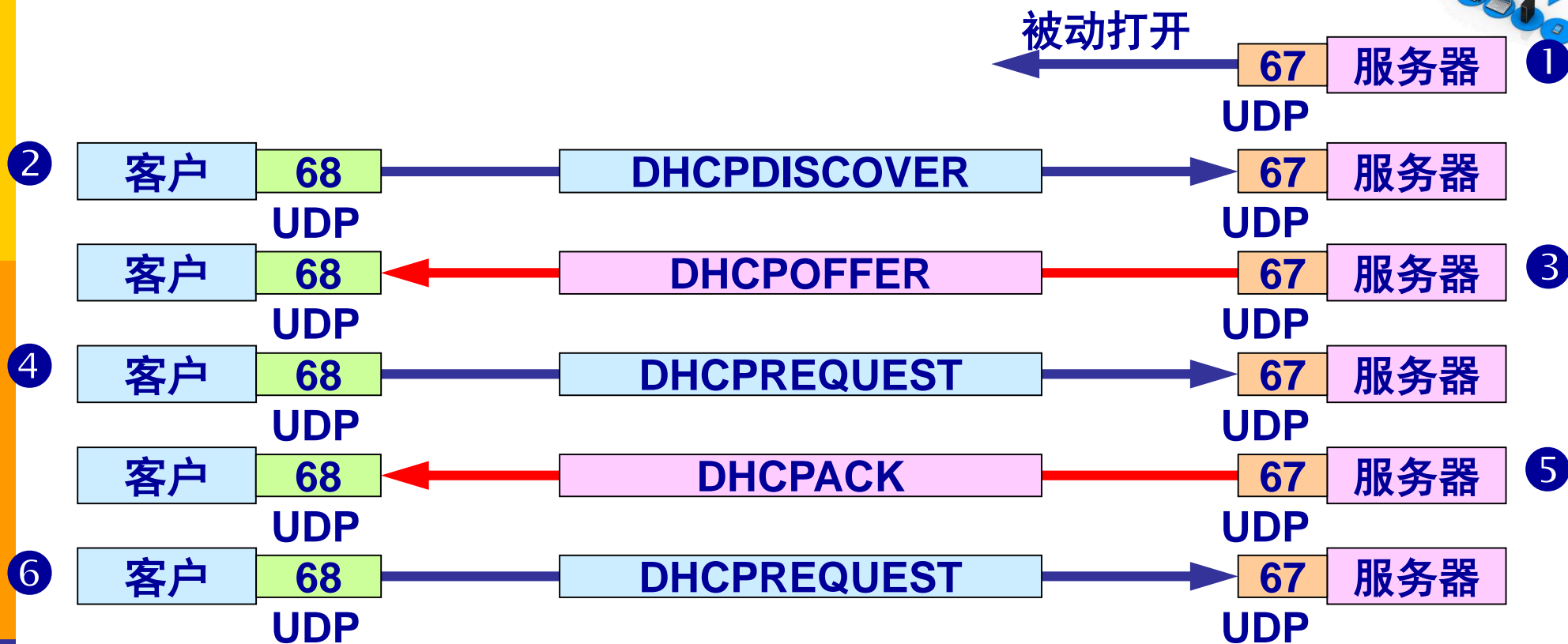
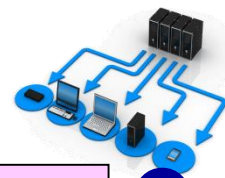
**注意：DHCP 报文只是 UDP 用户数据报中的数据。**

# 租用期 (lease period)



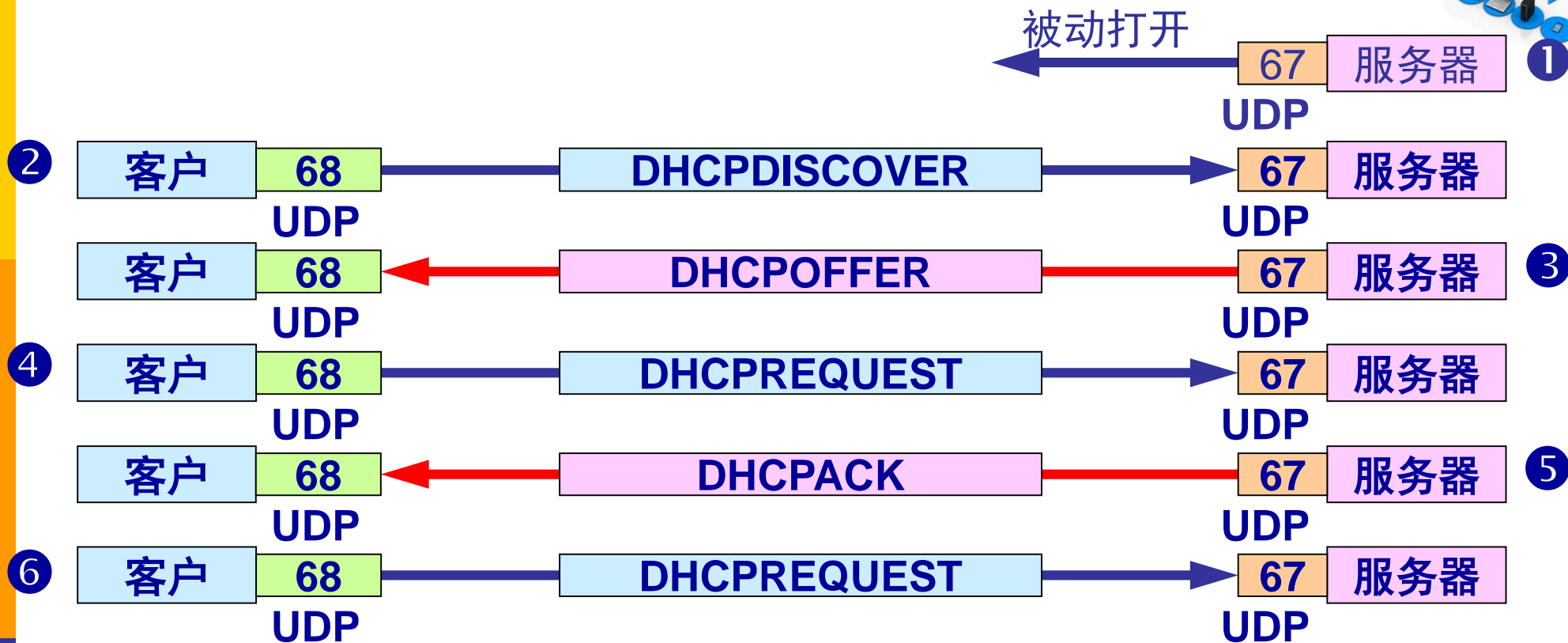
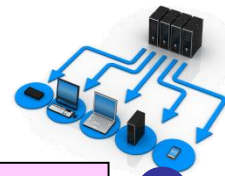
- DHCP 服务器分配给 DHCP 客户的 IP 地址的**临时的**，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。DHCP 协议称这段时间为**租用期**。
- 租用期的数值应由 DHCP 服务器自己决定。
- DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。

# DHCP 协议的工作过程



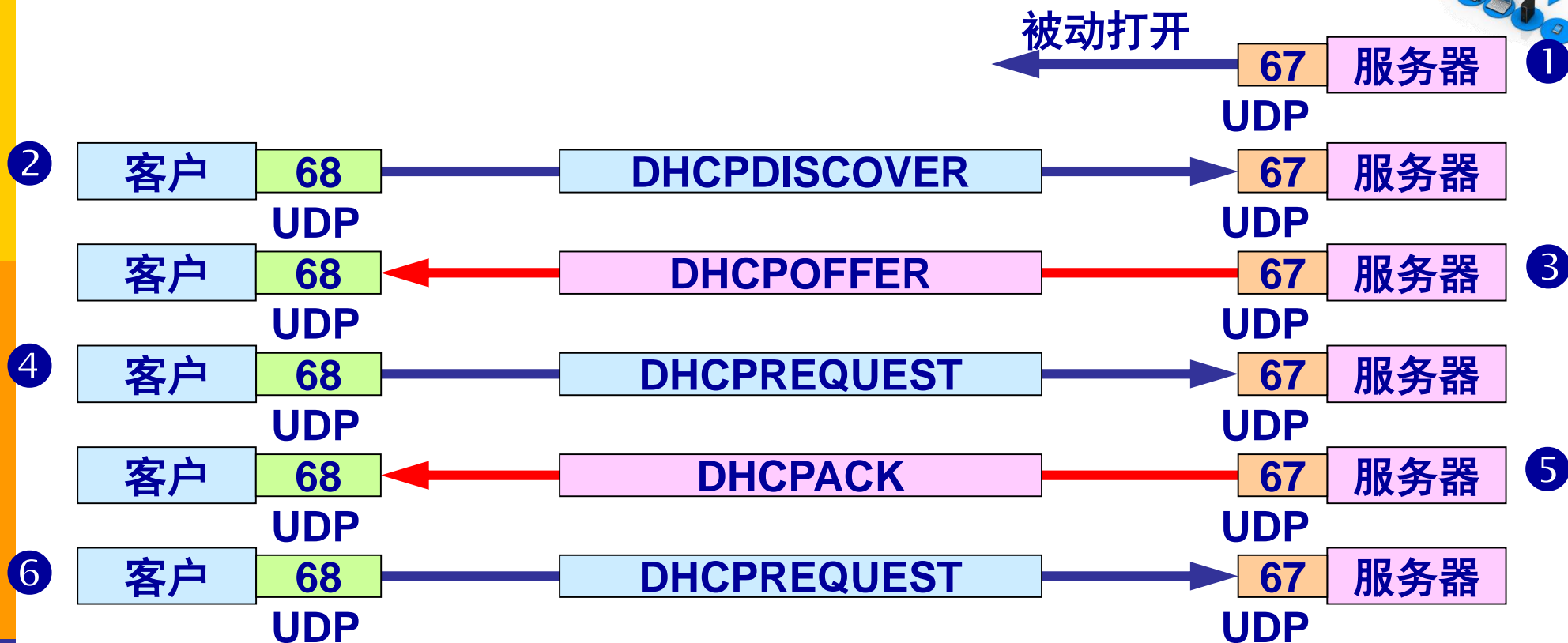
①：DHCP 服务器被动打开 UDP 端口 67，等待客户端发来的报文。

# DHCP 协议的工作过程



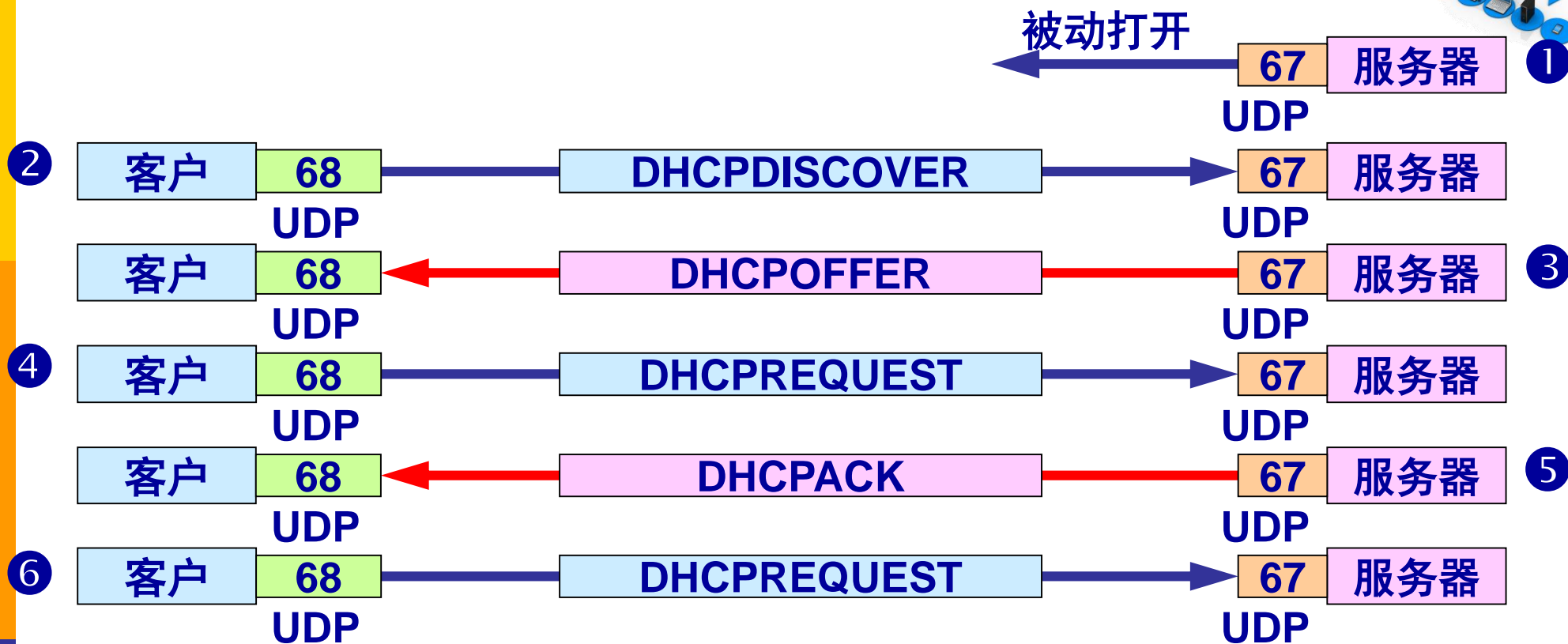
②: DHCP 客户从 UDP 端口 68 发送 DHCP 发现报文。

# DHCP 协议的工作过程



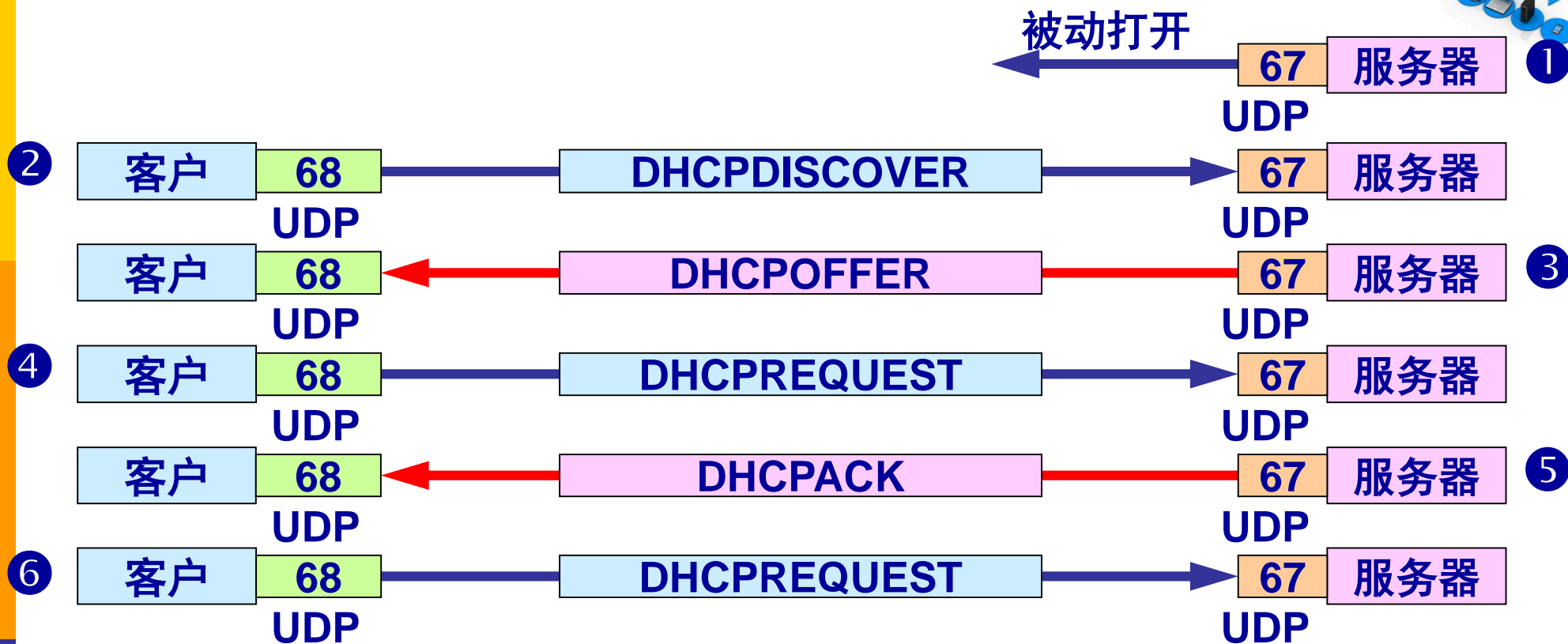
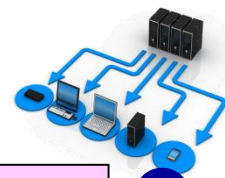
③：凡收到 DHCP 发现报文的 DHCP 服务器都发出 DHCP 提供报文，因此 DHCP 客户可能收到多个 DHCP 提供报文。

# DHCP 协议的工作过程



④：DHCP 客户从几个 DHCP 服务器中选择其中的一个，并向所选择的 DHCP 服务器发送 DHCP 请求报文。

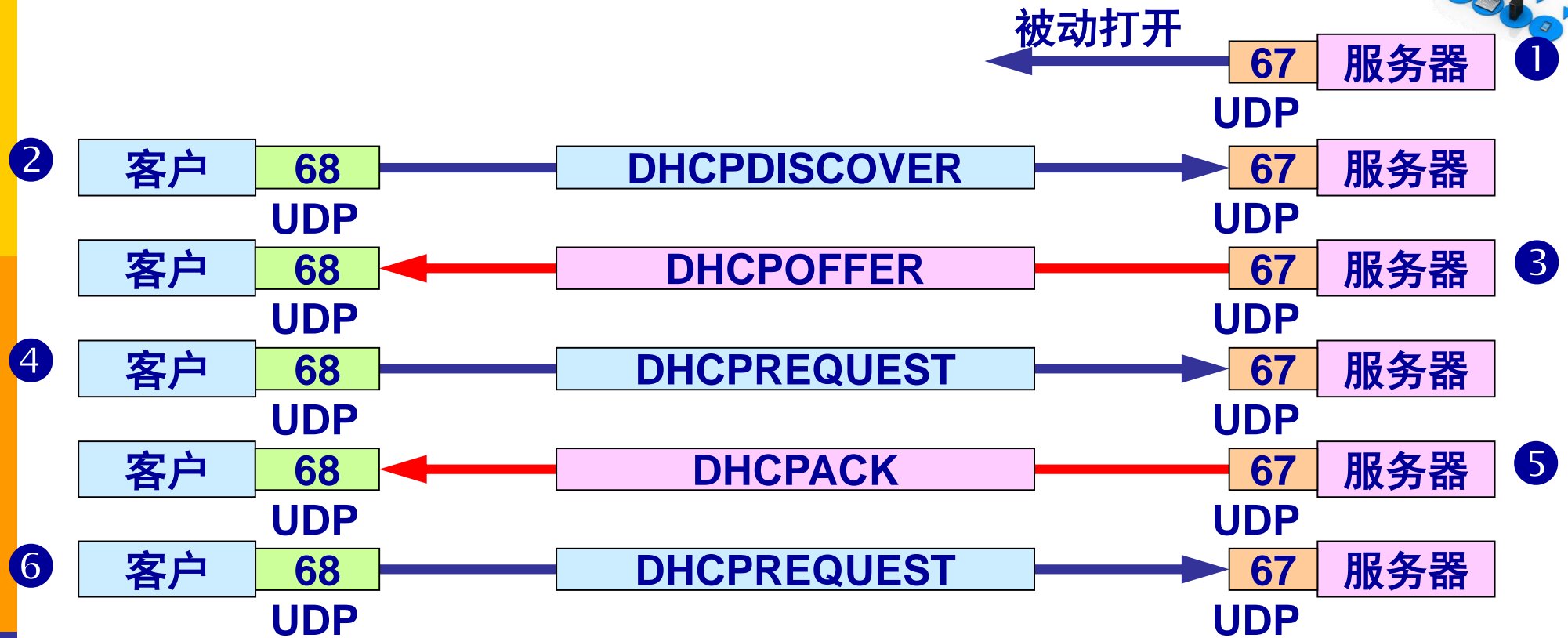
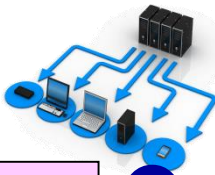
# DHCP 协议的工作过程



⑤：被选择的 DHCP 服务器发送确认报文 DHCPACK，进入已绑定状态，并可开始使用得到的临时 IP 地址了。

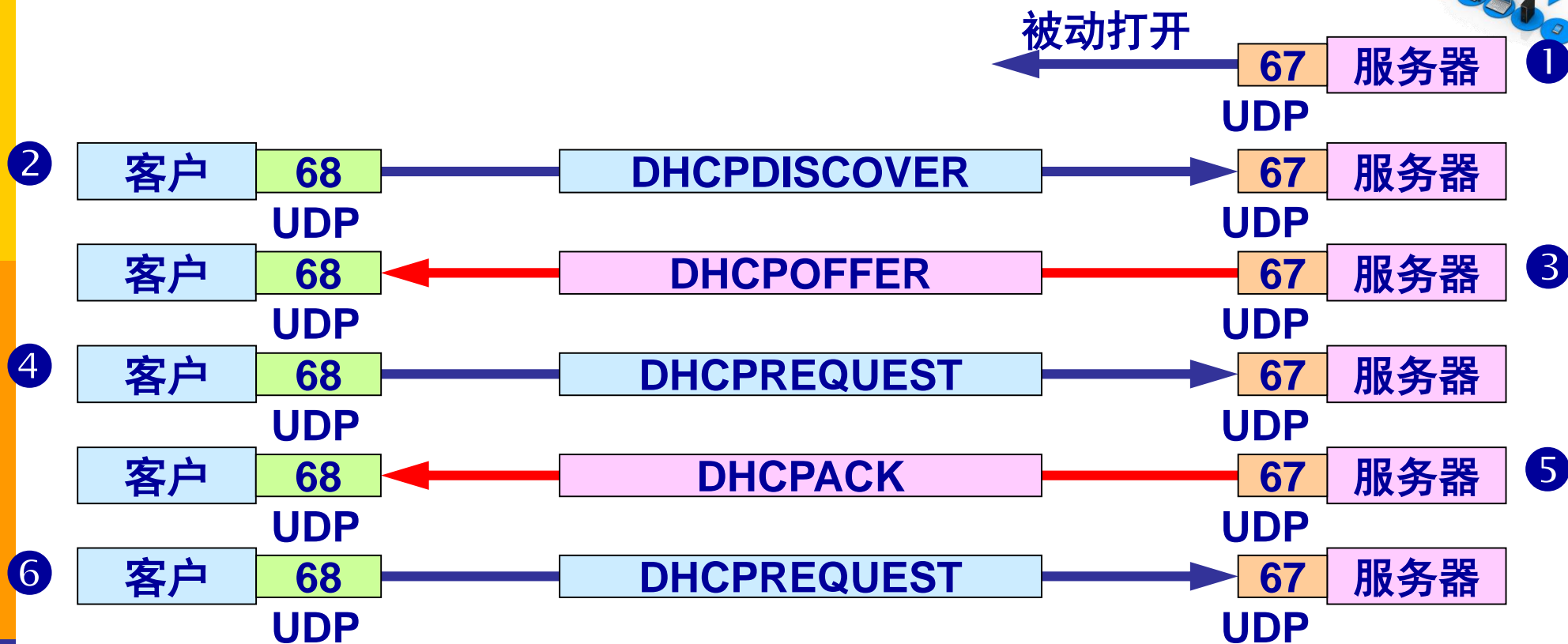


# DHCP 协议的工作过程



DHCP 客户现在要根据服务器提供的租用期  $T$  设置两个计时器  $T_1$  和  $T_2$ ，它们的超时时间分别是  $0.5T$  和  $0.875T$ 。当超时时间到就要请求更新租用期。

# DHCP 协议的工作过程

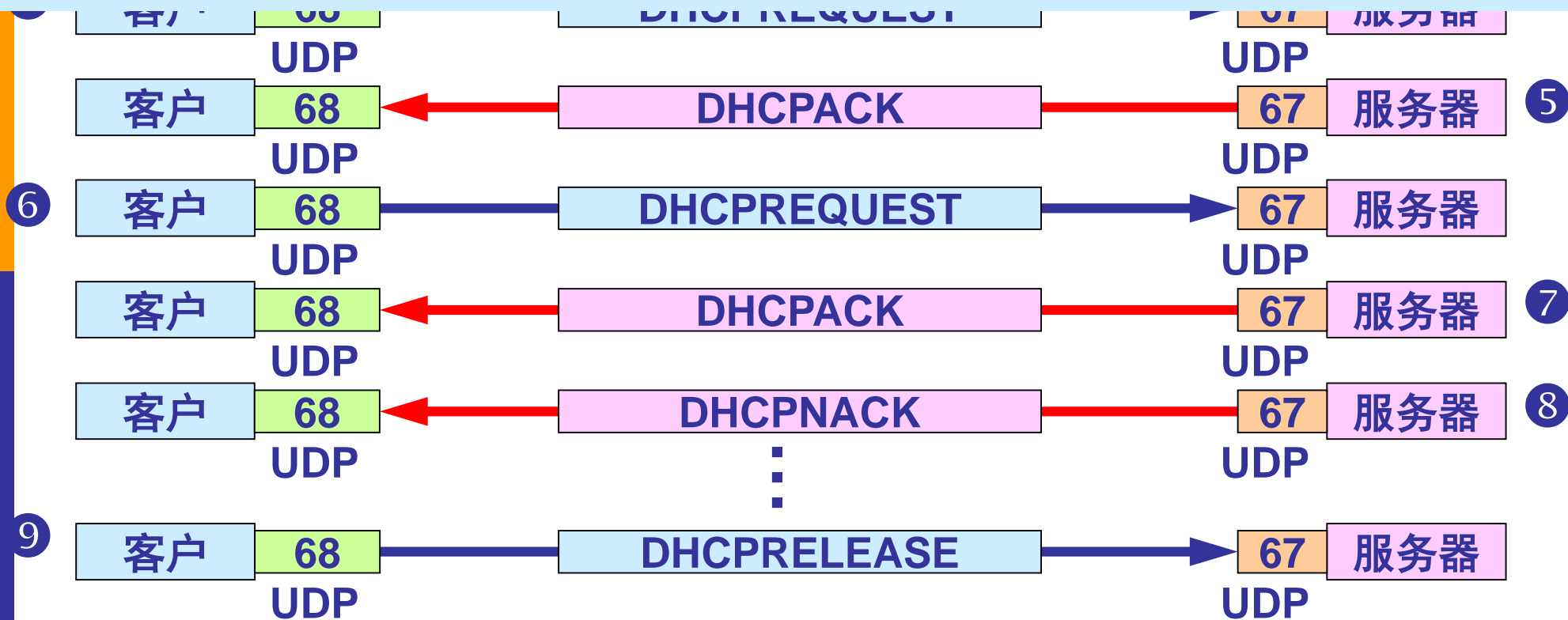


⑥：租用期过了一半（T1 时间到），DHCP 发送请求报文 DHCPREQUEST 要求更新租用期。

# DHCP 协议的工作过程



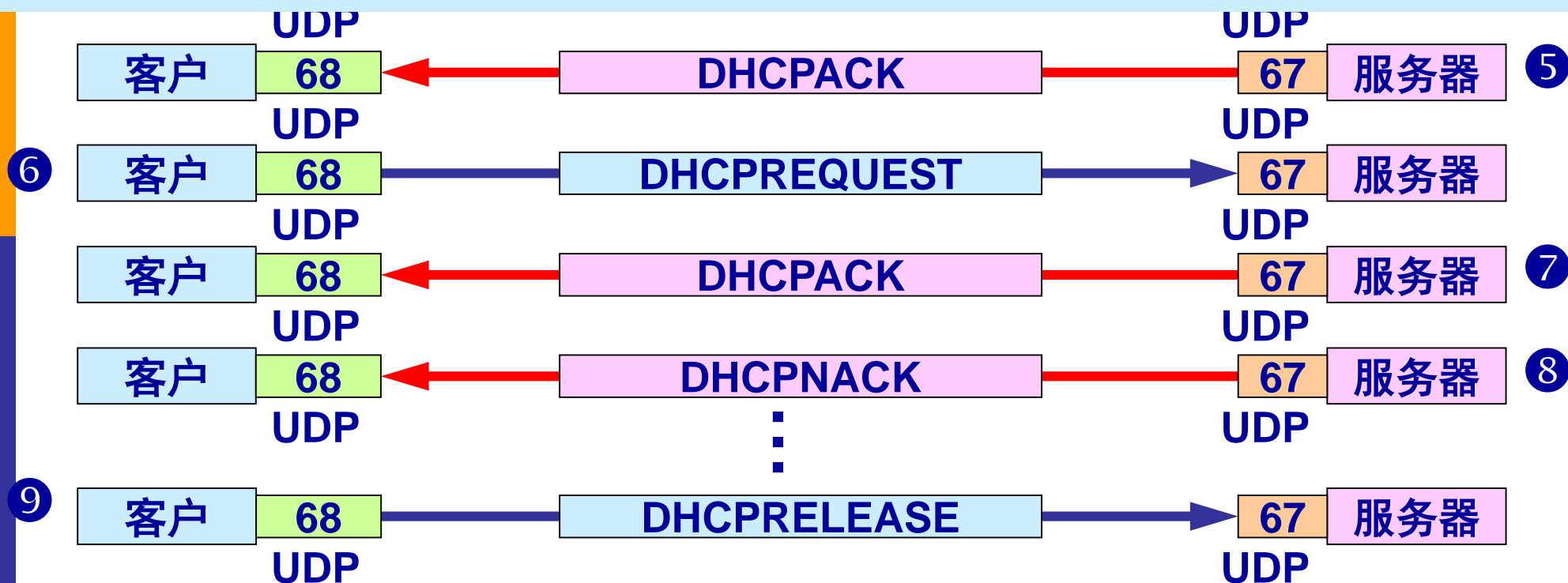
⑦：DHCP 服务器若同意，则发回确认报文 DHCPACK。DHCP 客户得到了新的租用期，重新设置计时器。



# DHCP 协议的工作过程



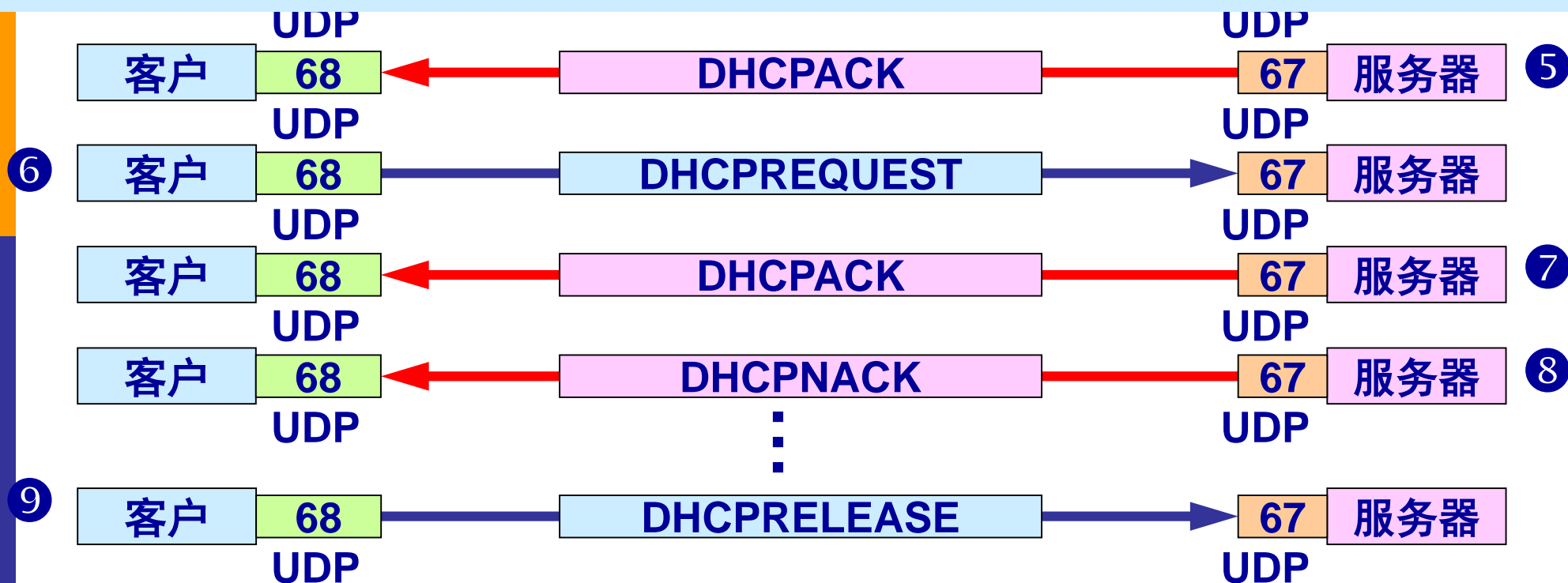
⑧：DHCP 服务器若不同意，则发回否认报文 DHCPNACK。这时 DHCP 客户必须立即停止使用原来的 IP 地址，而必须重新申请 IP 地址（回到步骤 ②）。



# DHCP 协议的工作过程



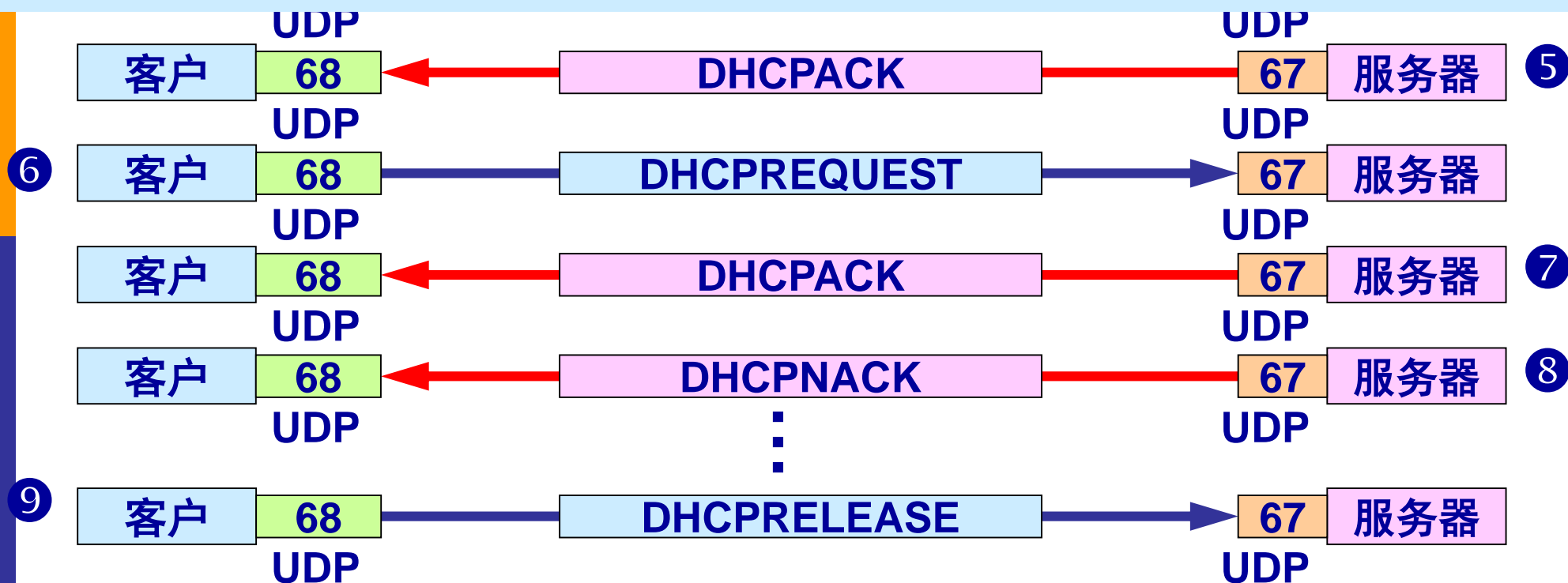
若DHCP服务器不响应步骤 ⑥ 的请求报文 DHCPREQUEST，则在租用期过了 87.5% 时，DHCP 客户必须重新发送请求报文 DHCPREQUEST（重复步骤 ⑥），然后又继续后面的步骤。



# DHCP 协议的工作过程



⑨：DHCP 客户可随时提前终止服务器所提供的租用期，这时只需向 DHCP 服务器发送释放报文 DHCPRELEASE 即可。





---

*THE END*