

计算机网络

第六章 应用层

- 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，服务端是服务提供方。

□ 网络中种种应用都可划入应用层的范畴，如今网络中最主要的就是因特网，具体要求掌握：

1. 域名系统：因特网的域名结构， 域名解析
2. 万维网 WWW
3. 电子邮件

6.1 域名系统 DNS

- 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**。

- 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。而 IP 地址是定长的 32 位二进制数字则非常便于机器进行处理。
- 域名中的“点”和点分十进制 IP 地址中的“点”并无一一对应的关系。点分十进制 IP 地址中一定是包含三个“点”，但每一个域名中“点”的数目则不一定正好是三个。

(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。

(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**。

新增加了下列的通用顶级域名

NaMI

网络与机器智能实验室

- .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)

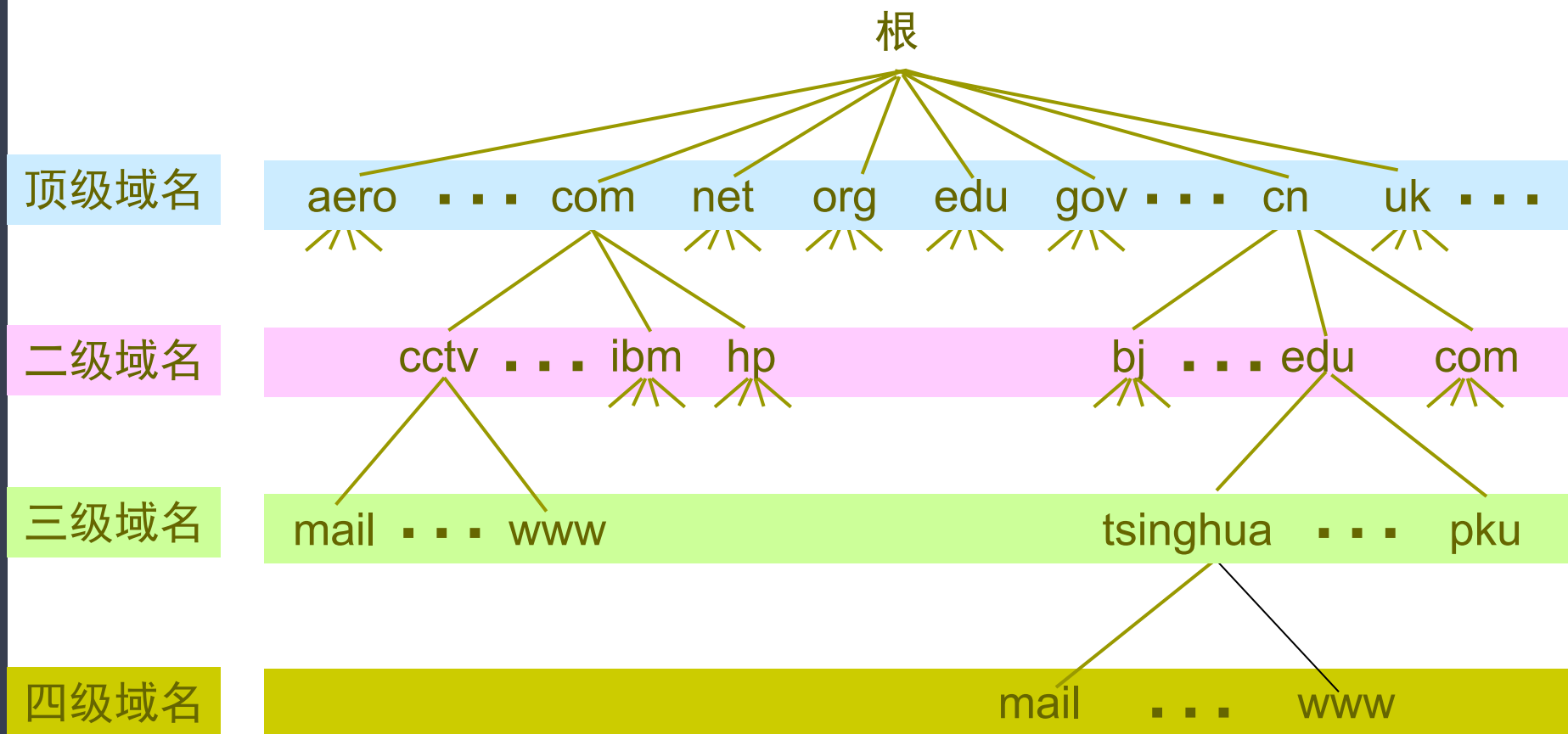
通用顶级域名

- .com 表示公司企业
- .net 表示网络服务机构
- .org 表示非赢利性组织
- .edu 表示教育机构
- .gov 表示政府部门
- .mil 表示军事部门

其他通用顶级域名例子 P238

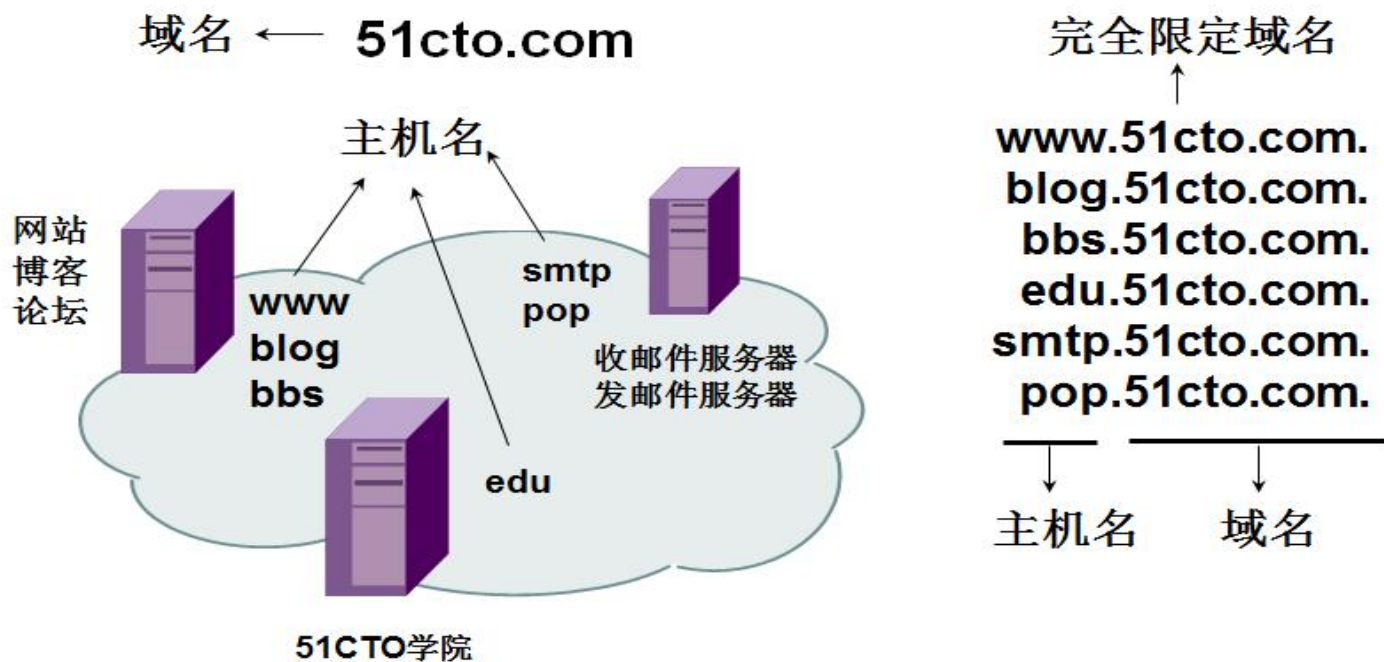
- .aero 用于航空运输企业
- .biz 用于公司和企业
- .coop 用于合作团体
- .info 适用于各种情况
- .museum 用于博物馆
- .name 用于个人

因特网的域名空间



域名的结构

- 一个域名下可以有多个主机，域名全球唯一，主机名+域名肯定也是全球唯一的，主机名+域名称为完全限定域名（FQDN）。
- FQDN是Fully Qualified Domain Name的缩写，含义是完整的域名。例如，一台机器主机名（hostname）是www，域名后缀（domain）是51cto.com，那么该主机的FQDN应该是www.51cto.com。。



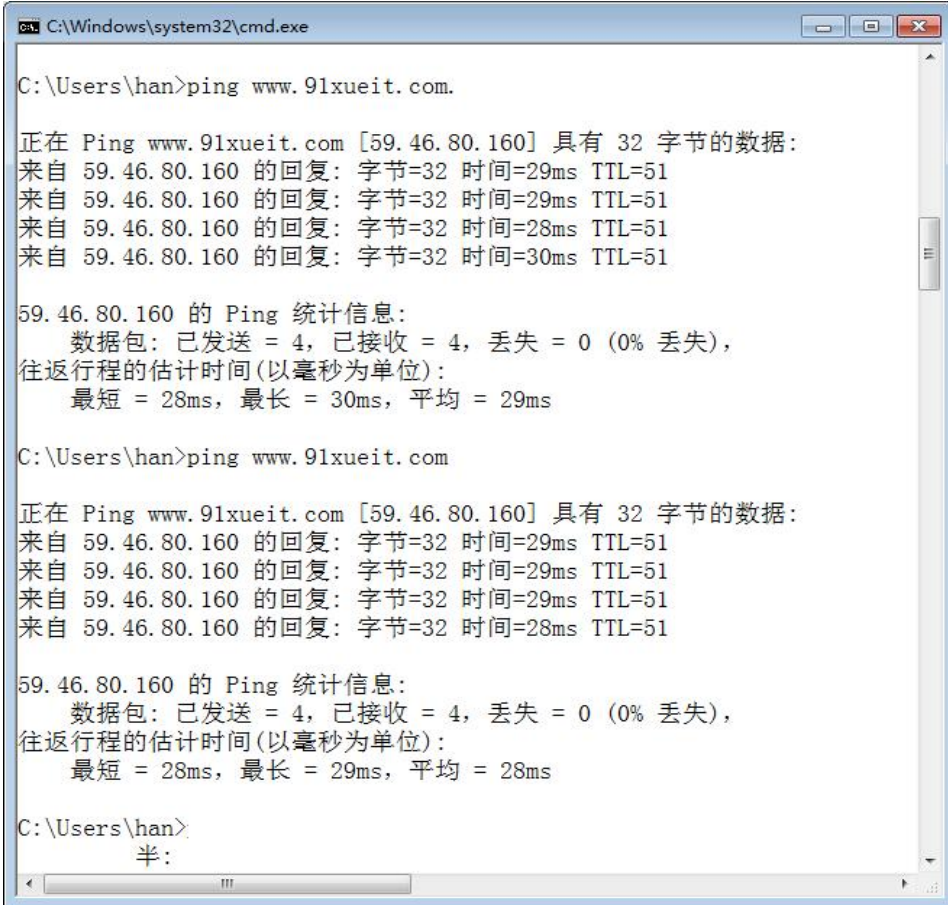
域名的结构

- 企业或个人申请了域名后，可以在该域名下添加多个主机名，也可以根据需要创建子域名，子域名下面，亦可以有多个主机名



域名的结构

- 所有域名都是以. 开始，不过我们在使用时，域名最后的 经常被省去，如图所示，在命令提示符下 ping www.91xueit.com. 和 ping www.91xueit.com 是一样的。



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

C:\Users\han>ping www.91xueit.com.

正在 Ping www.91xueit.com [59.46.80.160] 具有 32 字节的数据:
来自 59.46.80.160 的回复: 字节=32 时间=29ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=29ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=28ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=30ms TTL=51

59.46.80.160 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 28ms, 最长 = 30ms, 平均 = 29ms

C:\Users\han>ping www.91xueit.com

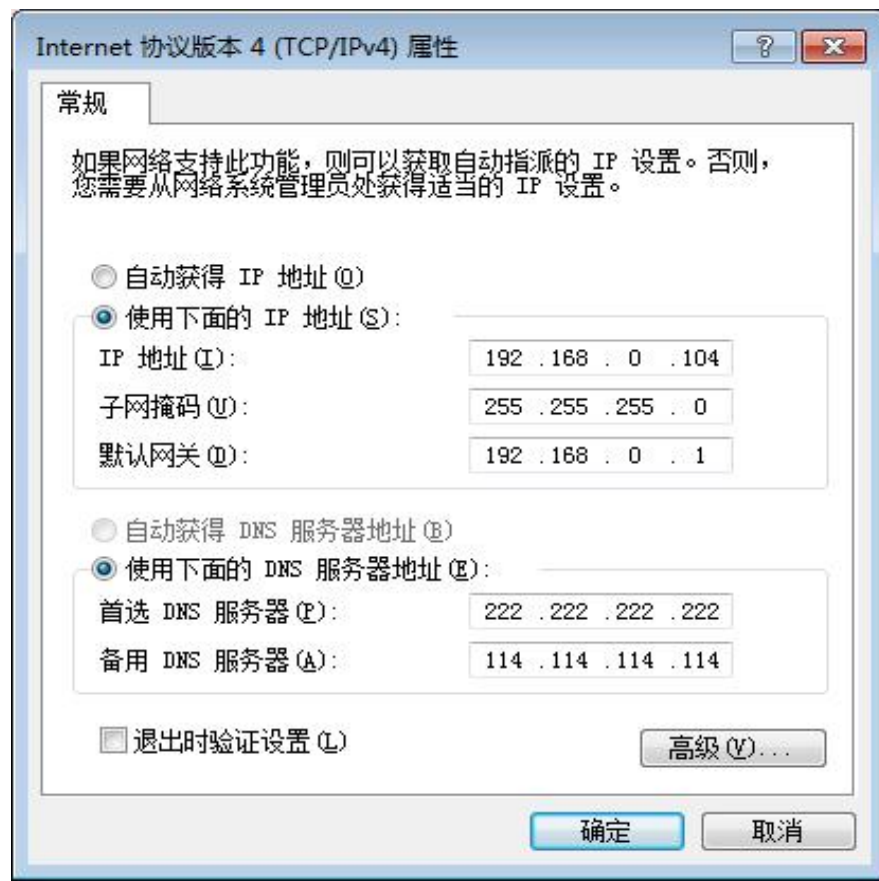
正在 Ping www.91xueit.com [59.46.80.160] 具有 32 字节的数据:
来自 59.46.80.160 的回复: 字节=32 时间=29ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=29ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=29ms TTL=51
来自 59.46.80.160 的回复: 字节=32 时间=28ms TTL=51

59.46.80.160 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 28ms, 最长 = 29ms, 平均 = 28ms

C:\Users\han>
```


Internet中的域名服务器

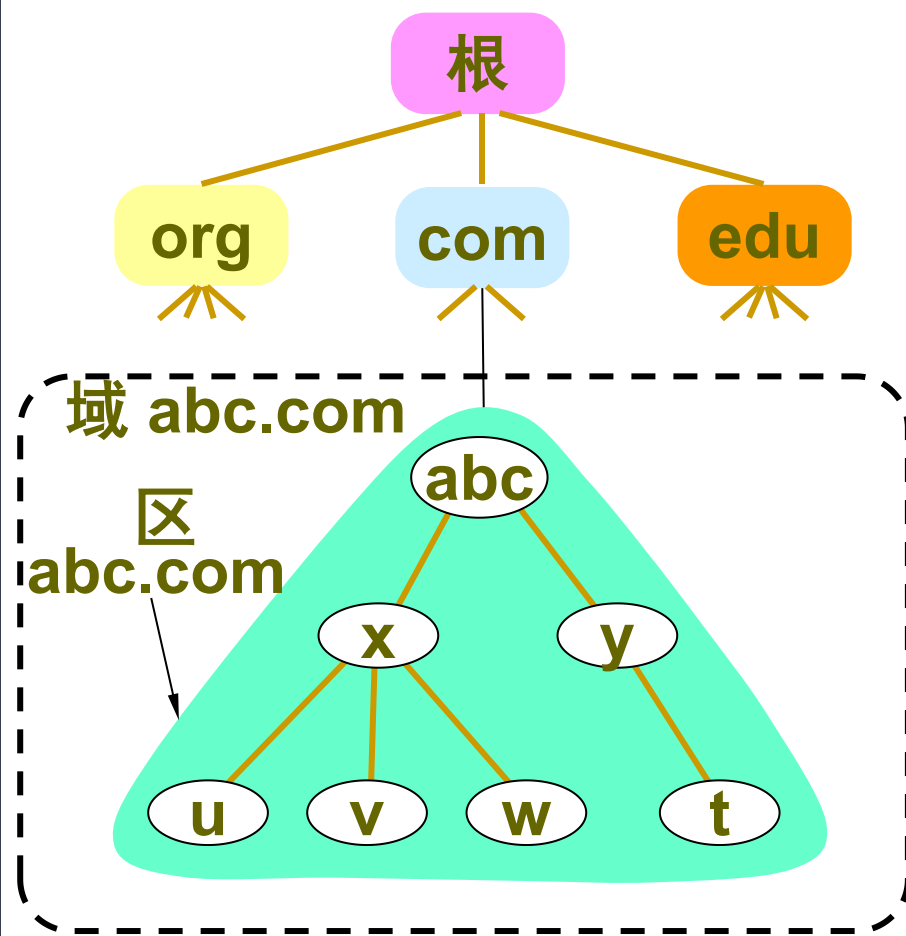
- 当我们通过域名访问网站或点击网页中的超链接跳转到其他网站，计算机需要将域名解析成IP地址才能访问这些网站。DNS服务器负责域名解析，因此你必须配置计算机使用哪些DNS服务器进行域名解析。



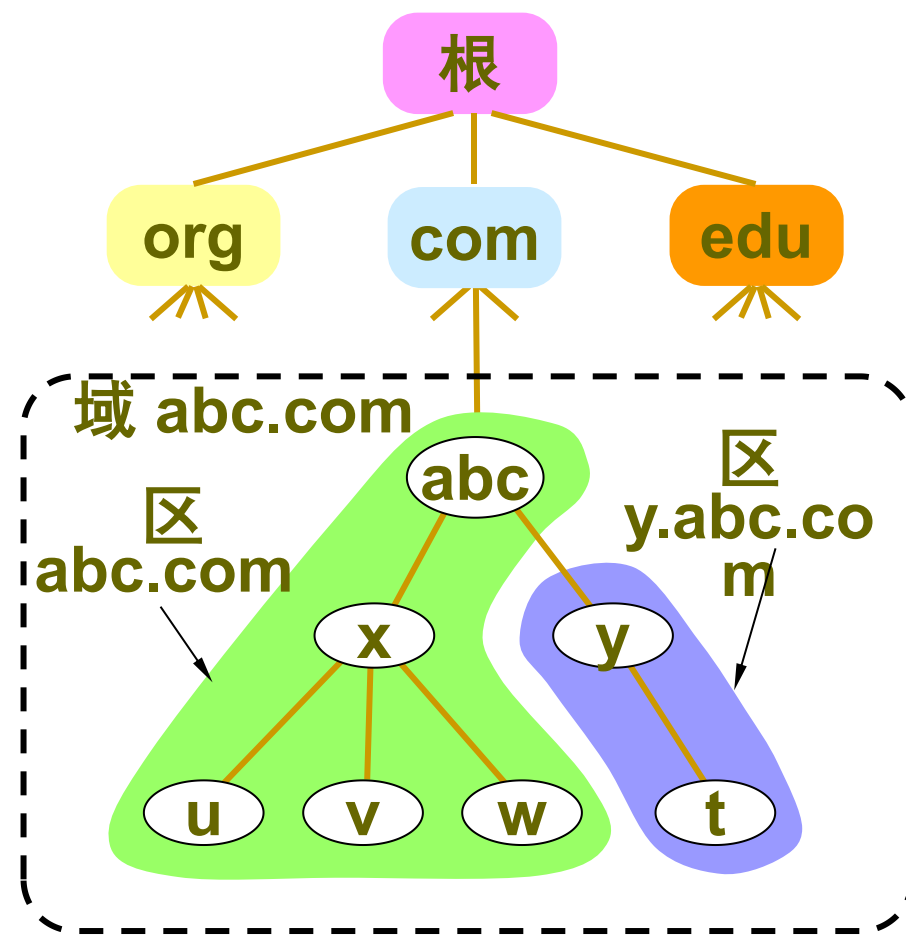
6.1.3 域名服务器

- 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。

区的不同划分方法举例

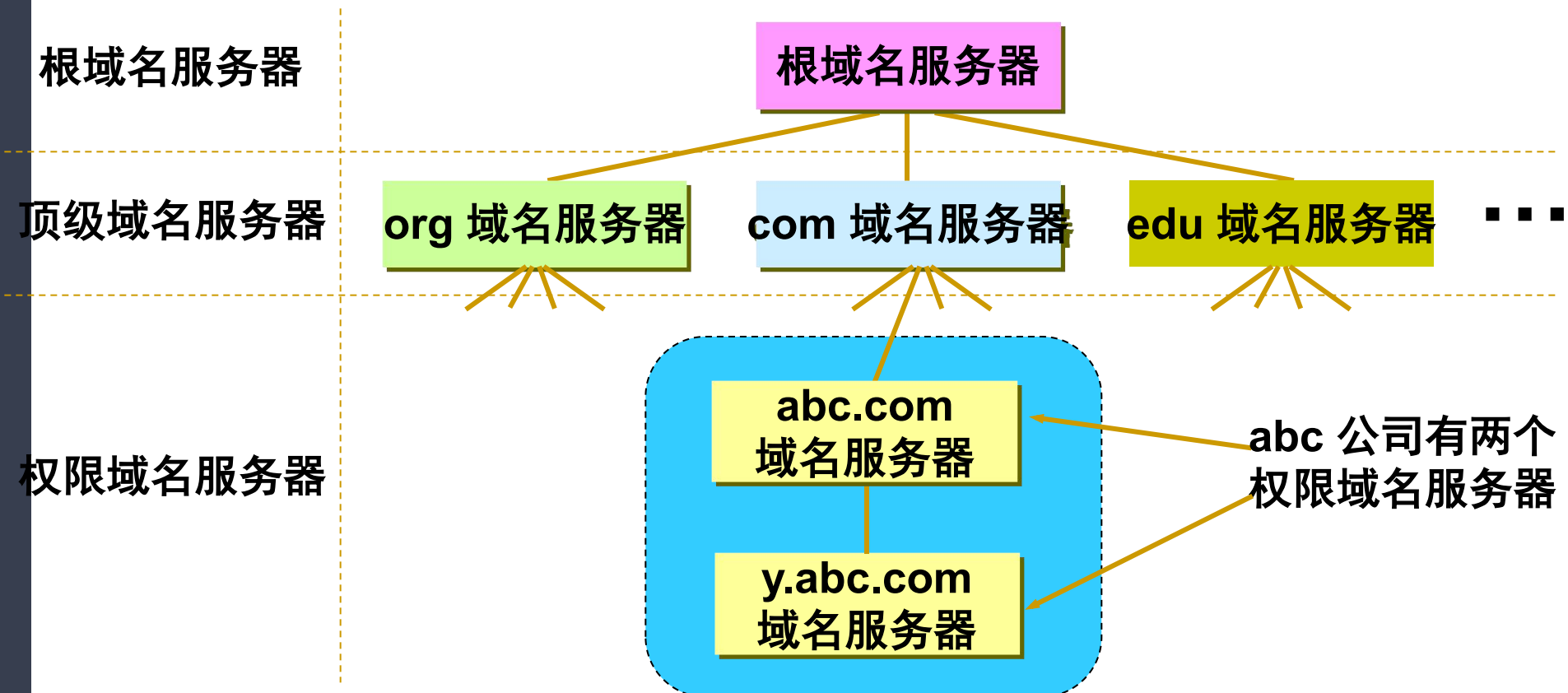


(a) 区 = 域



(b) 区 < 域

树状结构的 DNS 域名服务器



因特网上的**DNS**域名服务器也是按照层次安排的。每一个域名服务器都只对域名体系中的一部分进行管辖。根据域名服务器所起的作用，可以把域名服务器划分为以下四种类型：

- 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

- 根域名服务器是**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。
- 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）。

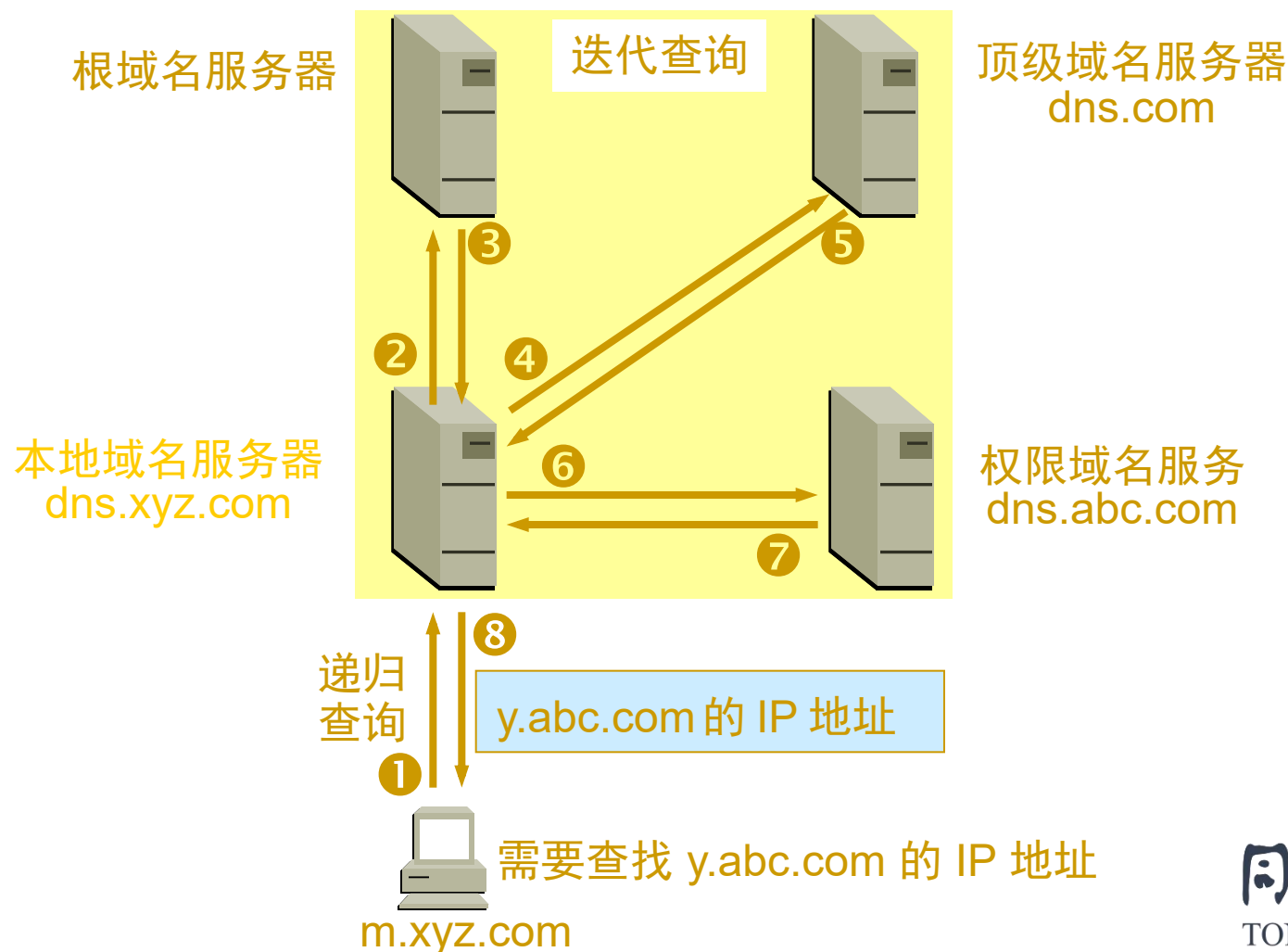
- 这就是前面已经讲过的负责一个区的域名服务器。
- 当一个权限域名服务器还不能给出最后的查询回答时，就会告诉发出查询请求的 DNS 客户，下一步应当找哪一个权限域名服务器。

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

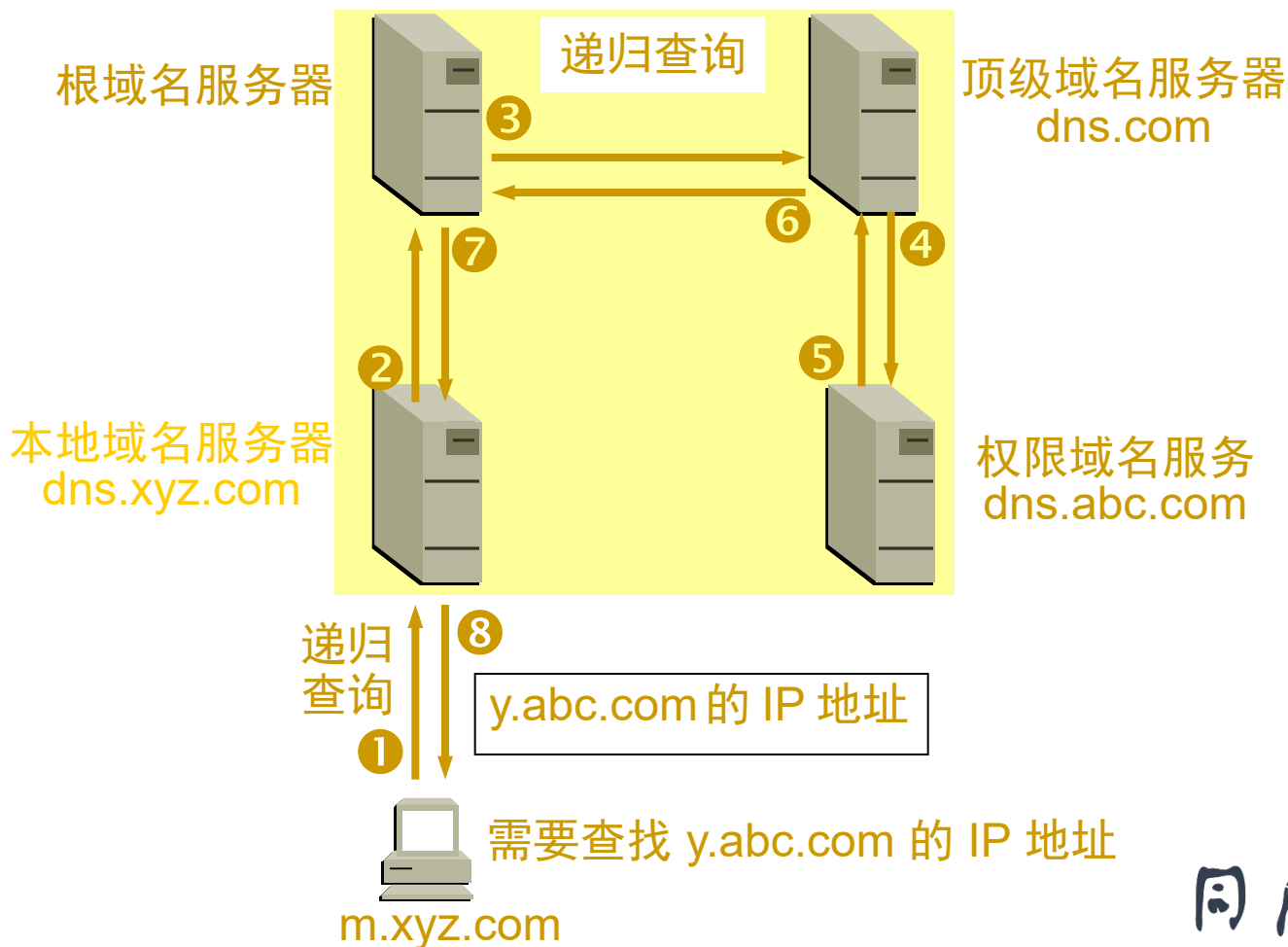
- DNS 域名服务器都把数据复制到几个域名服务器来保存，其中的一个是**主域名服务器**，其他的是**辅助域名服务器**。
- 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断。
- 主域名服务器定期把数据复制到辅助域名服务器中，而更改数据只能在主域名服务器中进行。这样就保证了数据的一致性。

- **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）



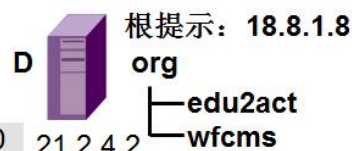
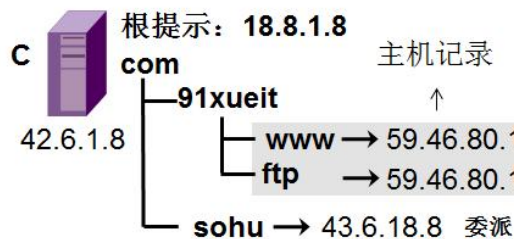
- **每个域名服务器都维护一个高速缓存**，存放最近用过的名字以及从何处获得名字映射信息的记录。
- 可大大减轻根域名服务器的负荷，使因特网上的DNS 查询请求和回答报文的数量大为减少。
- 为保持高速缓存中的内容正确，域名服务器应为每项内容设置计时器，并处理超过合理时间的项（例如，每个项目只存放两天）。
- 当权限域名服务器回答一个查询请求时，在响应中都指明绑定有效存在的时间值。增加此时间值可减少网络开销，而减少此时间值可提高域名转换的准确性。

域名解析过程

根域名服务器



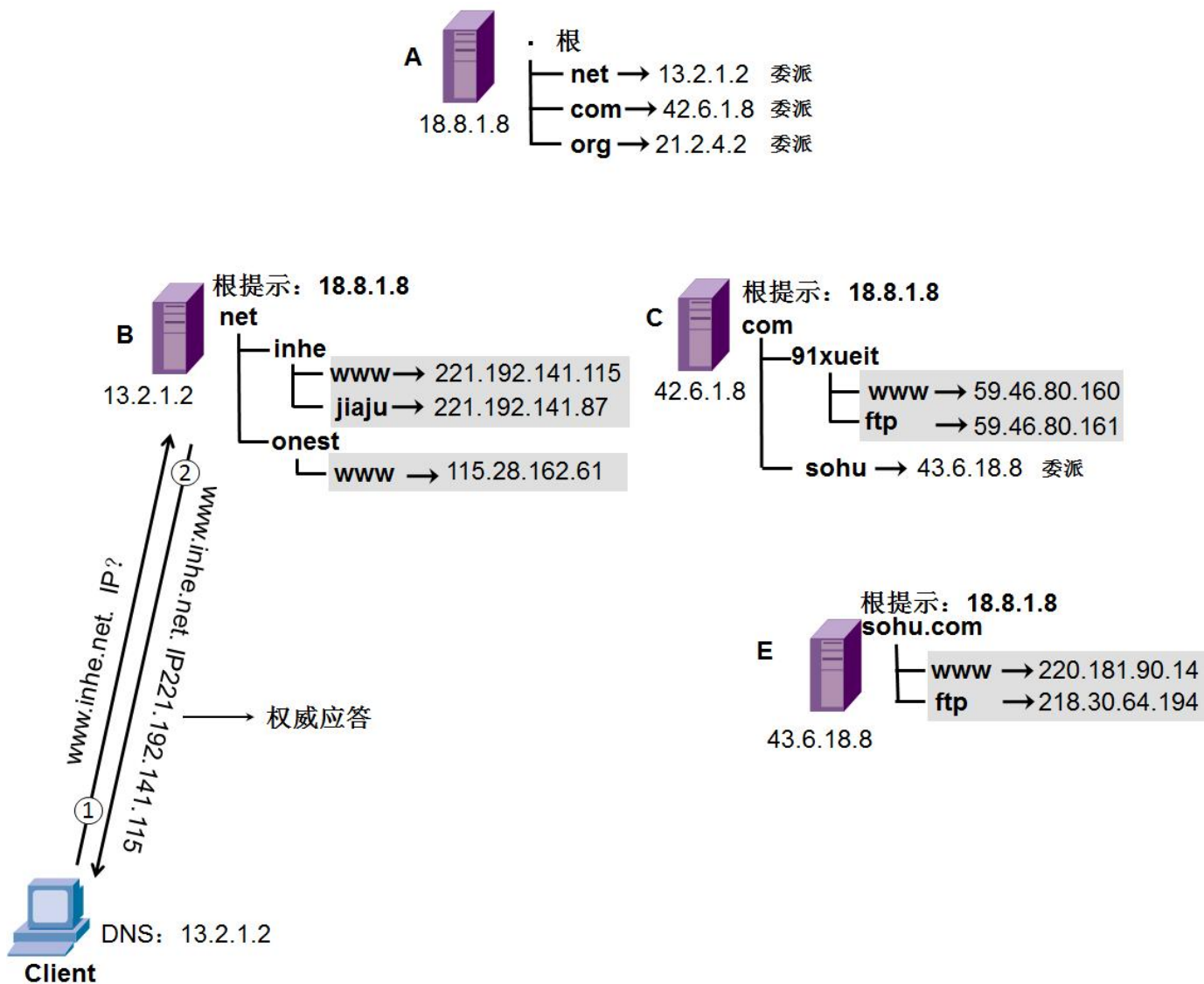
顶级域名服务器



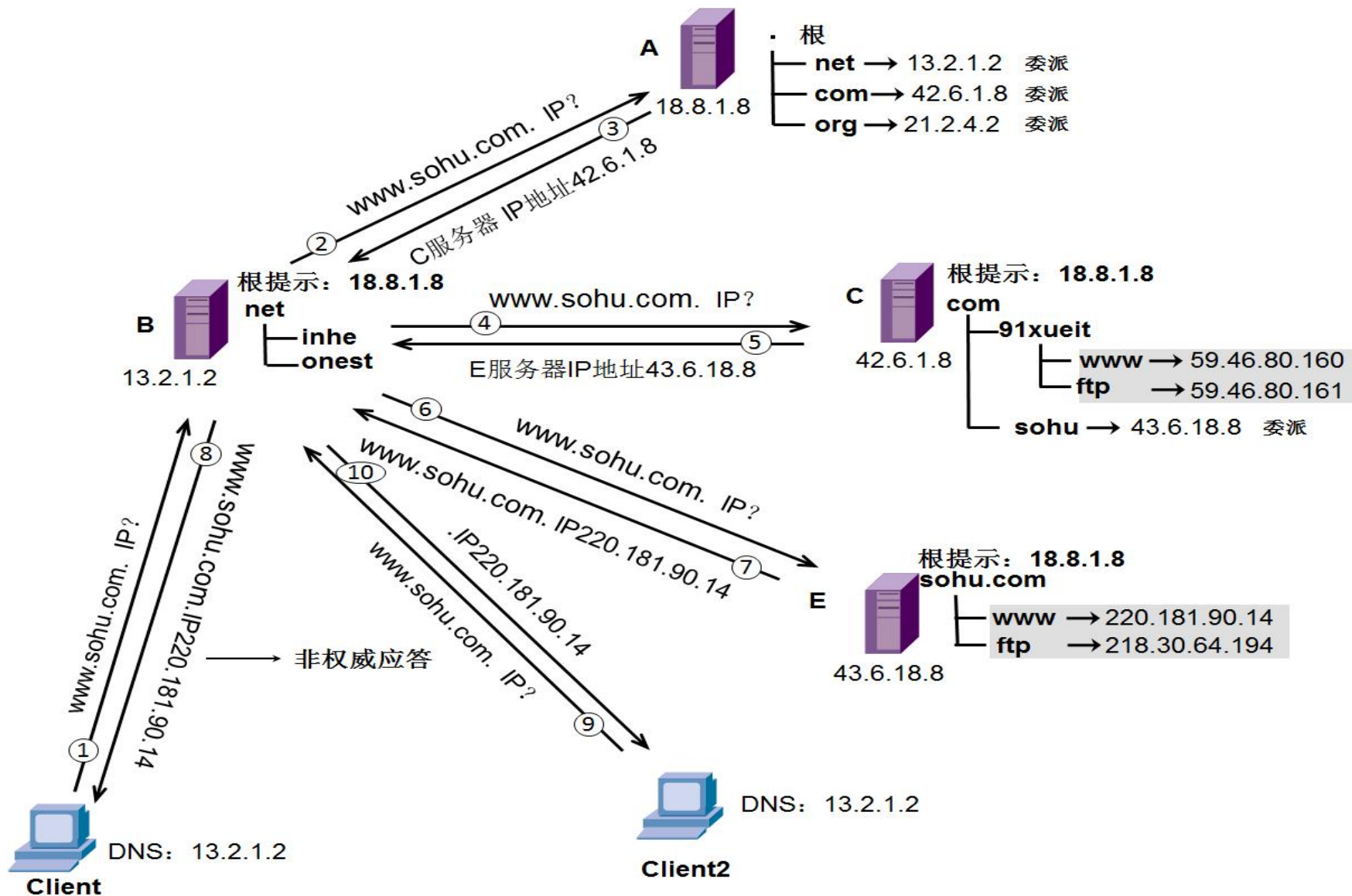
三级域名服务器



域名解析过程



域名解析过程

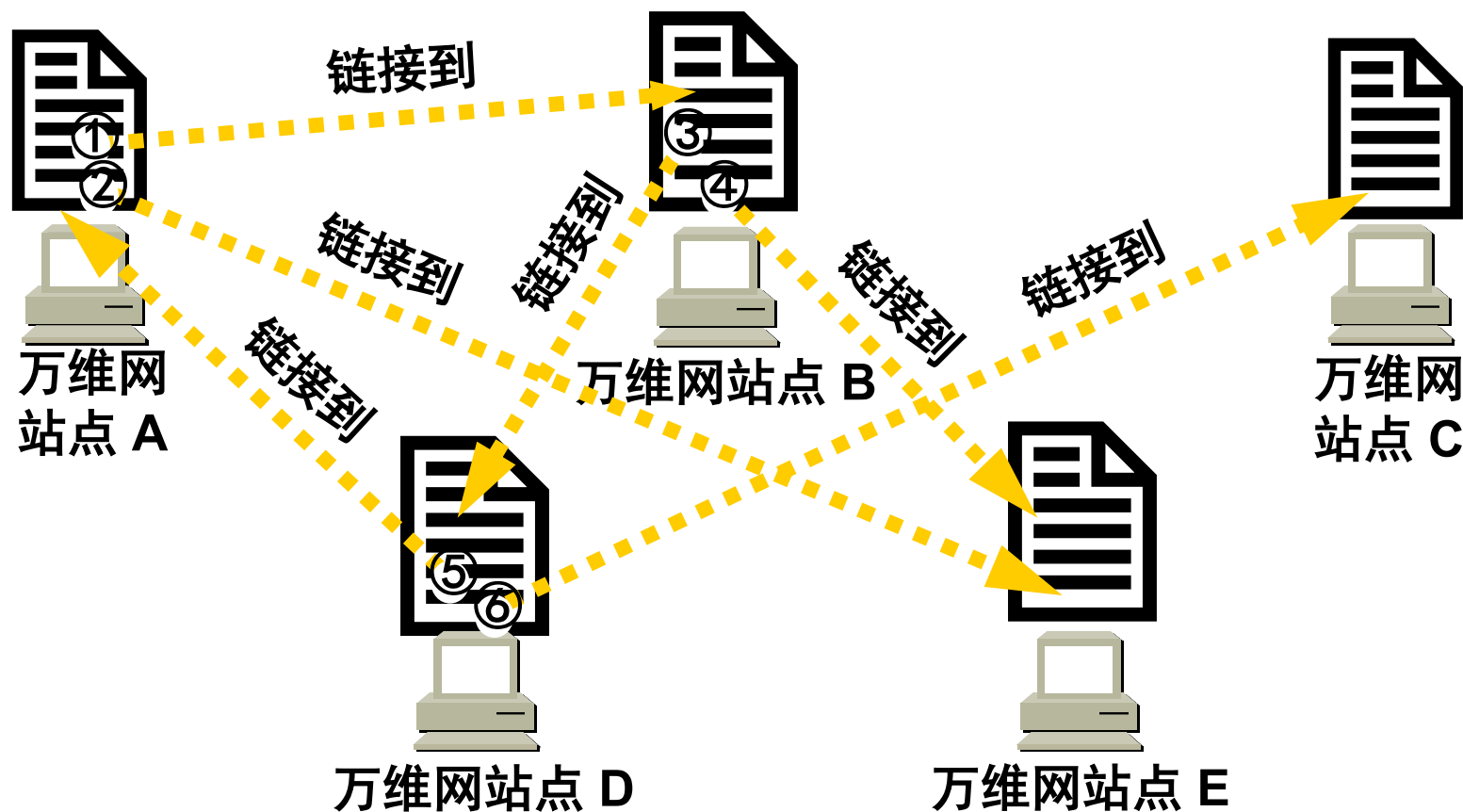


申请域名举例

- 有众多的网站都可以申请域名:
- 中国互联网络信息中心 <http://www.cnnic.cn>
- 中国万网 <http://www.net.cn>
- 二级域名申请 <http://www.9126.com>

- **万维网** WWW (World Wide Web)并非某种特殊的计算机网络。
- 万维网是个大规模、联机式**信息储藏所**。
- 万维网用链接方法能非常方便地从因特网上的一个站点访问另一站点，主动按需获取信息。这种访问方式称为“**链接**”。

万维网提供分布式服务



- 万维网是**分布式超媒体**(hypermedia)系统，它是**超文本**(hypertext)系统的扩充。
- 超文本是包含指向其他文档的链接的文本。
- 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。
- 流媒体是边接收边运行多媒体信息数据的一种方式

万维网的工作方式

- 万维网以**客户-服务器**方式工作。
- 浏览器/服务器 (Browse/Server B/S)
- **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**，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**。在客户主窗口上显示出的万维网文档称为**页面**(page)。

(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) 用何协议实现万维网各种超链的链接？

- ▣ 在万维网客户程序与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) 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）。

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，在 URL 中的字符对大写或小写没有要求。
- URL 的一般形式是：

<URL的访问方式>://<主机>:<端口>/<路径>

ftp —— 文件传送协议 FTP

http —— 超文本传送协议 HTTP

.....

URL 的一般形式

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

<URL的访问方式>://<主机>:<端口>/<路径>

<主机> 是存放资源的主机
在因特网中的域名

URL 的一般形式

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

<URL的访问方式>://<主机>:<端口>/<路径>

有时可省略

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。

对于万维网的网点的访问要使用HTTP协议。

□ 使用 HTTP 的 URL 的一般形式

http://<主机>:<端口>/<路径>

这表示使用 HTTP 协议

□ 使用 HTTP 的 URL 的一般形式

□ `http://<主机>:<端口>/<路径>`

冒号和两个斜线是规定的格式

□ 使用 HTTP 的 URL 的一般形式

□ `http://<主机>:<端口>/<路径>`

这里写主机的域名

□ 使用 HTTP 的 URL 的一般形式

□ `http://<主机>:<端口>/<路径>`

HTTP 的默认端口号是 80，通常可省略

□ 使用 HTTP 的 URL 的一般形式

□ `http://<主机>:<端口>/<路径>`

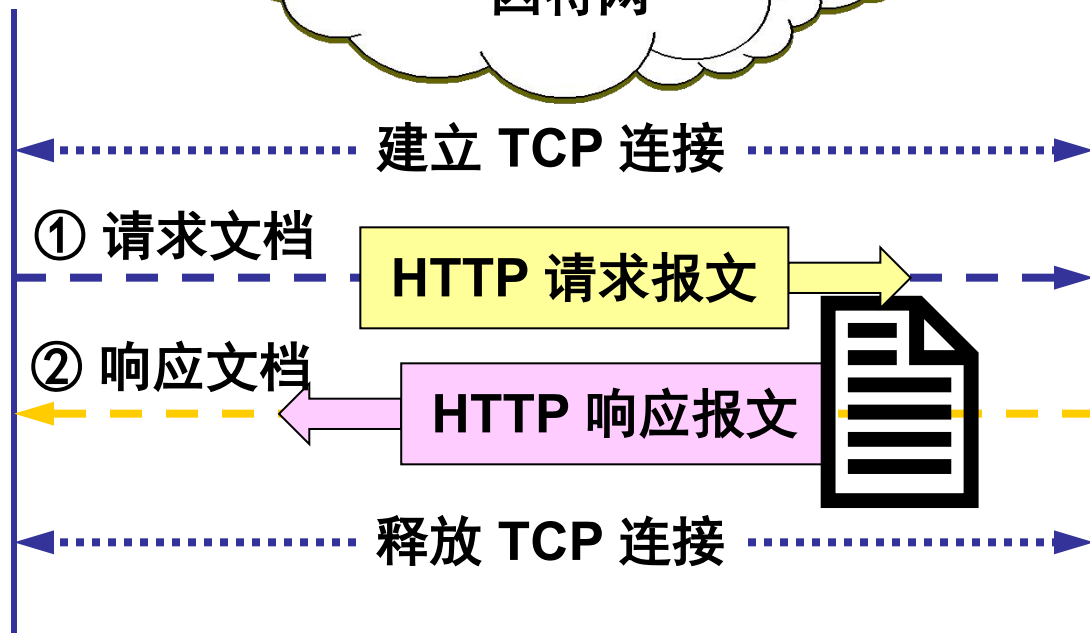
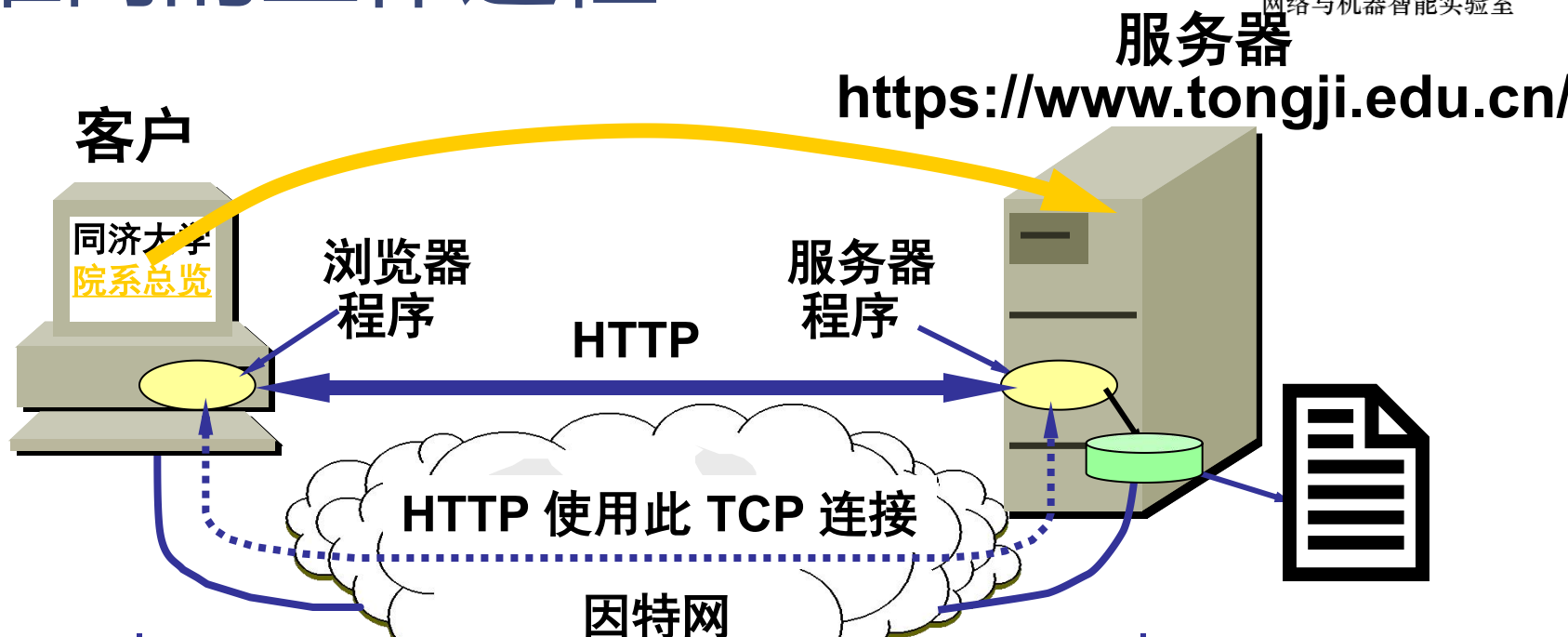
若再省略文件的<路径>项，则 URL 就指到因特网上的某个主页(home page)。

- 为了使超文本的链接能够高效率地完成，需要用 HTTP 协议传送一切必须信息。
- 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

NaMI

网络与机器智能实验室



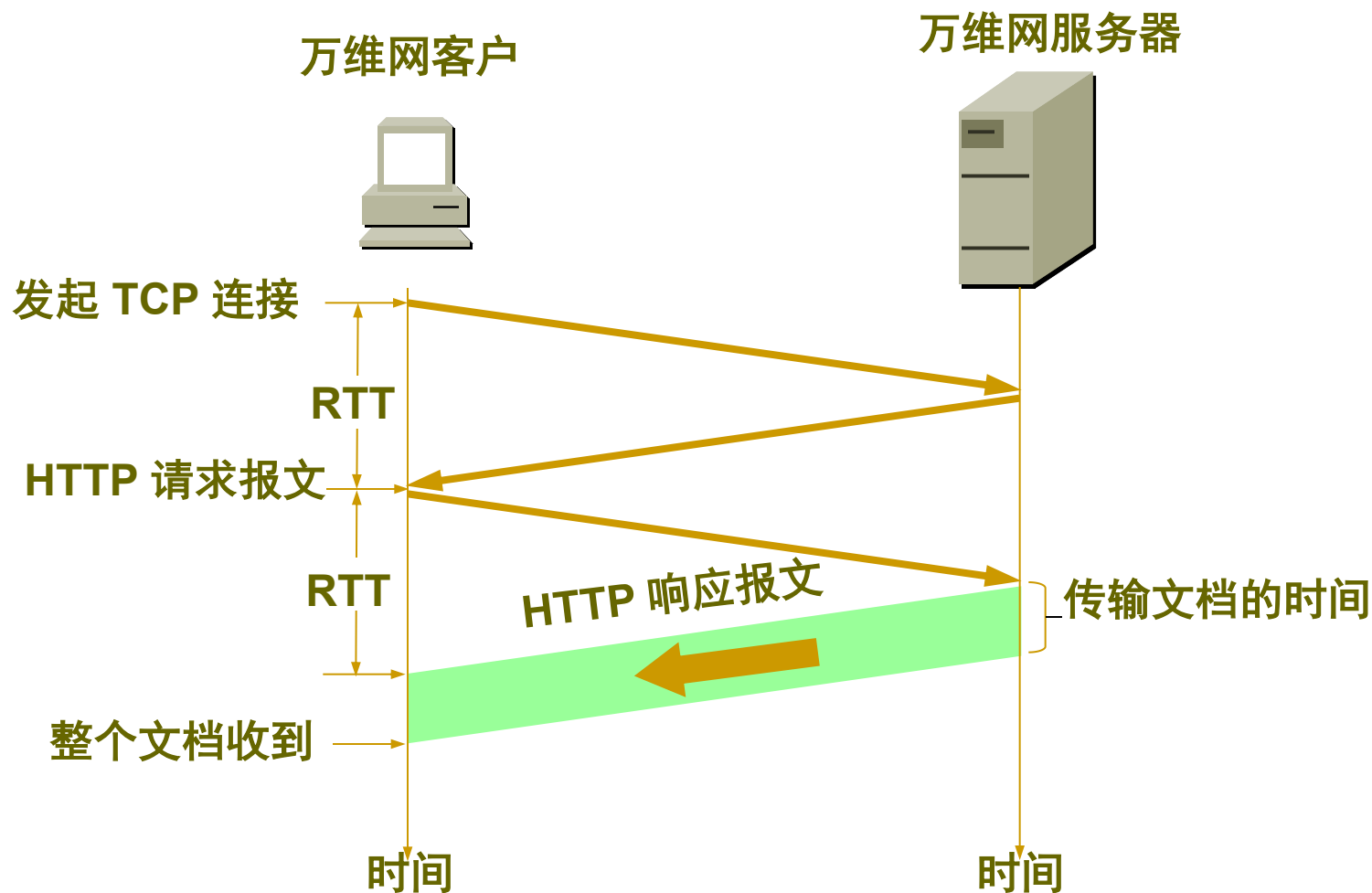
用户点击鼠标后所发生的事件

- (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.tongji.edu.cn` 的 IP 地址。
- (3) 域名系统 DNS 解析出同济大学服务器的 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/yxjg1/yxzl.htm`
- (6) 服务器响应，把文件 `yxjg1/yxzl.htm` 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示 “同济大学院系总览”

HTTP 的主要特点

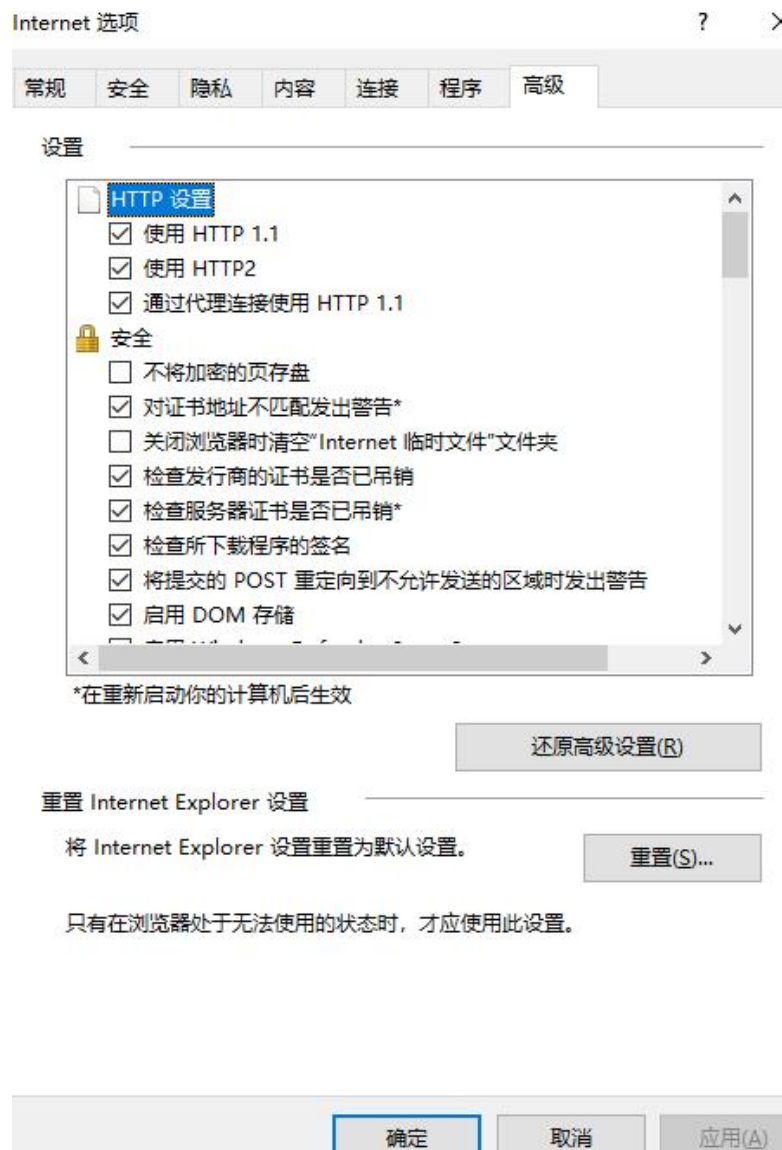
- HTTP 是面向事务的客户服务器协议。
- HTTP 1.0 协议是**无状态的**(stateless)。
- HTTP 协议本身也是无连接的，虽然它使用了面向连接的 TCP 向上提供的服务。

请求一个万维网文档所需的时间



万维网客户把HTTP请求报文作为三次握手的第三个报文的数据发给服务器

- 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。

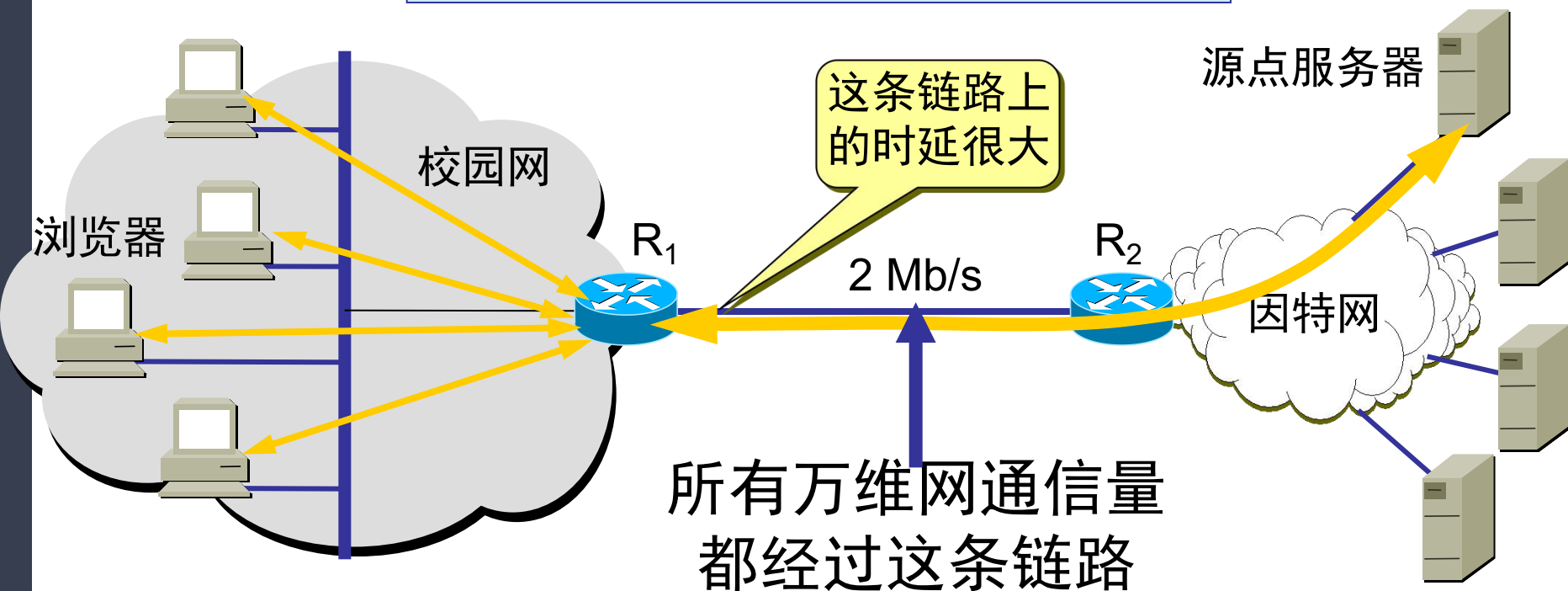


持续连接的两种工作方式

- **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。

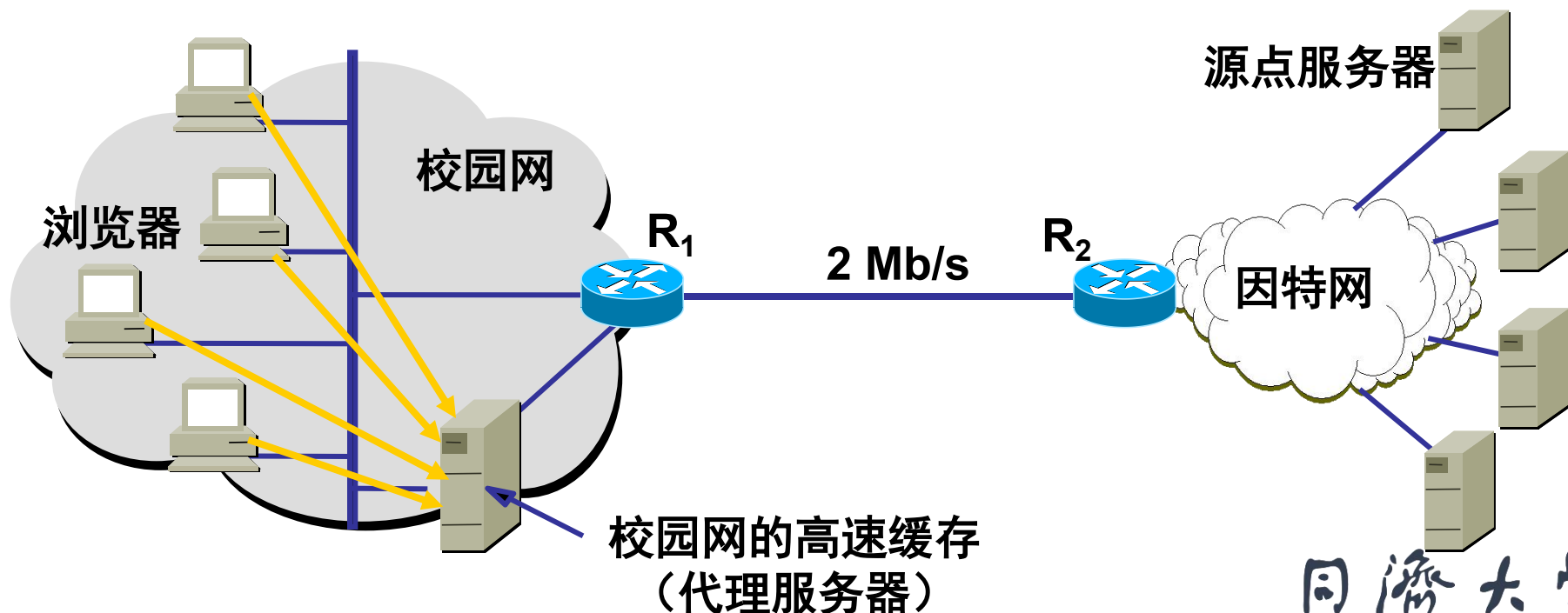
- 代理服务器(proxy server)又称为**万维网高速缓存**(Web cache), 它代表浏览器发出 HTTP 请求。
- 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求相同的新请求到达时, 万维网高速缓存就把暂存的响应发送出去, 而不需要按 URL 的地址再去因特网访问该资源。

没有使用高速缓存的情况



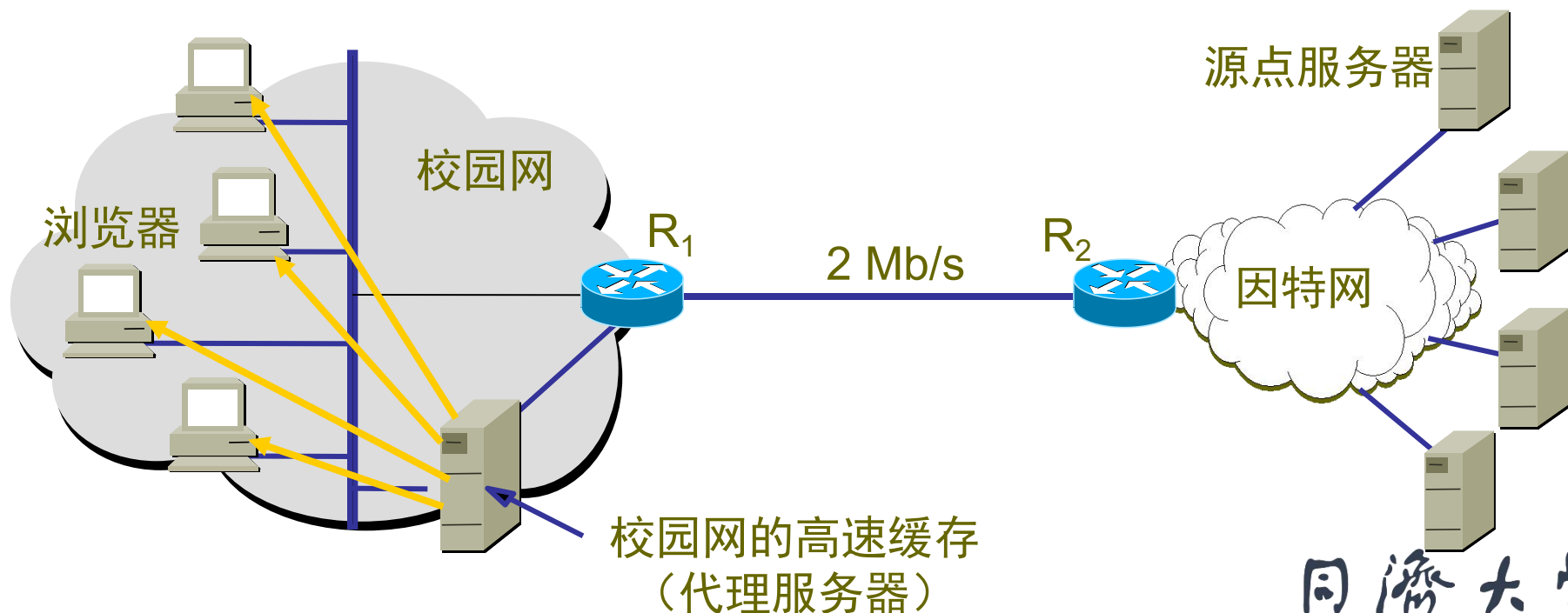
使用高速缓存的情况

(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



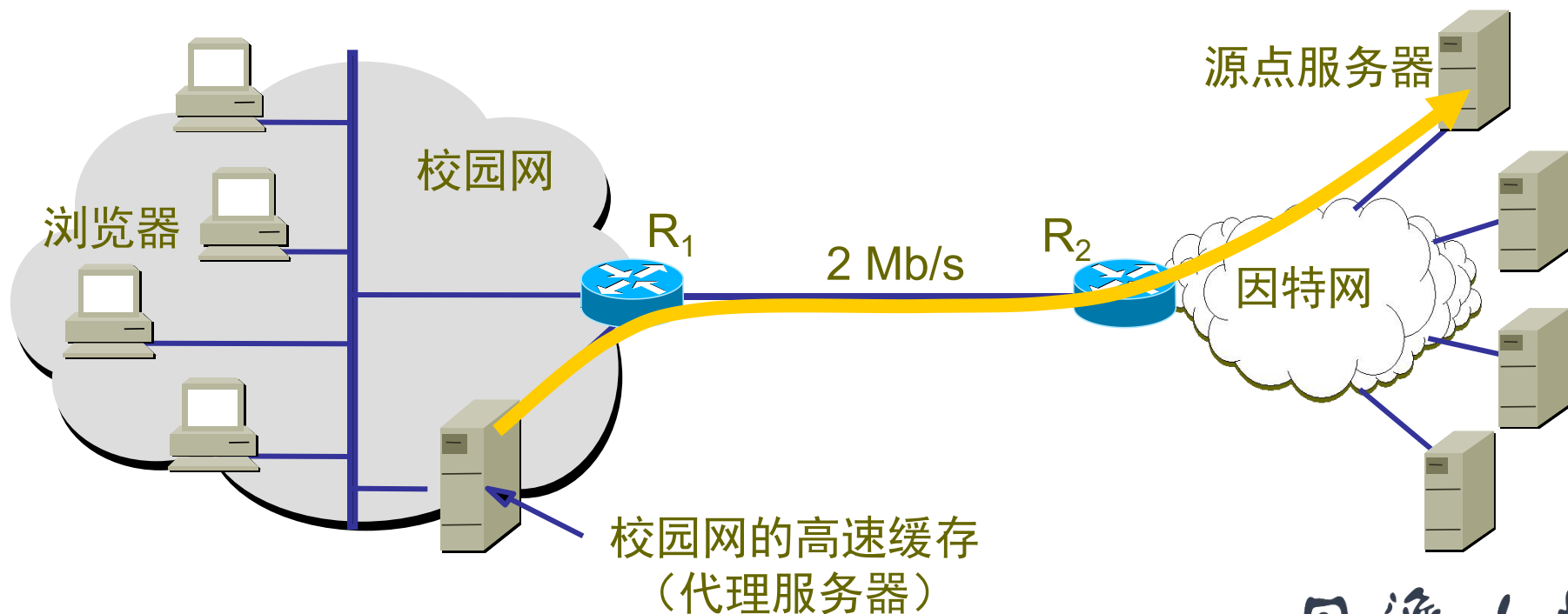
使用高速缓存的情况

(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。



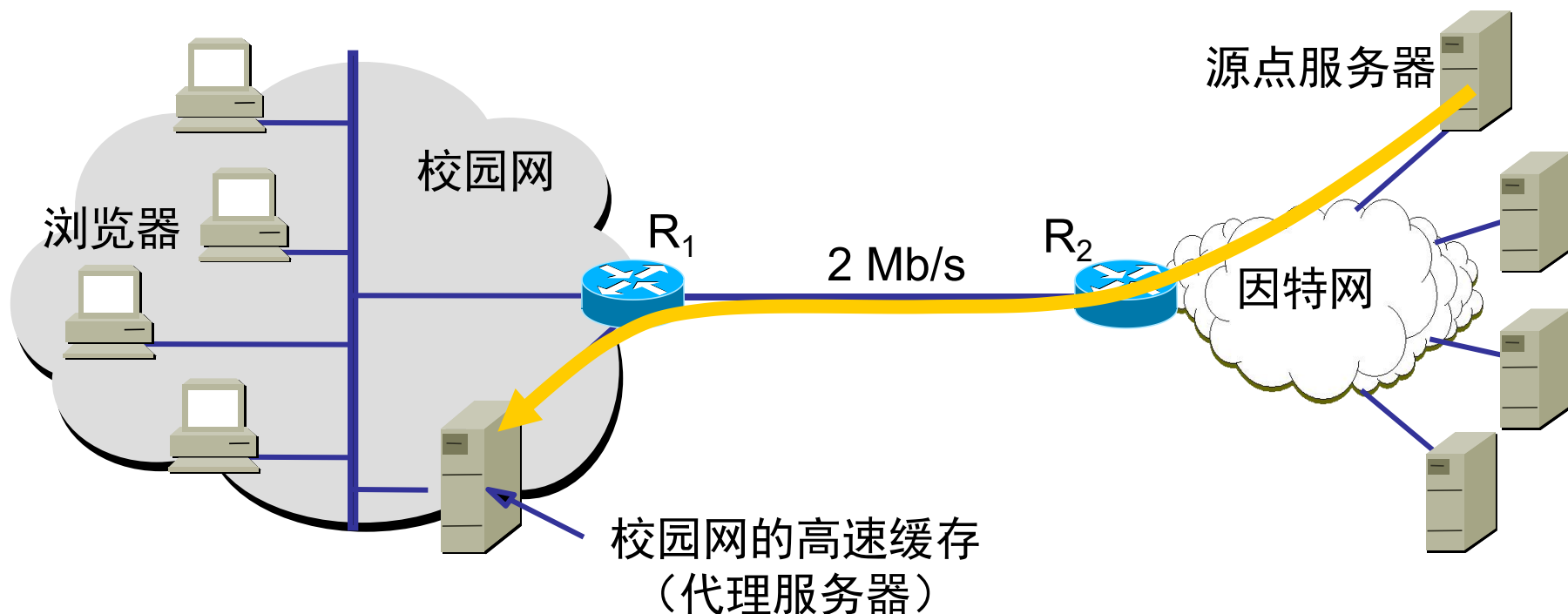
使用高速缓存的情况

(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。



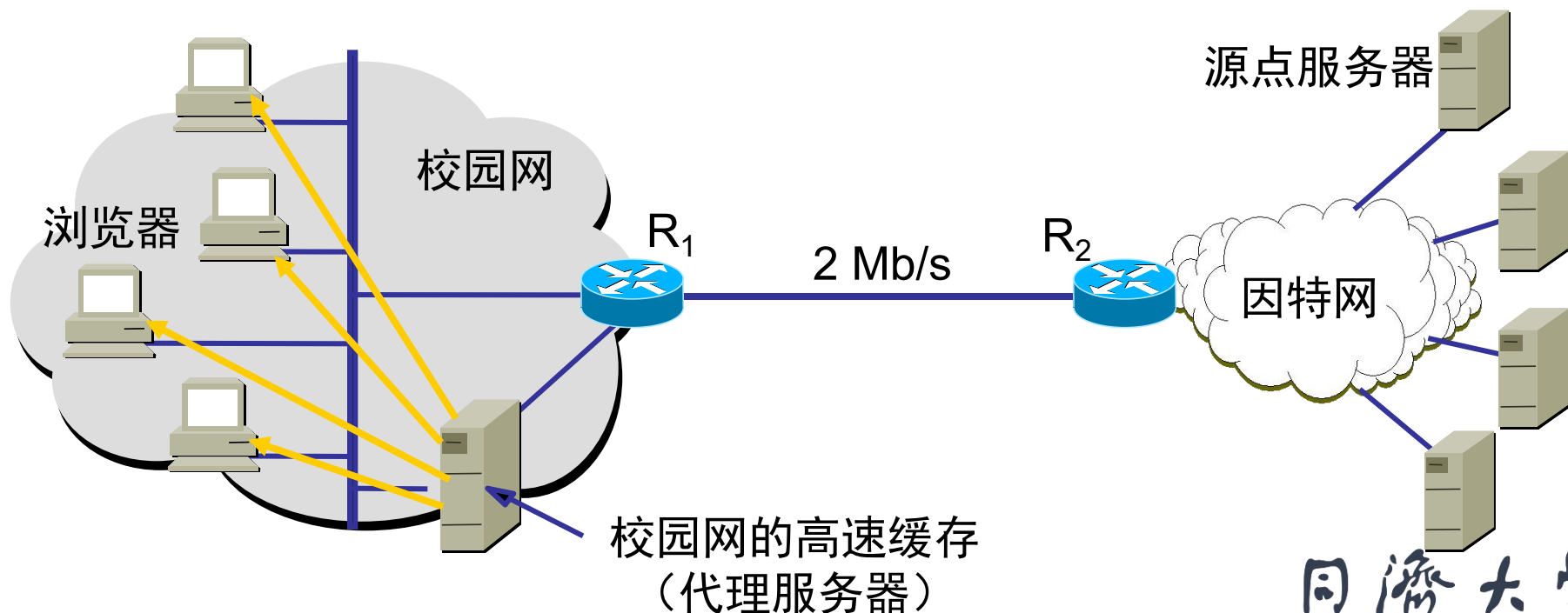
使用高速缓存的情况

(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。

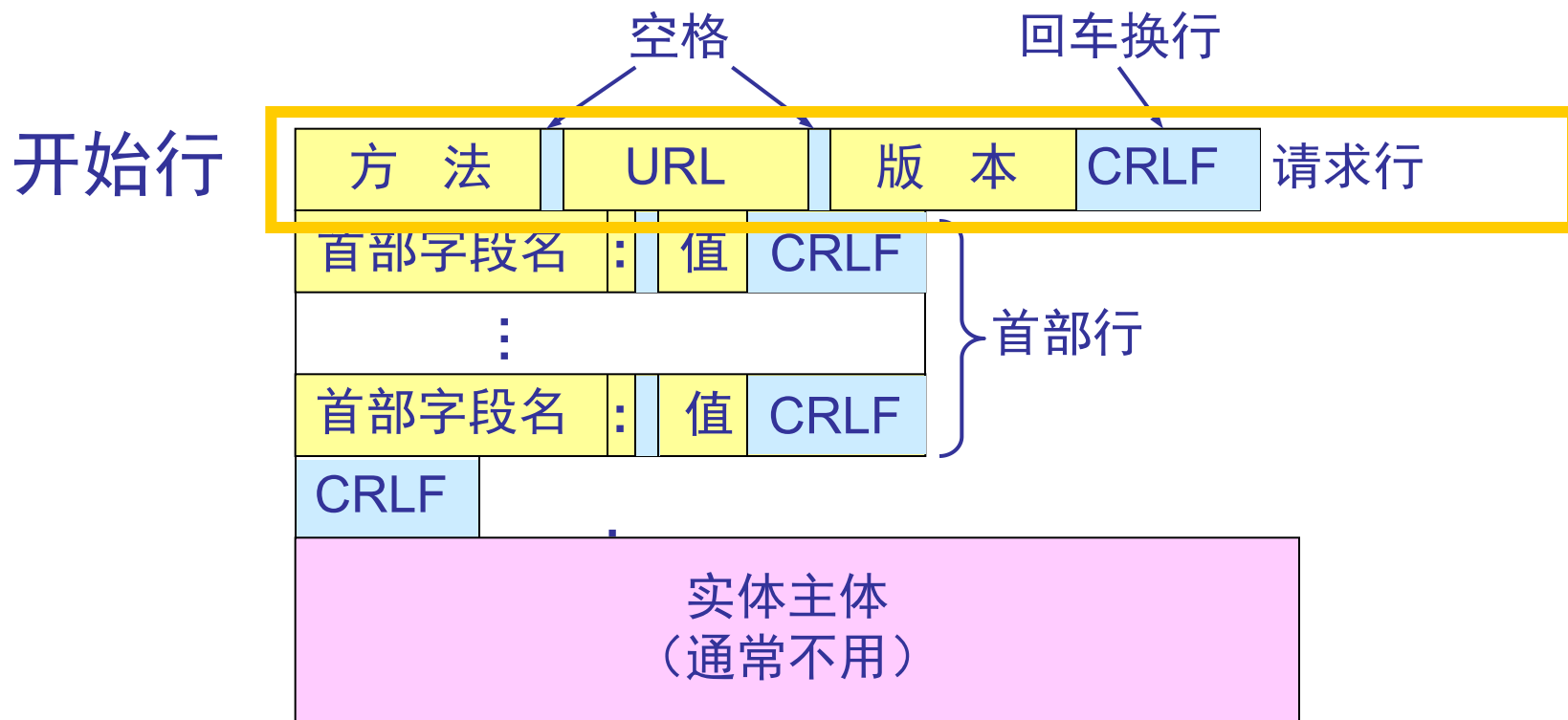


HTTP 有两类报文：

- **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。

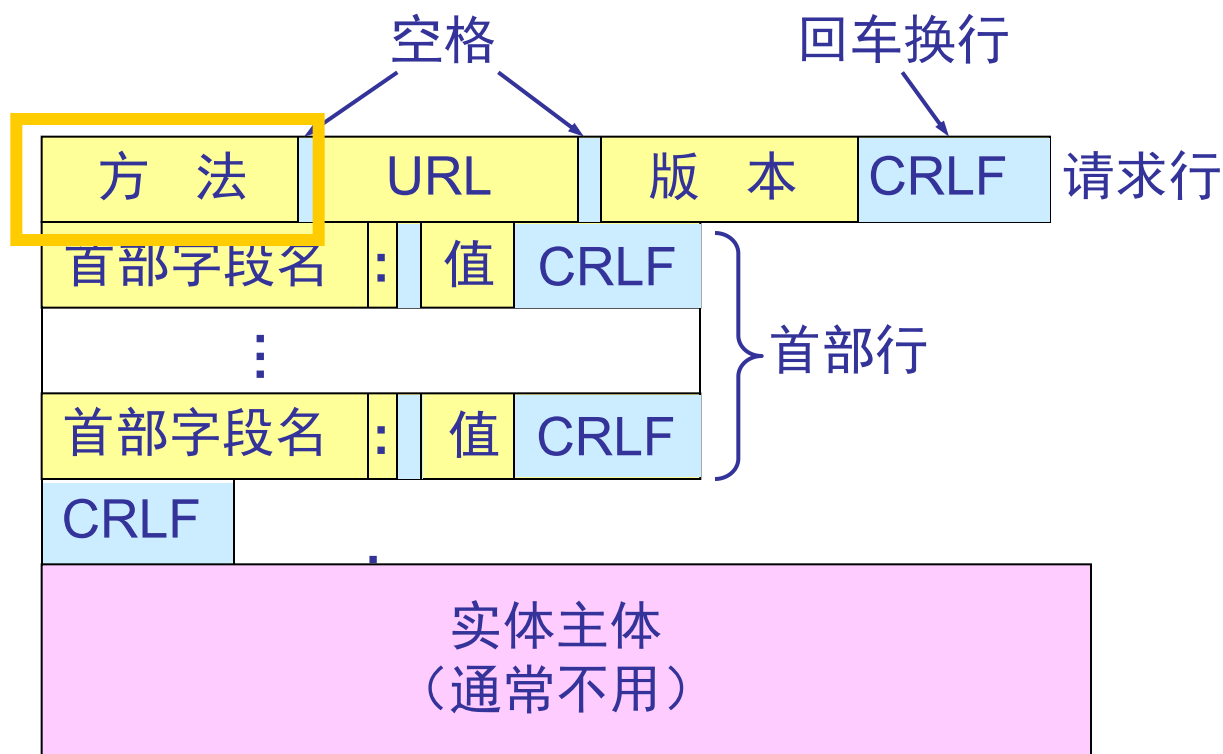
由于 HTTP 是面向文本的(text-oriented)，因此在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的。

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）

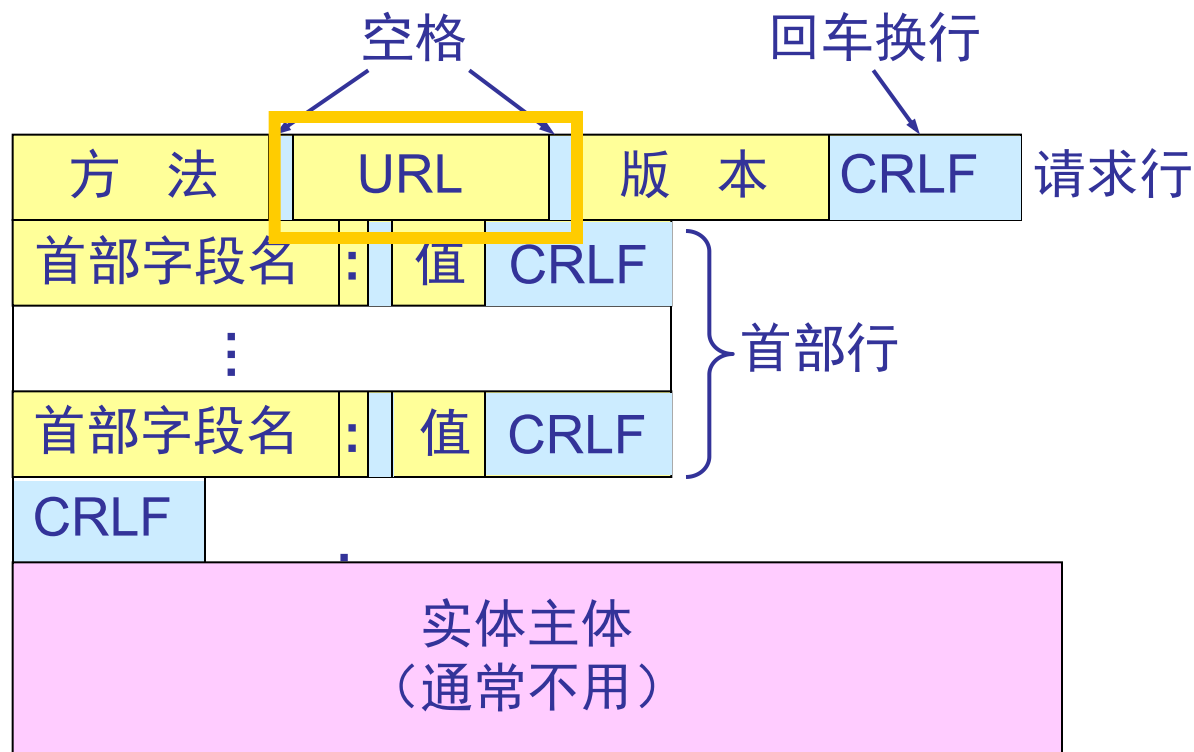


“**方法**”是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些**命令**。因此，请求报文的类型是由它所采用的方法决定的。

HTTP 请求报文的一些方法

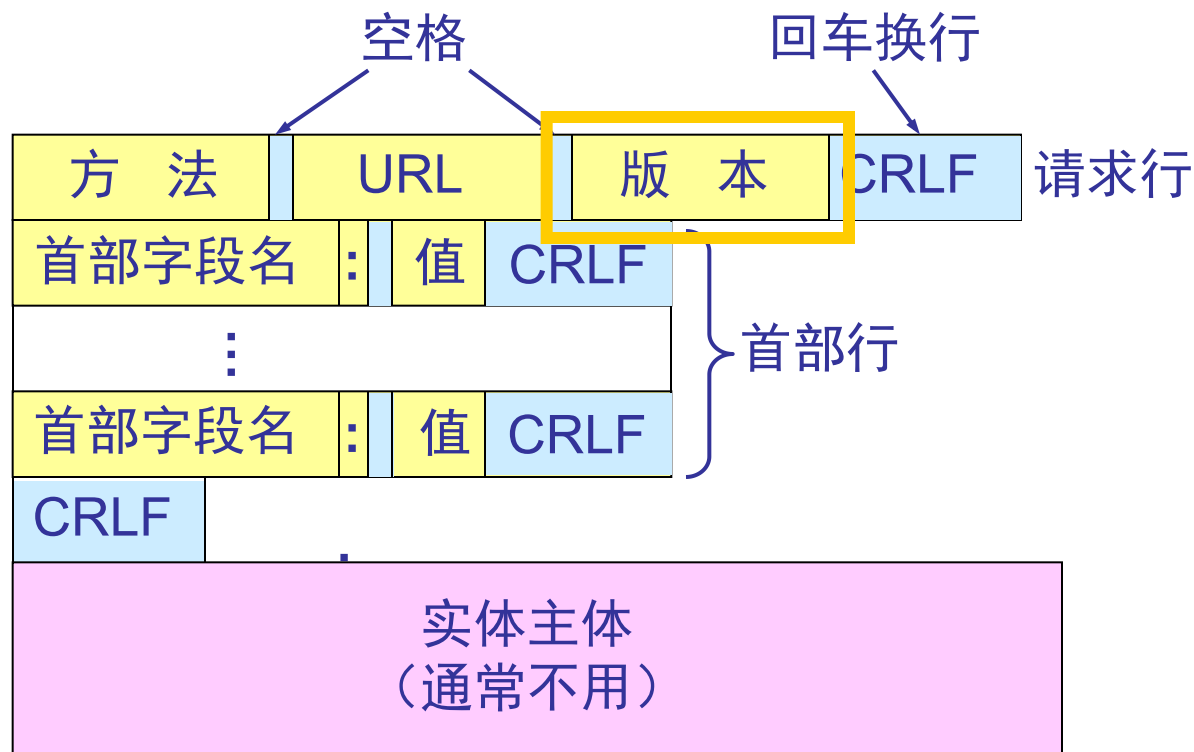
方法 (操作)	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL 所标志的信息
HEAD	请求读取由 URL 所标志的信息的首部
POST	给服务器添加信息 (例如, 注释)
PUT	在指明的 URL 下存储一个文档
DELETE	删除指明的 URL 所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

HTTP 的报文结构（请求报文）



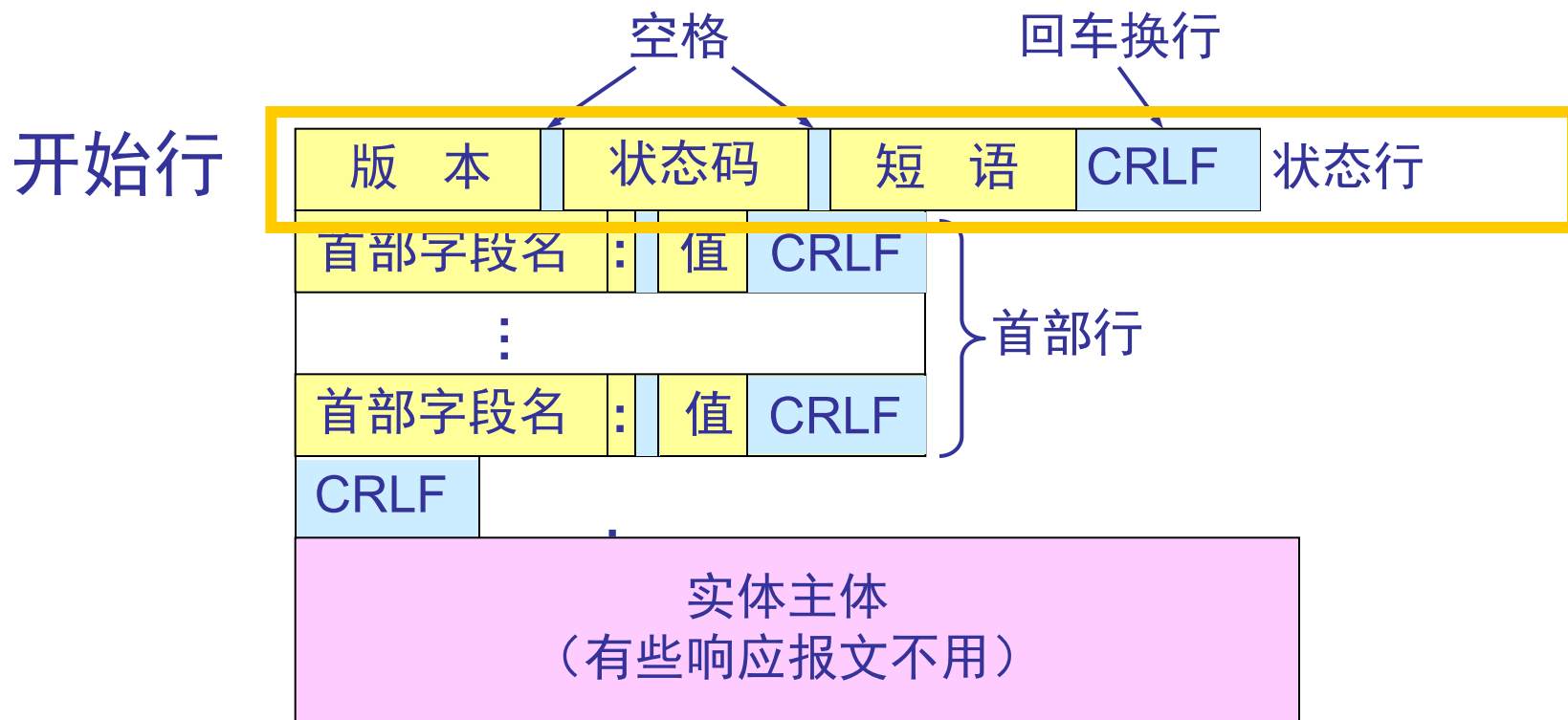
“URL”是所请求的资源的 URL。

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。

- HTTP/1.1 200 Success (成功)
- HTTP/1.1 202 Accepted (接受)
- HTTP/1.1 400 Bad Request (错误的请求)
- HTTP/1.1 404 Not Found (找不到)

- Cookie意为“甜饼”，是由W3C组织提出，最早由Netscape社区发展的一种机制。目前Cookie已经成为标准，所有的主流浏览器如IE、Netscape、Firefox、Opera等都支持Cookie。
- 由于HTTP是一种无状态的协议，服务器单从网络连接上无从知道客户身份。怎么办呢？就给客户端们颁发一个通行证吧，每人一个，无论谁访问都必须携带自己通行证。这样服务器就能从通行证上确认客户身份了。这就是Cookie的工作原理。
- Cookie可以导出导入。

- 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。

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

- 当 HTML 文档是以.html 或 .htm 为后缀时，浏览器才对此文档各种标签解释。
- 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。

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

- HTML 用一对标签（即一个开始标签和一个结束标签）或几对标签来标识一个元素。
- 开始标签由一个小于字符 “<”、一个标签名、和一个大于字符 “>”组成。
- 结束标签和开始标签的区别只是在小于字符的后面要加上一个斜线字符 “/”。
- 标签名并不区分大写和小写。
- 有一些标签可以将结束标签省略。

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

<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

插入图像

开始标签

结束标签

插入图像

高度是 300 像素

宽度是 500 像素

插入的图像文件名是 example.jpg

两种不同的链接

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

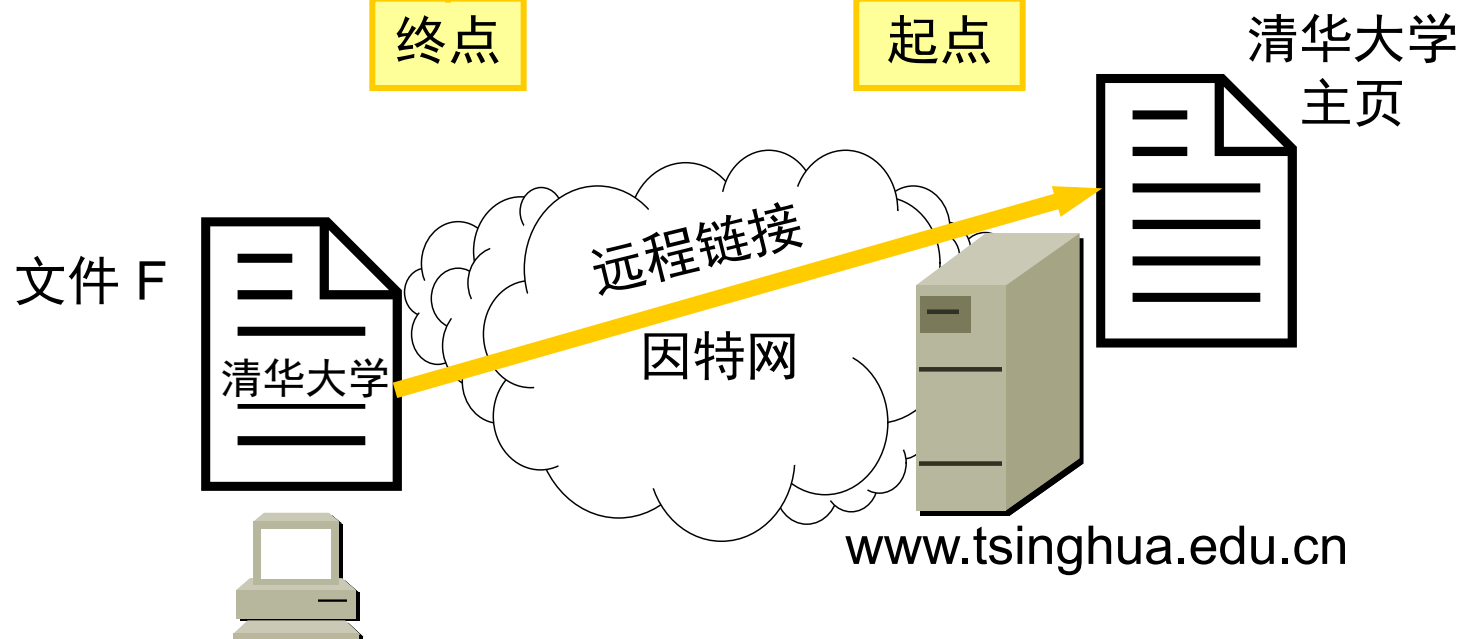
远程链接举例

远程链接：

清华大学

终点

起点



- 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。

- XHTML (Extensible HTML) 是可扩展超文本标记语言，它与 HTML 4.01 几乎是相同的。
- 但 XHTML 是更严格的 HTML 版本，也是一个 W3C 标准（2000年1月），是作为一种 XML 应用被重新定义的 HTML，并将逐渐取代 HTML。
- 新的浏览器都支持 XHTML。

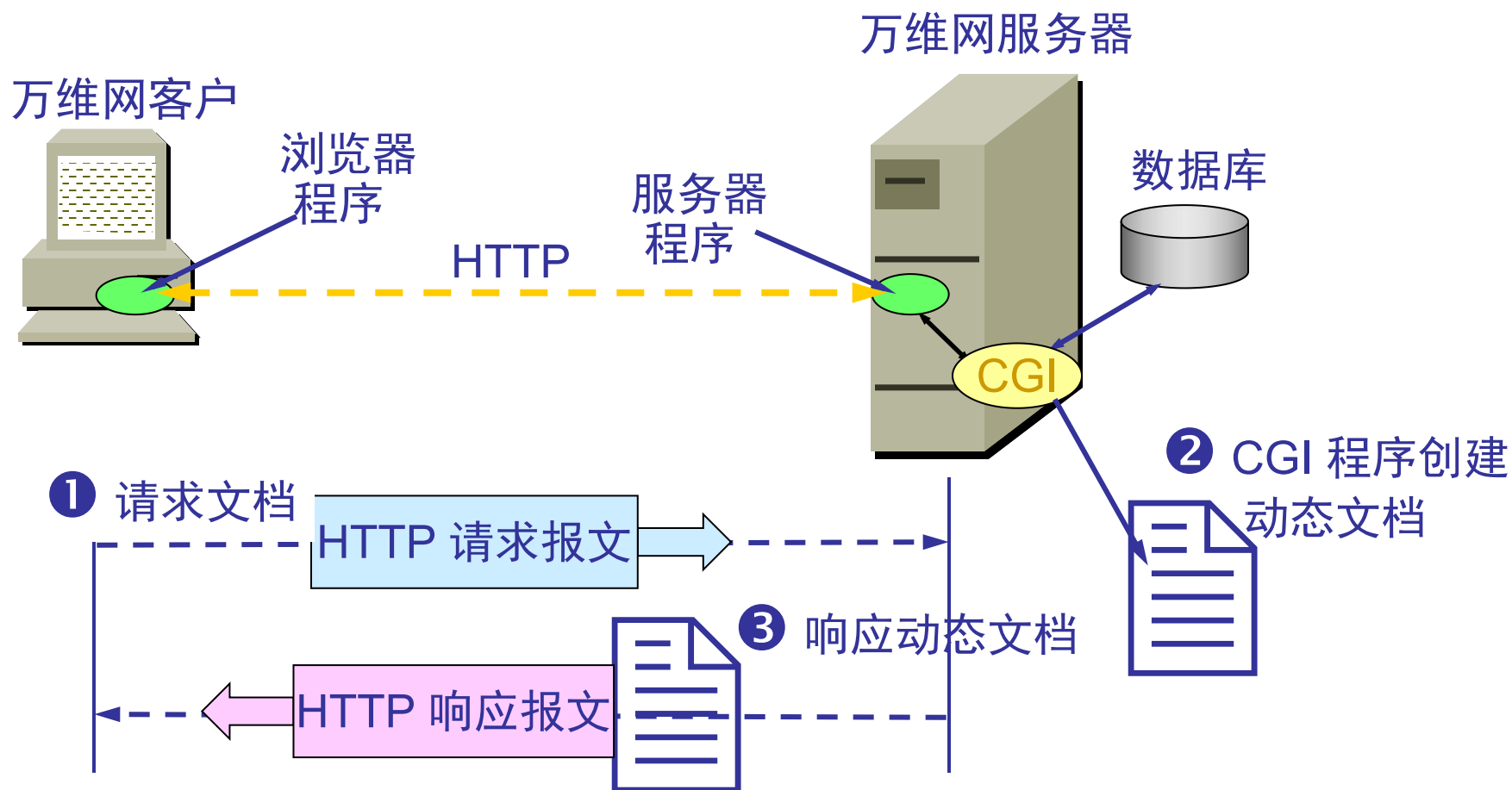
- CSS (Cascading Style Sheets) 是层叠样式表，它是一种样式表语言，用于为 HTML 文档定义布局。
- CSS 与 HTML 的区别就是：HTML 用于结构化内容，而 CSS 则用于格式化结构化的内容。

- **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。

要实现动态文档就必须在以下两个方面对万维网服务器的功能进行扩充：

- (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

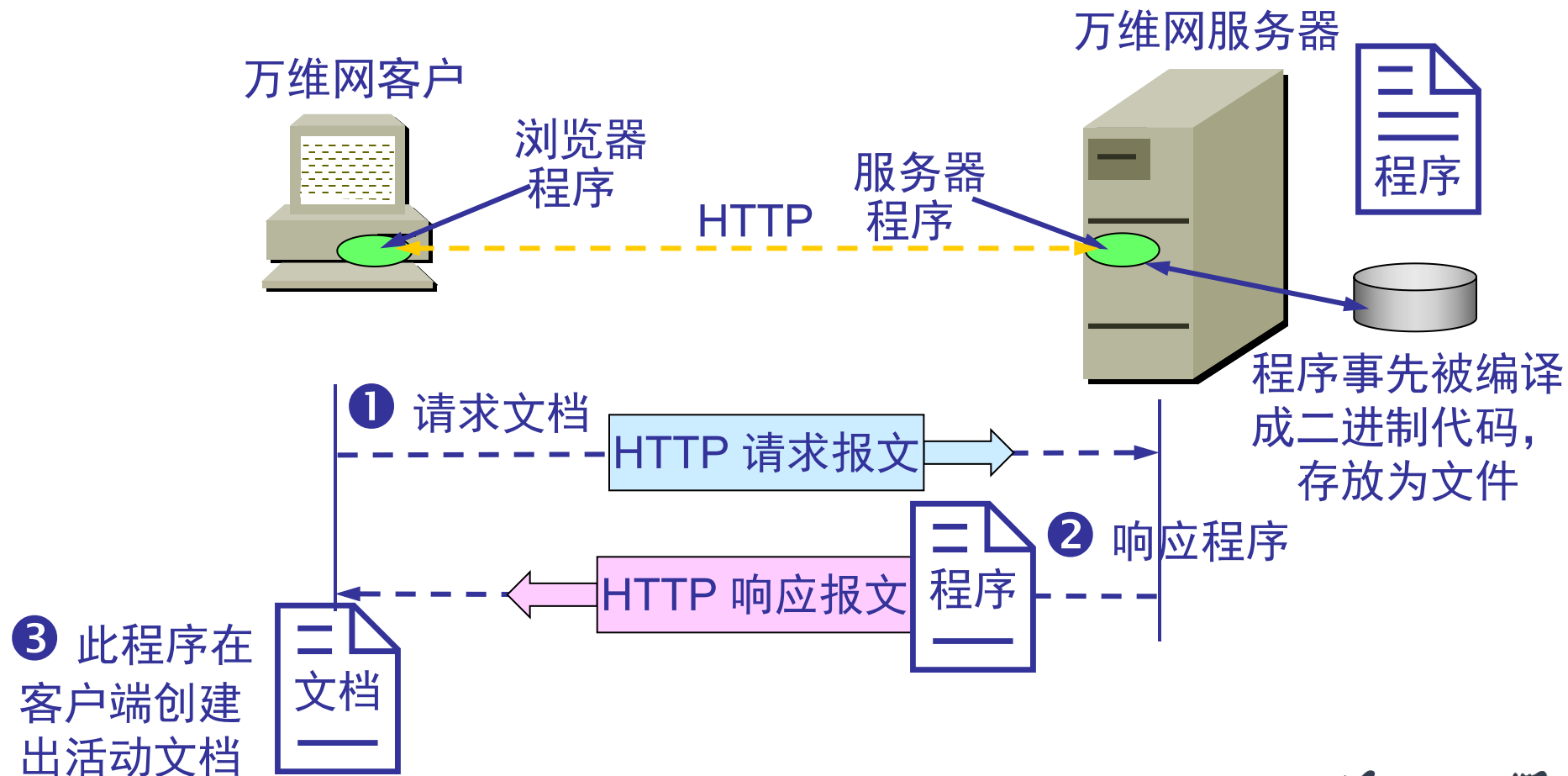


- CGI (Common Gateway Interface) 是一种标准，它定义了动态文档应如何创建，输入数据应如何提供给应用程序，以及输出结果应如何使用。
- 万维网服务器与 CGI 的通信遵循 CGI 标准。
- “通用”：CGI 标准所定义的规则对其他任何语言都是通用的。
- “网关”：CGI 程序的作用像网关。
- “接口”：有一些已定义好的变量和调用等可供其他 CGI 程序使用。

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

- **活动文档**(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



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

1. 全文检索搜索和分类目录搜索

- 在万维网中用来进行搜索的程序叫做**搜索引擎**。
- **全文检索搜索引擎**是一种纯技术型的检索工具。它的工作原理是通过搜索软件到互联网上的各网站收集信息，找到一个网站后可以从这个网站再链接到另一个网站。然后按照一定的规则建立一个很大的在线数据库供用户查询。
- 用户在查询时只要输入关键词，就从已经建立的索引数据库上进行查询（并不是实时地在互联网上检索到的信息）。

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



一些著名的搜索引擎

□ 最著名的全文检索搜索引擎：

- Google（谷歌）(www.google.com)
- 百度 (www.baidu.com)

□ 最著名的分类目录搜索引擎：

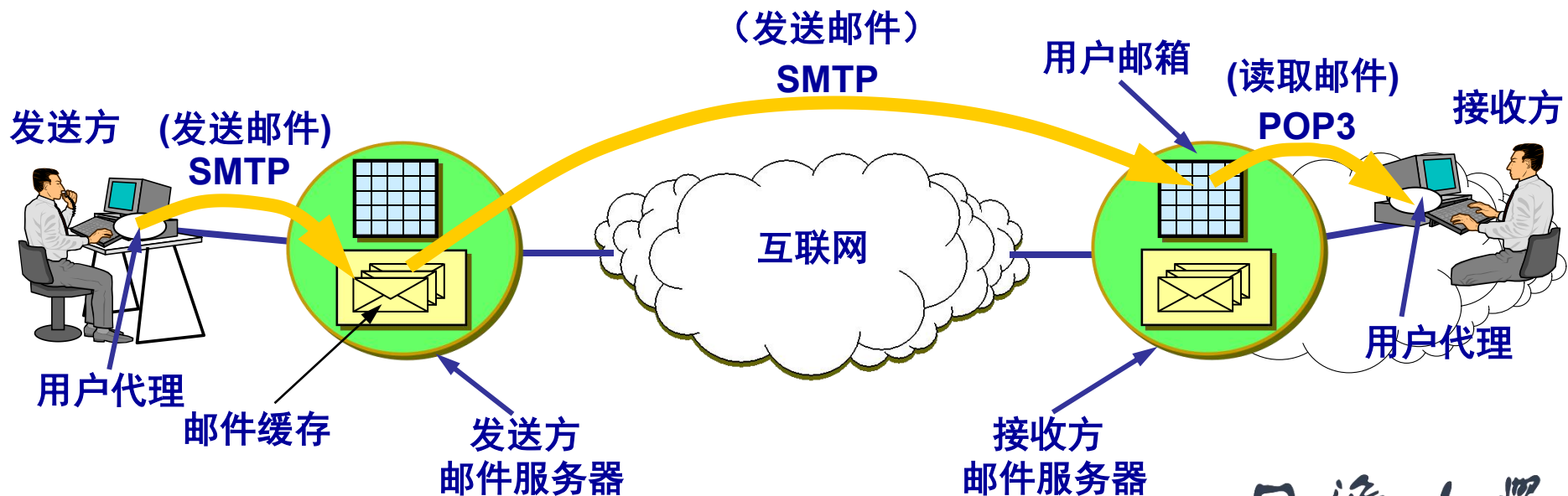
- 新浪 (www.sina.com)
- 搜狐 (www.sohu.com)
- 网易 (www.163.com)

6.5.1 电子邮件概述

- ❑ 电子邮件(e-mail)是因特网上使用得最多的和最受用户欢迎的一种应用。
- ❑ 电子邮件把邮件发送到收件人的邮件服务器，并放在其中的收信人邮箱中，收信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。

- 发送邮件的协议：SMTP (Simple Mail Transfer Protocol)
- 读取邮件的协议：POP3 (Post Office Protocol - Version 3) 和 IMAP (Internet Mail Access Protocol)
- 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 机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。

- 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，最重要的就是收信人的地址。

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：

收信人邮箱名@邮箱所在主机的域名
符号 “@”读作 “at”，表示 “在” 的意思。

- 例如，电子邮件地址
xiexiren@tsinghua.org.cn

这个用户名在该域名的范围内是惟一的。

邮箱所在的主机的域名在全世界必须是惟一的

- 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。

- 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。

- “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。

- 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 (Internet Message Access Protocol) 也是按**客户服务器**方式工作，现在较新的版本是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 还允许收信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 缺点是如果用户没有将邮件复制到自己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

- ❑ 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。

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。

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。



- 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。

- 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。

- 连接在因特网的计算机的协议软件需要配置的项目包括：
 - (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。

6.6动态主机配置协议

- ▣ 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。

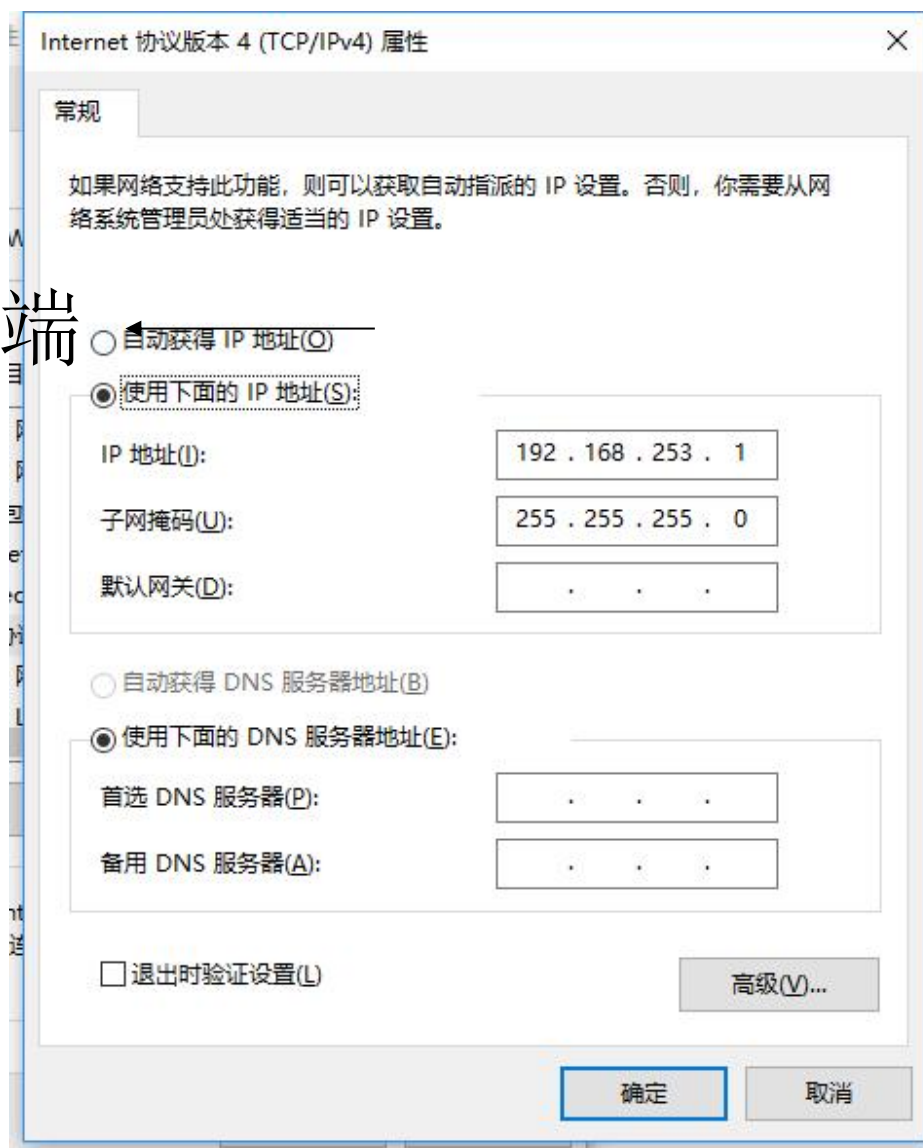
□ 使用静态地址的情况：

- IP地址不经常更改的设备就使用静态地址。比如企业中服务器会单独在一个网段，很少更改IP地址或移动到到其他网段，这些服务器通常使用静态地址，使用静态地址还方便企业员工使用地址访问这些服务器。比如学校机房，都是台式机，很少移动，这些计算机最好也使用静态地址。

□ 使用动态地址的情况：

- 网络中的计算机不固定，就应该使用动态地址。
- 无线设备最好也使用动态地址。
- ADSL拨号上网通常也是使用自动获得IP地址。

DHCP客户端

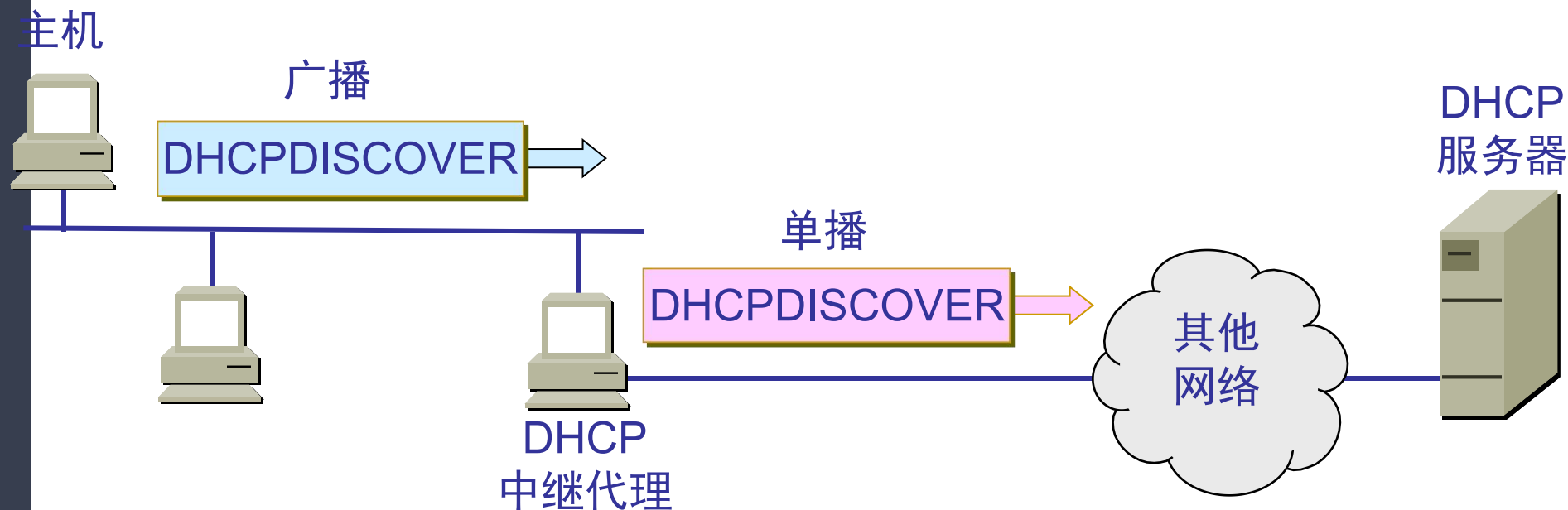


- DHCP客户端会在以下所列举的几种情况下，从DHCP服务器获取一个新的IP地址。
 - 该客户端计算机是第一次从DHCP服务器获取IP地址。
 - 该客户端计算机原先所租用的IP地址已经被DHCP服务器收回，而且已经又租给其他计算机了，因此该客户端需要重新从DHCP服务器租用一个新的IP地址。
 - 该客户端自己释放原先所租用的IP地址，并要求租用一个新的IP地址。
 - 客户端计算机更换了网卡。
 - 客户端计算机转移到另一个网段。

- 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）。

- 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

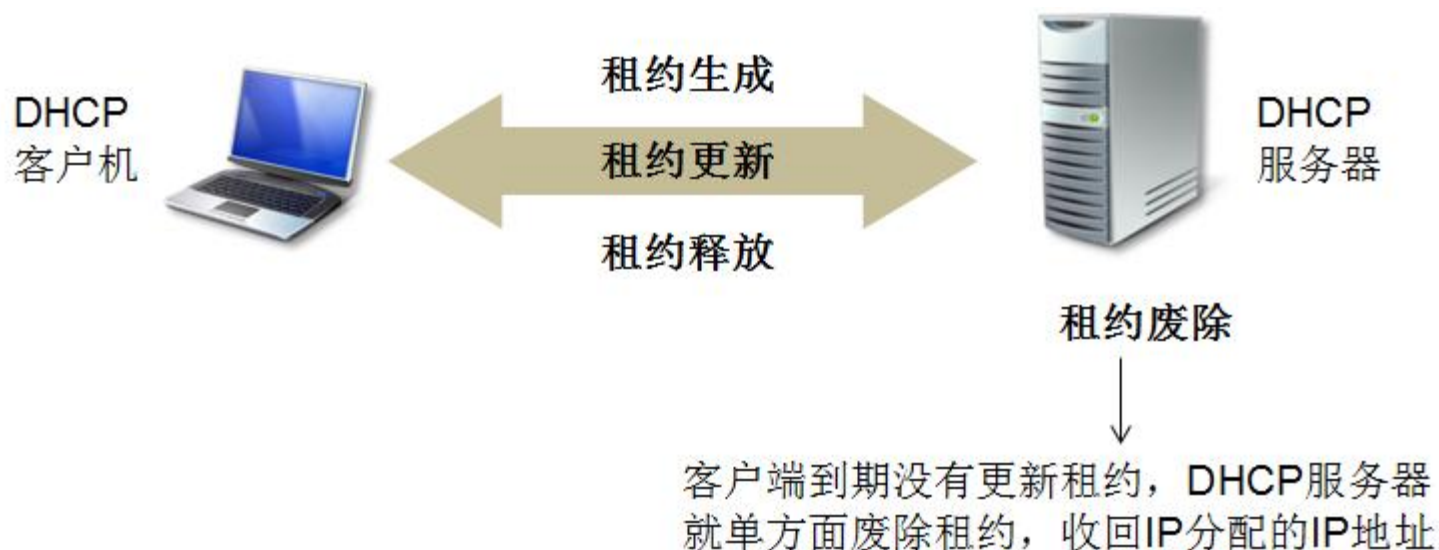


注意：DHCP 报文只是 UDP 用户数据报中的数据。

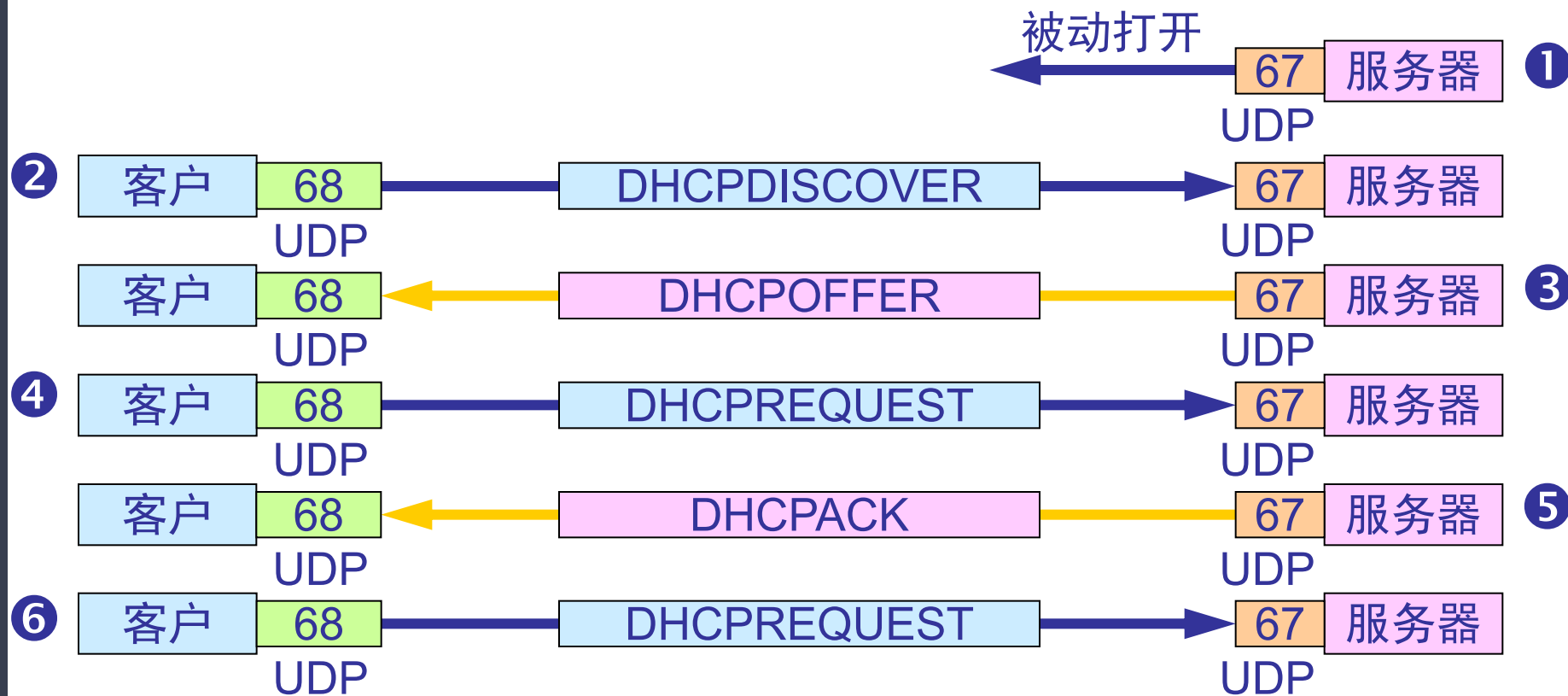
租用期(lease period)

- DHCP 服务器分配给 DHCP 客户的 IP 地址是临时的，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。

地址以租约的形式提供给客户端

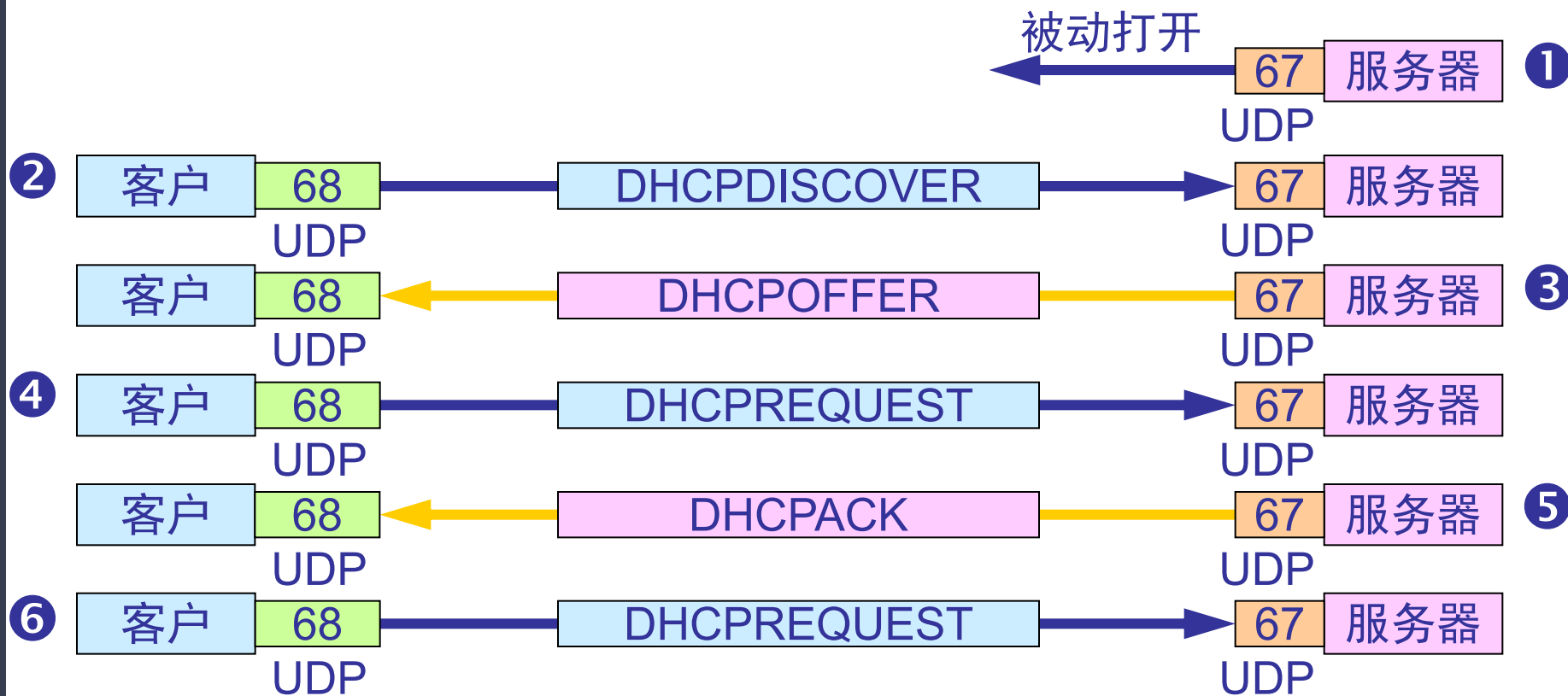


DHCP 协议的工作过程



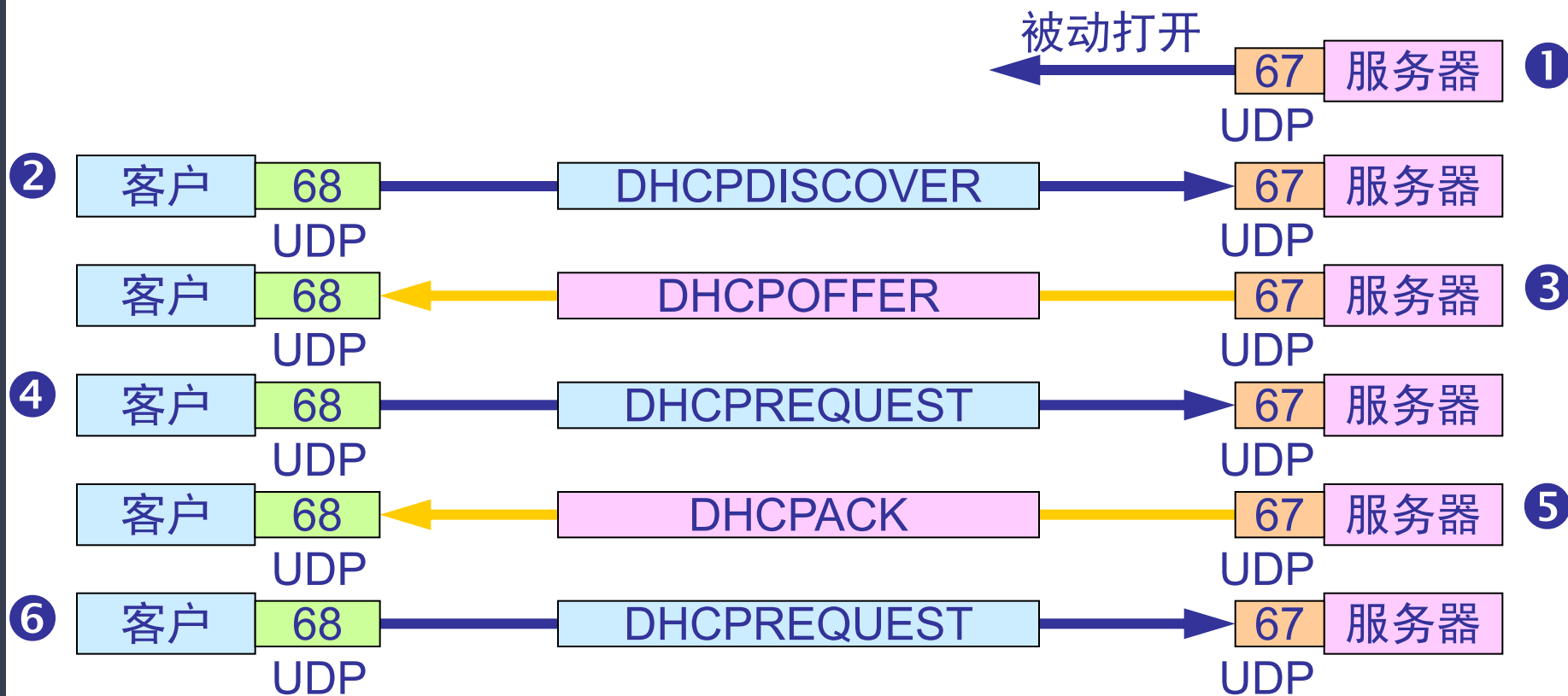
❶：DHCP 服务器被动打开 UDP 端口 67，等待客户端发来的报文。

DHCP 协议的工作过程



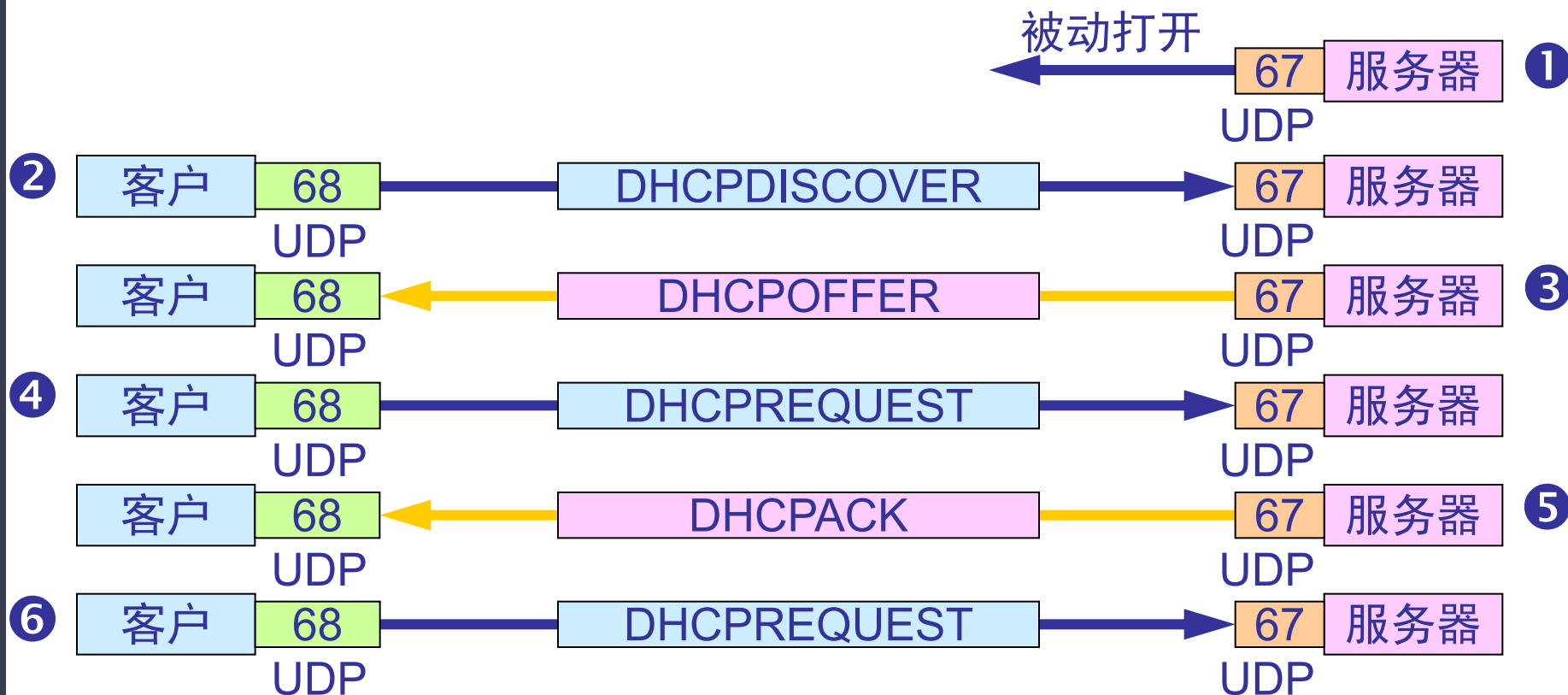
②： DHCP 客户从 UDP 端口 68 发送 DHCP 发现报文。

DHCP 协议的工作过程



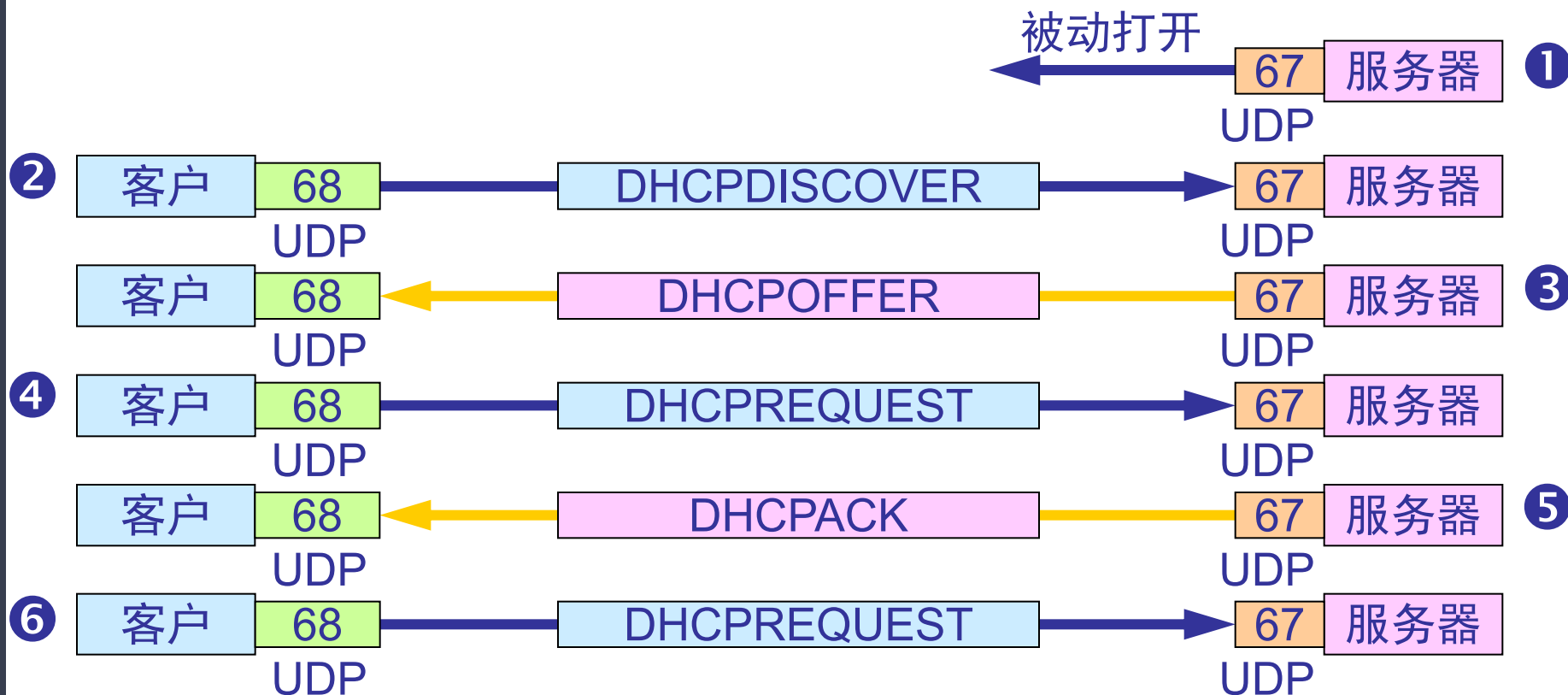
③： 凡收到 DHCP 发现报文的 DHCP 服务器都发出 DHCP 提供报文，因此 DHCP 客户可能收到多个 DHCP 提供报文。

DHCP 协议的工作过程



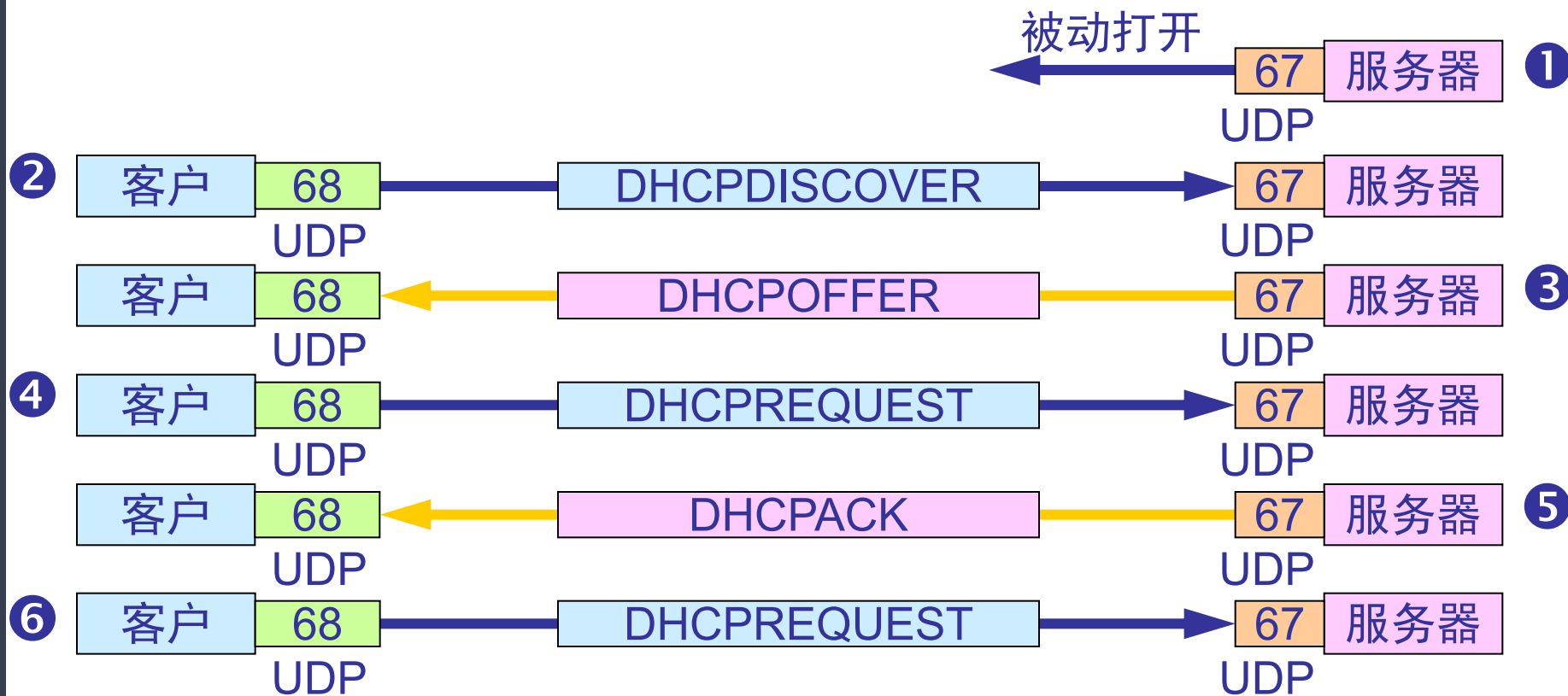
④： DHCP 客户从几个 DHCP 服务器中选择其中的一个，并向所选择的 DHCP 服务器发送 DHCP 请求报文。

DHCP 协议的工作过程



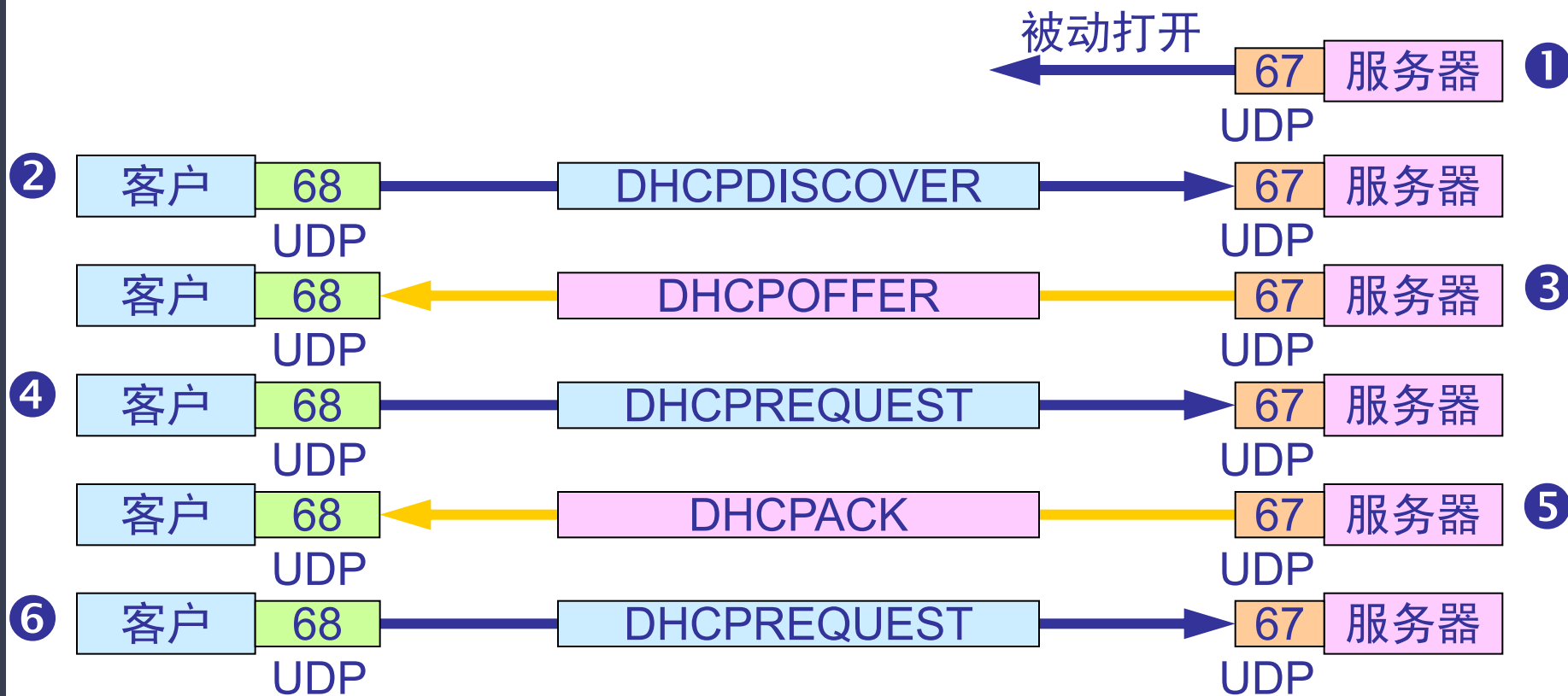
⑤： 被选择的 DHCP 服务器发送确认报文 DHCPACK，进入已绑定状态，客户端并可开始使用得到的临时 IP 地址了。

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。

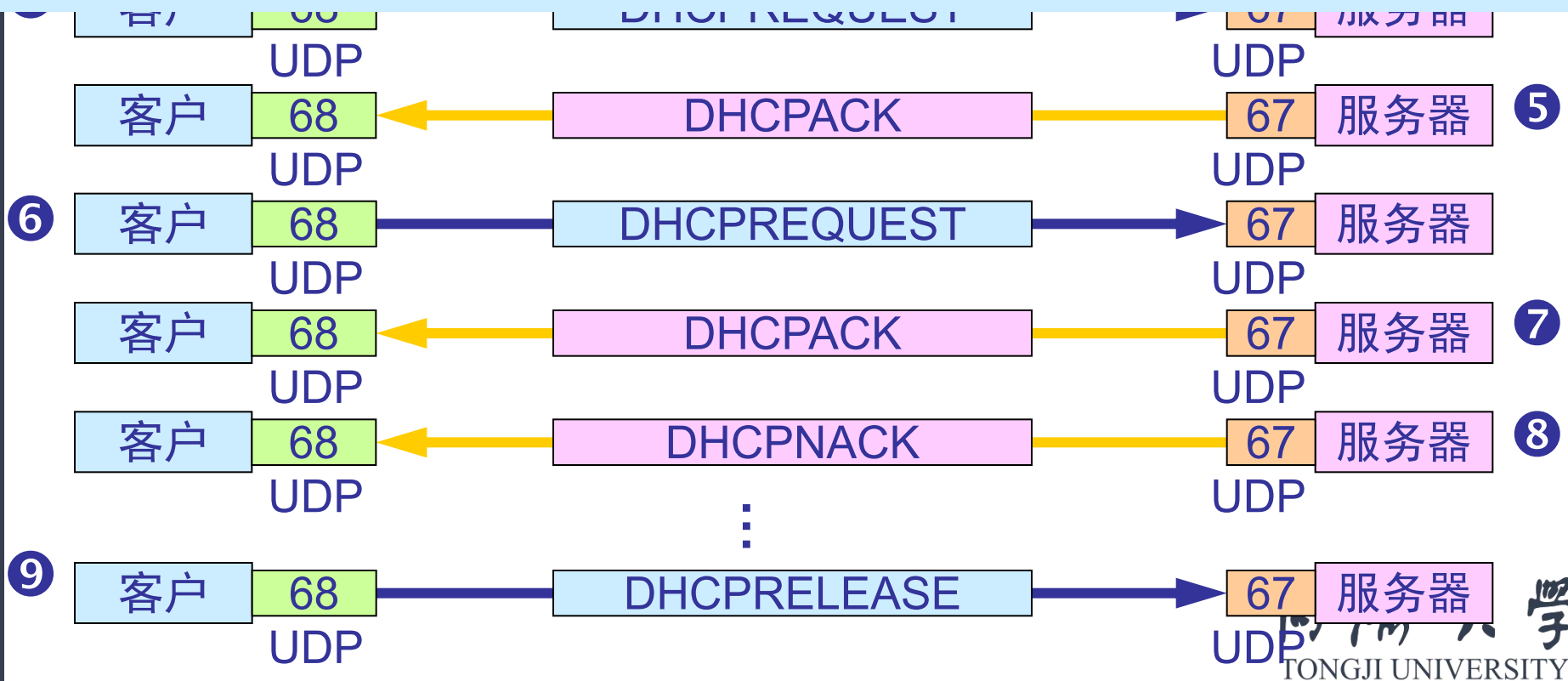
DHCP 协议的工作过程



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。

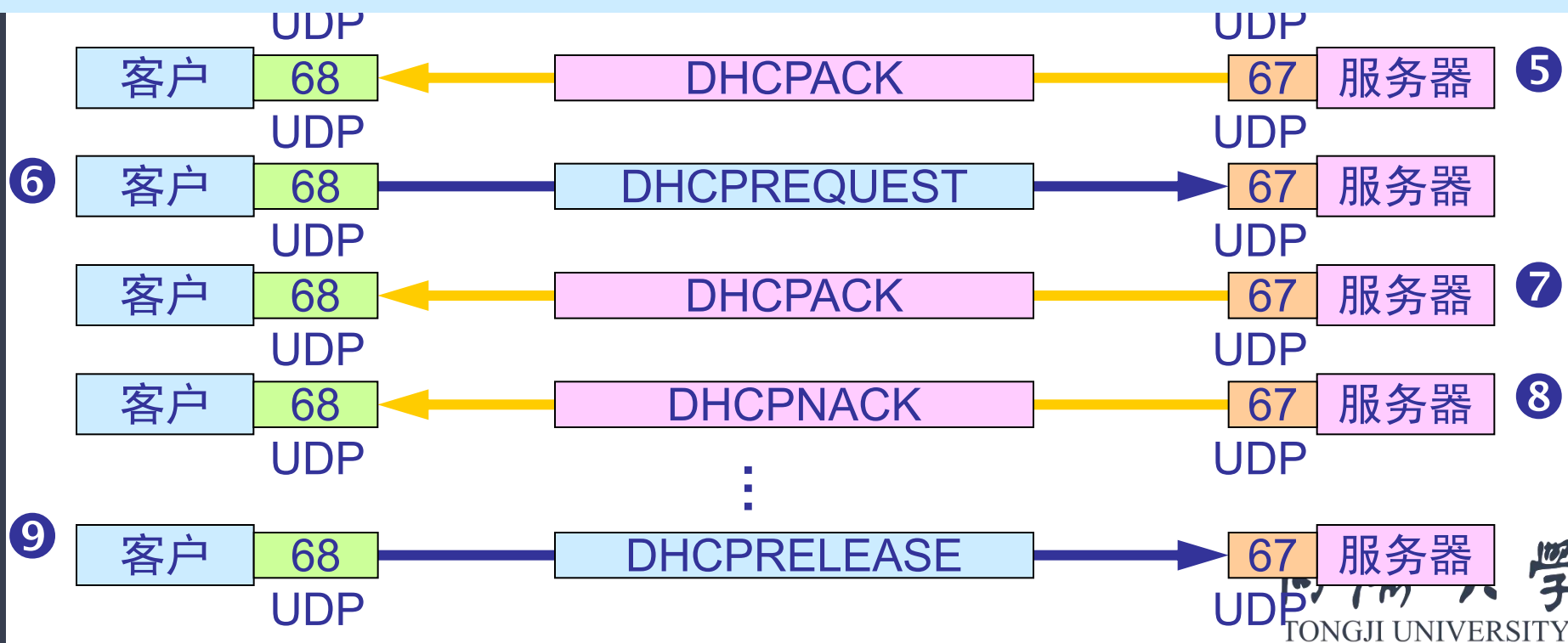
DHCP 协议的工作过程

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。



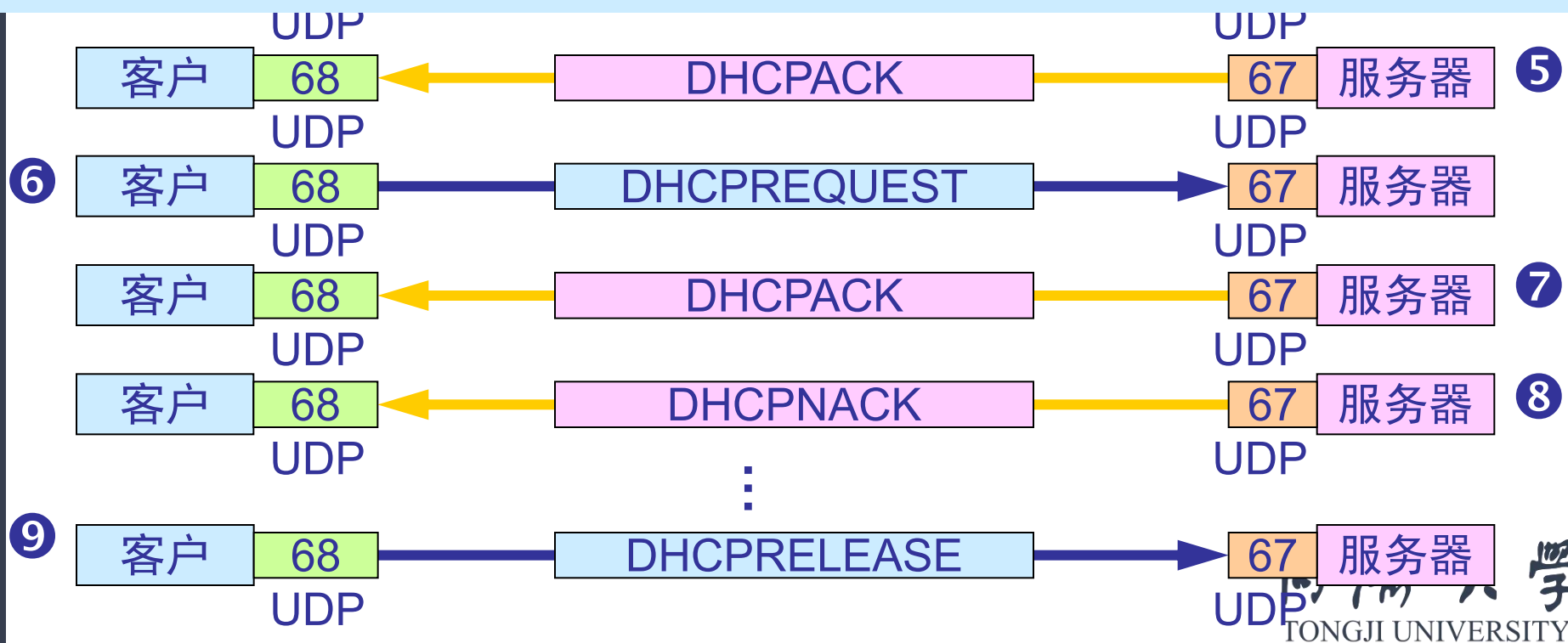
DHCP 协议的工作过程

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）。



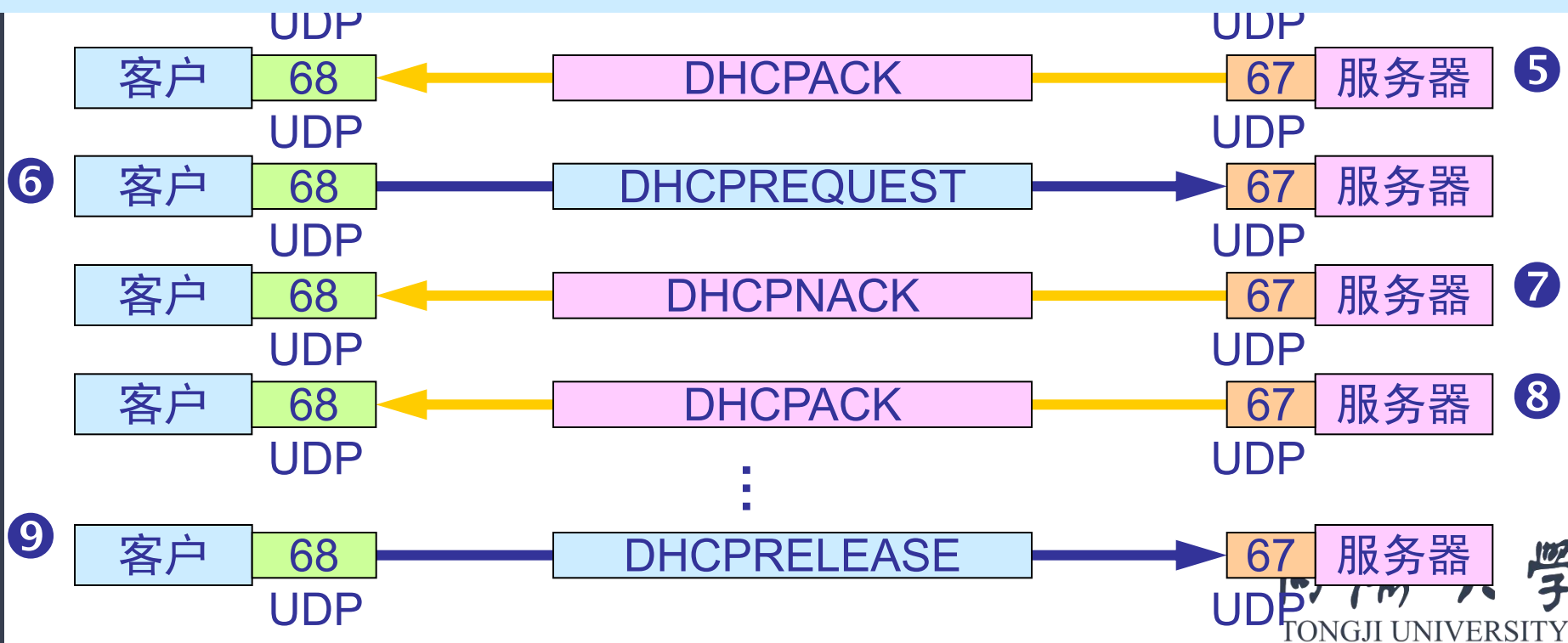
DHCP 协议的工作过程

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。



- 通本章给大家介绍了几个常见的应用层协议，我们能够的到什么结论呢？
 - 每个应用层协议都是为了解决特定问题、实现特定功能。比如HTTP协议为了能让浏览器请求网页、Web服务器给客户端返回网页，DNS协议为了实现域名解析，SMTP协议为了实现发送电子邮件，POP3为了实现让电子邮件客户端从服务器下载电子邮件，DHCP协议为了DHCP服务器给计算机分配IP地址。
 - 应用层协议就是为了让客户端和服务端能够交换信息提前定义好一些规范。比如客户端需要向服务器发送哪些操作请求（比如HTTP协议定义的访问网站的GET、POST等方法，SMTP协议定义的发送邮件的命令HELO、RCPT TO等），服务器向客户端发送哪些响应（比如网站响应状态代码，SMTP服务器给SMTP客户端返回的状态代码，客户端要能够明白代码代表的意义）。

- 协议三要素：语法、语义和同步。
 - 请求报文和响应报文格式就是协议的语法。
 - 报文中的每个字段不同的值的所代表的意义就是协议的语义。
 - 客户端和服务端交互顺序（比如命令执行的顺序）就是协议的同步。



Thanks for your
attention!