



武汉大学

WUHAN UNIVERSITY

计算机网络

林海

Lin.hai@whu.edu.cn



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
- 6.6 动态主机配置协议 DHCP
- 6.7 简单网络管理协议 SNMP
- 6.8 应用进程跨越网络的通信
- 6.9 P2P 应用



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

- 机器使用**IP**地址
 - 但**IP**地址不便于记住
 - 一旦移动机器，**IP**变化
- 引入可读性好的名字
 - 如www.whu.edu.cn
- 域名系统
 - 一种基于层次的、基于域的命名方案，并且用一个分布式数据库系统加以实现
 - 将主机名映射成**IP**地址



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。



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）

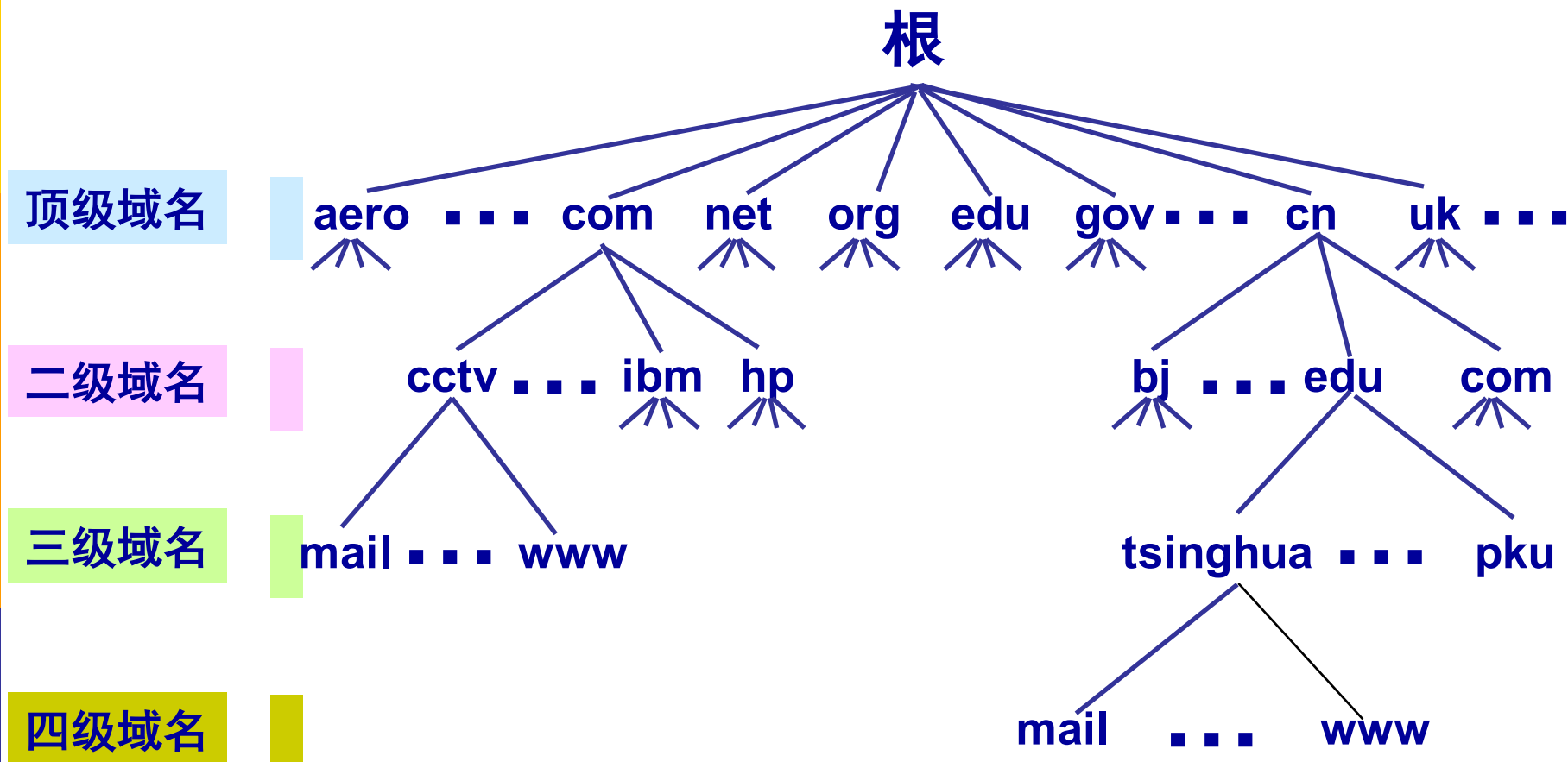
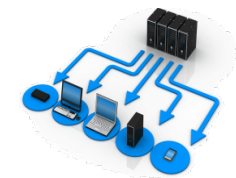
.int （国际组织）



顶级域名 TLD （续）

- (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**。

互联网的域名空间

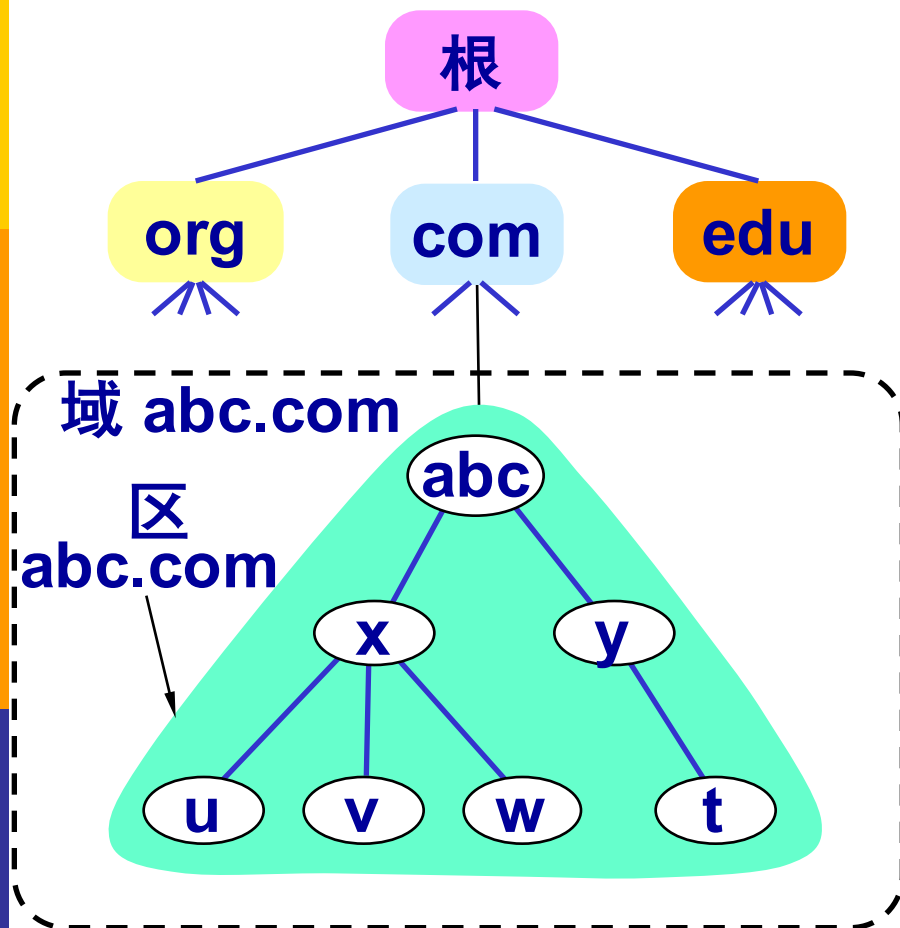
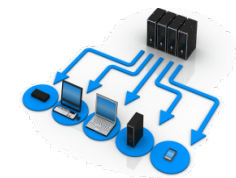




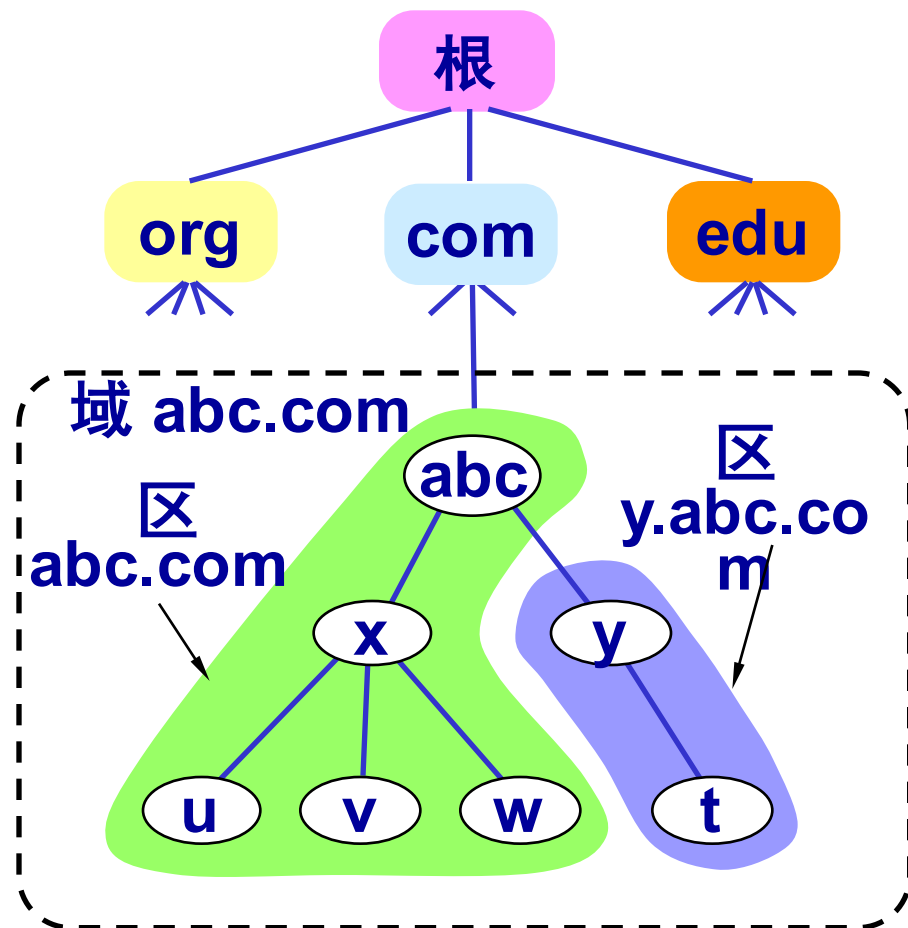
6.1.3 域名服务器

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

区的不同划分方法举例

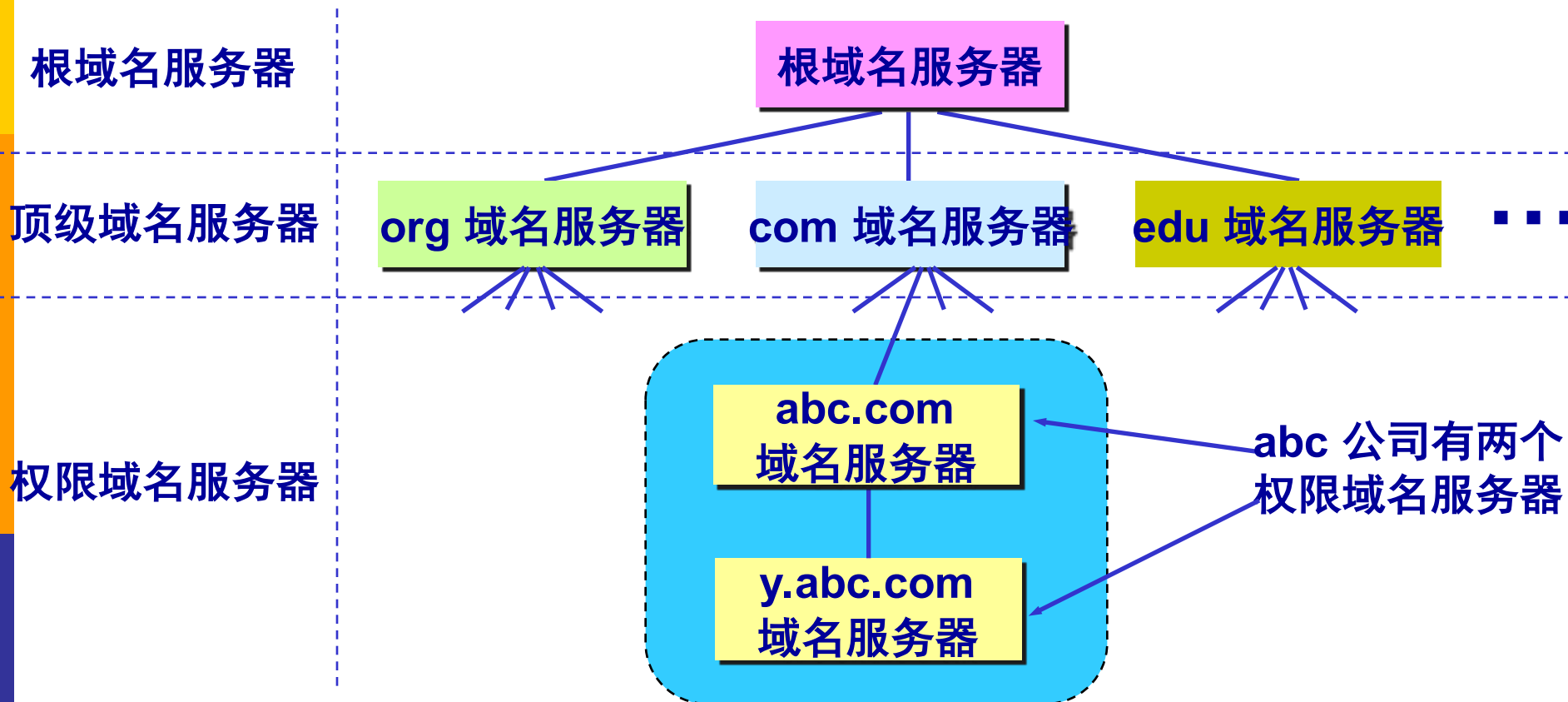


(a) 区 = 域



(b) 区 < 域

树状结构的 DNS 域名服务器





域名服务器有以下四种类型

- 根域名服务器
- 顶级域名服务器
- 权限域名服务器
- 本地域名服务器



根域名服务器

- 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。
 - 13个根域名服务器部署在10个在美国，2个在欧洲，1个在日本。而中国只有3个根域名镜像服务器



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

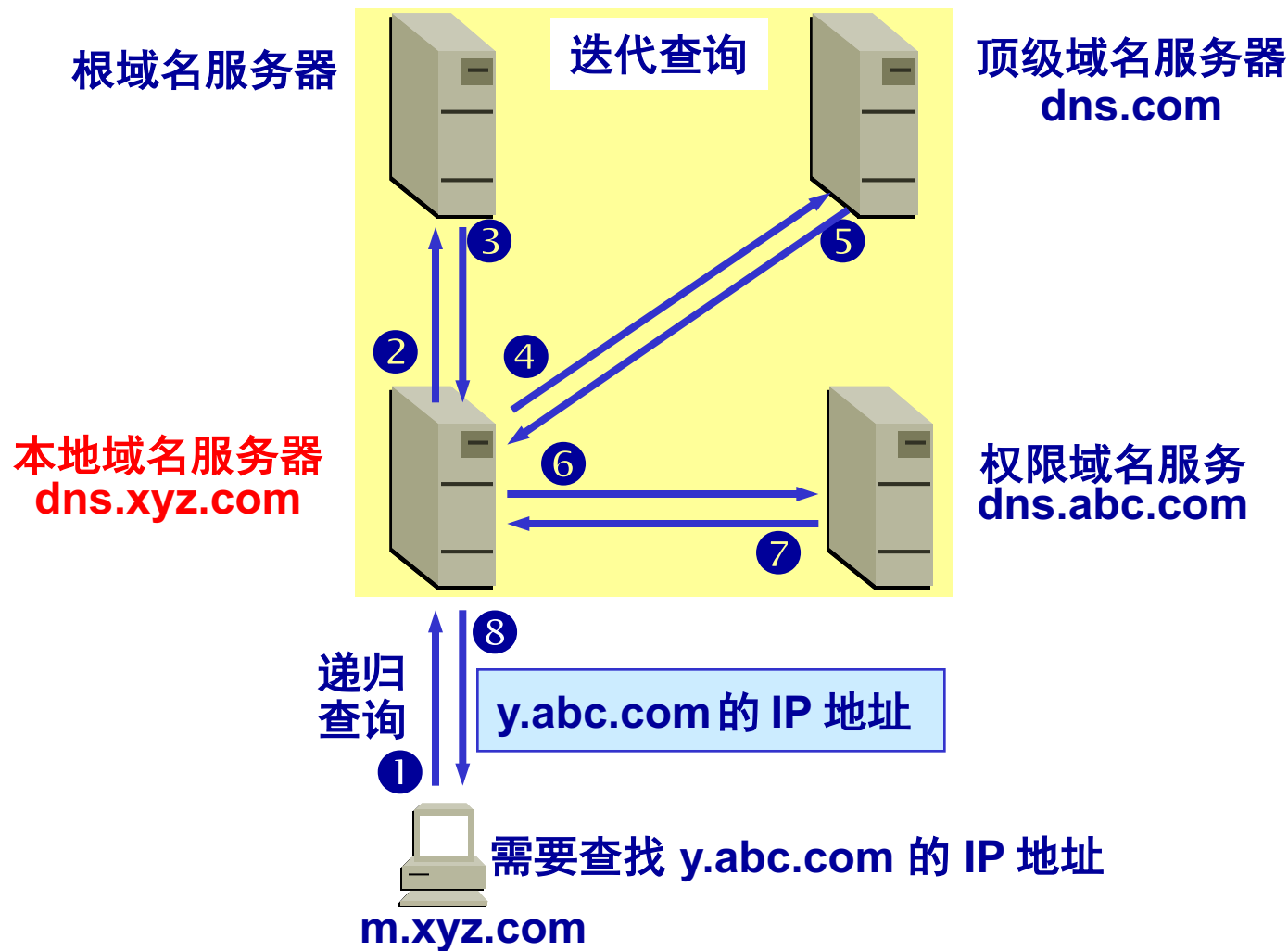
- 本地域名服务器对域名系统非常重要。
- 当一个主机发出 **DNS** 查询请求时，这个查询请求报文就发送给本地域名服务器。
- 每一个互联网服务提供者 **ISP**，或一个大学，甚至一个大学里的系，都可以拥有一个本地域名服务器，
- 这种域名服务器有时也称为默认域名服务器。



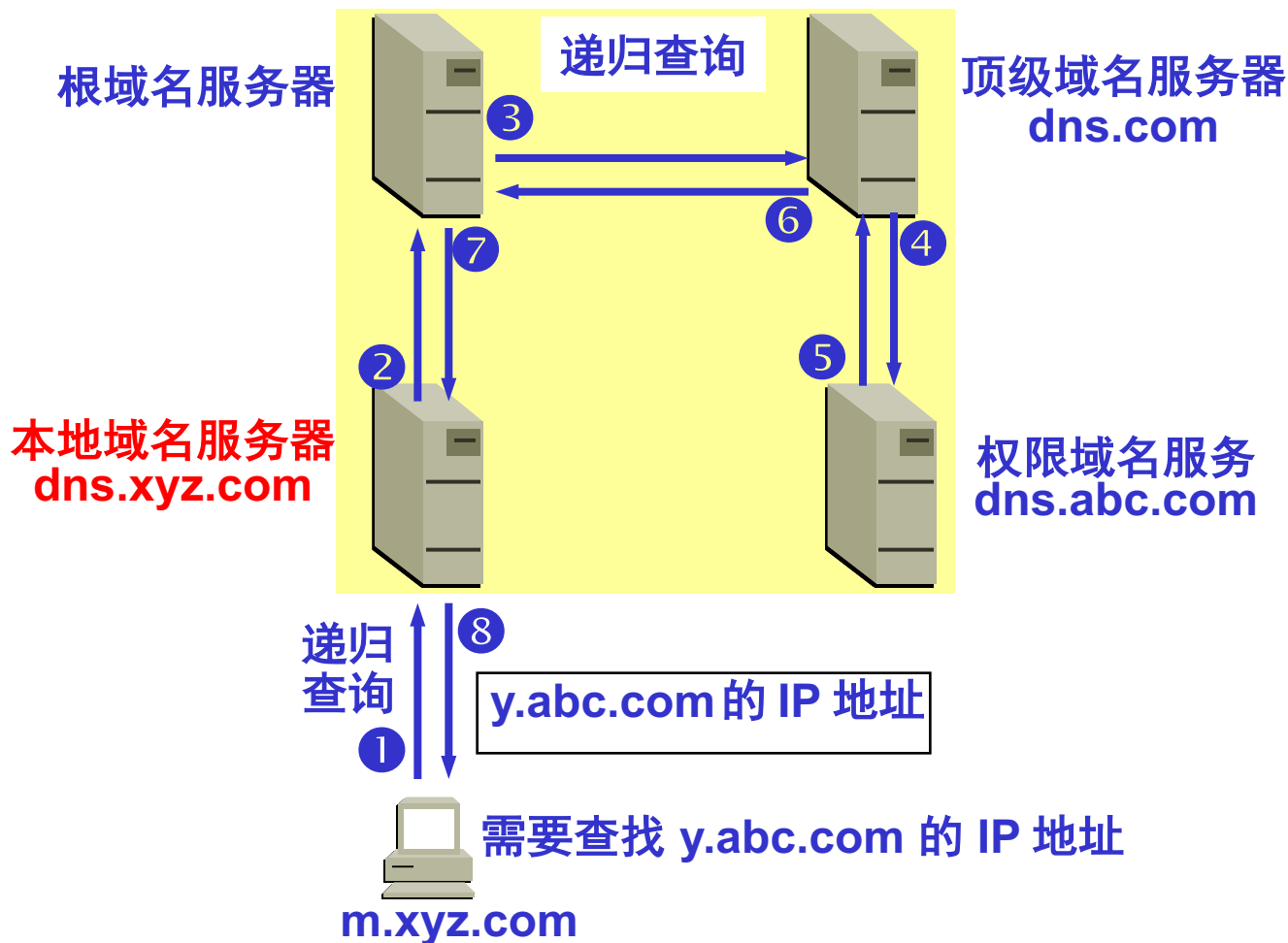
域名解析

- 域名解析：查询一个名字和找出其对应地址的过程
- 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。
- 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。

本地域名服务器采用迭代查询



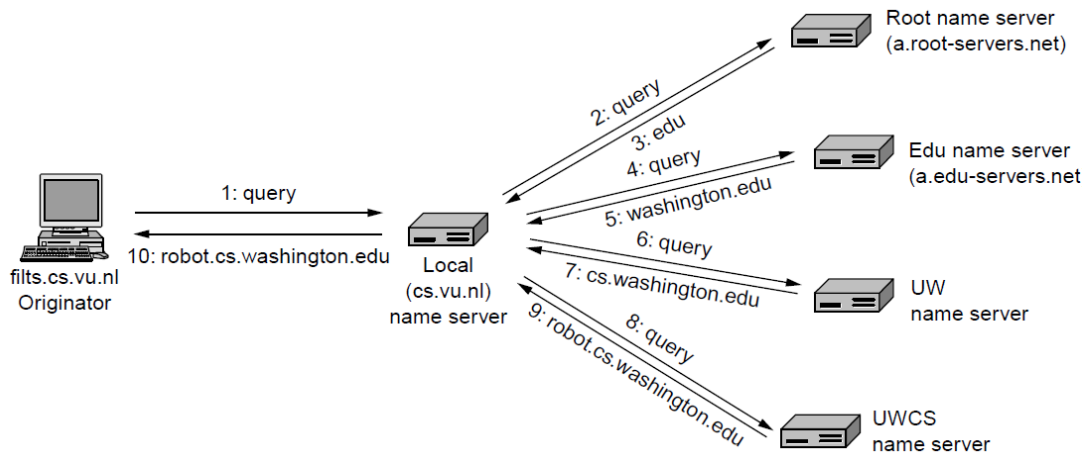
本地域名服务器采用递归查询 (比较少用)





域名服务器

- 域名解析是DNS的主要过程，使用DNS协议
- Cs.vu.nl上的机器fits查询robot.cs.washington.edu地址的过程：
 - Fits发送DNS请求（请求包括被查域名、类型A、类IN）到本地DNS服务器
 - 本地DNS服务器如能够解析这个域名，就返回；否则就问能够解析.edu的服务器在哪？因为.edu是顶级域名，需要问根服务器
 - 根服务器返回.edu的域名服务器地址，本地服务器继续问.edu服务器。Edu服务器本身无法解决这个解析，但它知道它所属的一个域名服务器可以解决，所以返回washington.edu域名服务器
 - 本地服务器继续查询，直到找到能够解析robot.cs.washington.edu主机的域名服务器cs.washington.edu服务器





域名服务器

- DNS 协议:
 - 基于UDP，端口号53，
 - 重传机制：消息丢失，需要重传
- 高速缓存
 - DNS服务器可以缓存DNS的应答消息，以便将来再有相同的请求时，无需再次执行相同的流程，而可以直接回复
 - 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）。



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。



6.2.2 FTP 的基本工作原理

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。



FTP 特点

- 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 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

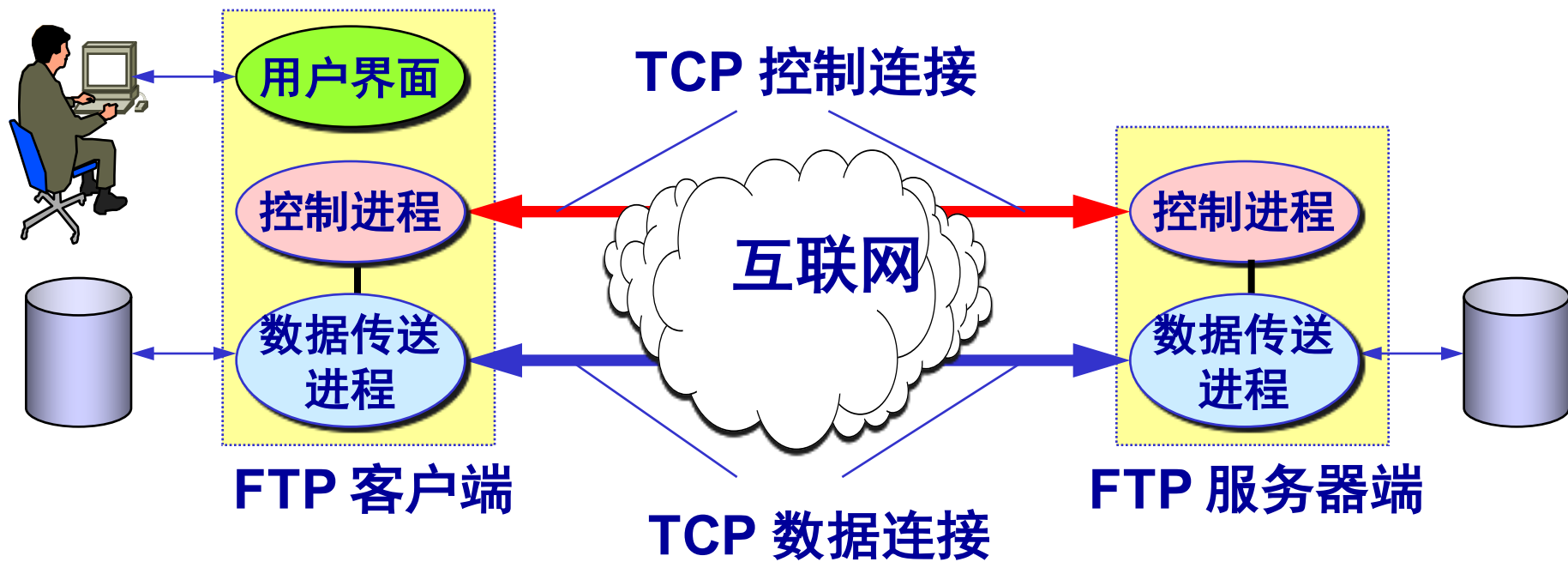
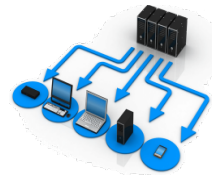
- 打开熟知端口（端口号为 21），使客户进程能够连接上。
- 等待客户进程发出连接请求。
- 启动从属进程来处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。
- 回到等待状态，继续接受其他客户进程发来的请求。主进程与从属进程的处理是并发地进行。



两个连接

- **控制连接**在整个会话期间一直保持打开，FTP 客户发出的传送请求通过控制连接发送给服务器端的控制进程，但控制连接不用来传送文件。
- 实际用于传输文件的是“**数据连接**”。服务器端的控制进程在接收到 FTP 客户发送来的文件传输请求后就创建“数据传送进程”和“数据连接”，用来连接客户端和服务器的数据传送进程。
- 数据传送进程实际完成文件的传送，在传送完毕后关闭“数据传送连接”并结束运行。

FTP 使用的两个 TCP 连接





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）。



6.2.3 简单文件传送协议 TFTP

- TFTP (Trivial File Transfer Protocol) 是一个很小且易于实现的文件传送协议。
- TFTP 使用客户服务器方式和使用 **UDP 数据报**，因此 TFTP **需要自己提供可靠性**。
- TFTP 只支持文件传输而不支持交互。



TFTP 的主要特点

- (1) 每次传送的数据 PDU 中有 512 字节的数据，但最后一次可不足 512 字节。
- (2) 数据 PDU 也称为文件块 (block)，每个块按序编号，从 1 开始。
- (3) 支持 ASCII 码或二进制传送。
- (4) 可对文件进行读或写。
- (5) 使用很简单的首部。



TFTP 的工作很像停止等待协议

- 发送完一个文件块后就等待对方的确认，确认时应指明所确认的块编号。
- 发完数据后在规定时间内收不到确认就要重发数据 **PDU**。
- 发送确认 **PDU** 的一方若在规定时间内收不到下一个文件块，也要重发确认 **PDU**。这样就可保证文件的传送不致因某一个数据报的丢失而告失败。？



TFTP 的工作很像停止等待协议

- 开始工作时，TFTP 客户进程发送一个读请求 PDU 或写请求 PDU 给 TFTP 服务器进程，其熟知端口号码为 69。
- TFTP 服务器进程要选择一个新的端口和 TFTP 客户进程进行通信。
- 若文件长度恰好为 512 字节的整数倍，则在文件传送完毕后，还必须在最后发送一个只含首部而无数据的数据 PDU。
- 若文件长度不是 512 字节的整数倍，则最后传送数据 PDU 的数据字段一定不满 512 字节，这正好可作为文件结束的标志。



6.3 远程终端协议 TELNET

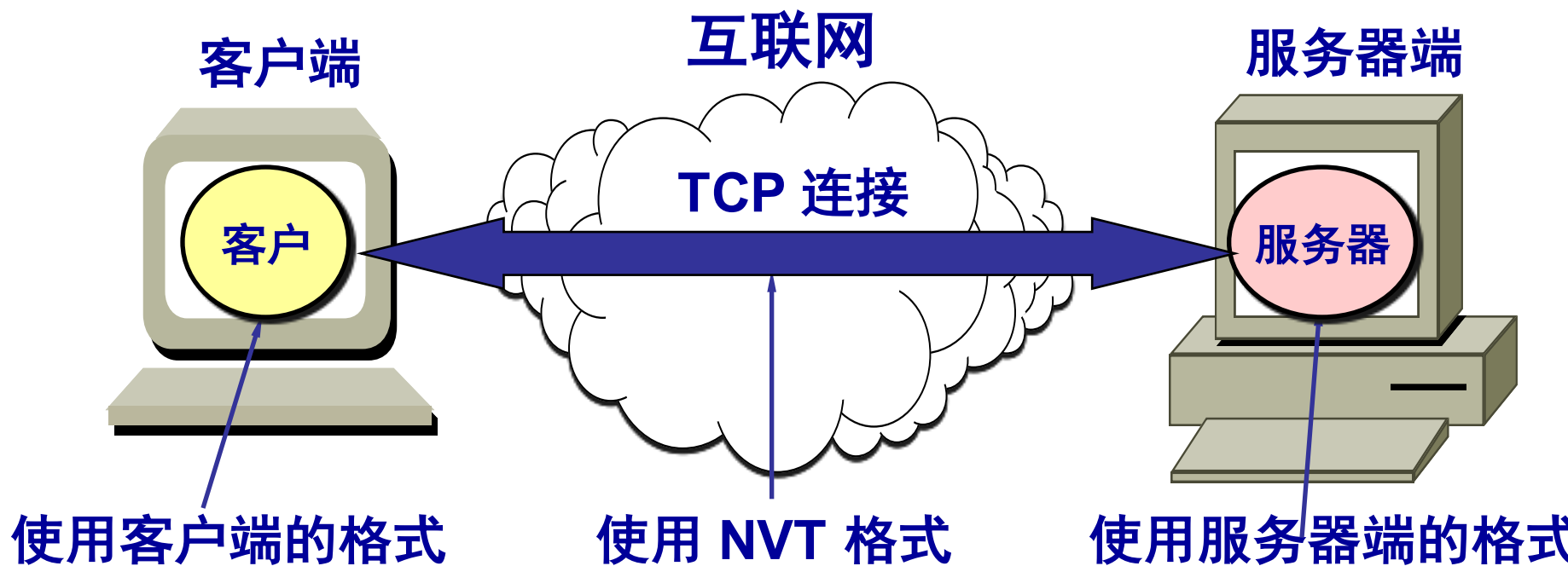
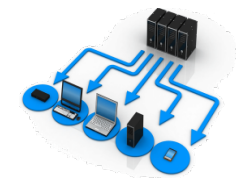
- TELNET 是一个简单的远程终端协议，也是互联网的正式标准。
- 用户用 TELNET 就可在其所在地通过 TCP 连接注册（即登录）到远地的另一个主机上（使用主机名或 IP 地址）。
- TELNET 能将用户的击键传到远地主机，同时也能将远地主机的输出通过 TCP 连接返回到用户屏幕。这种服务是透明的，因为用户感觉到好像键盘和显示器是直接连在远地主机上。



客户-服务器方式

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

TELNET 使用 网络虚拟终端 NVT 格式





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。



6.4 万维网 WWW

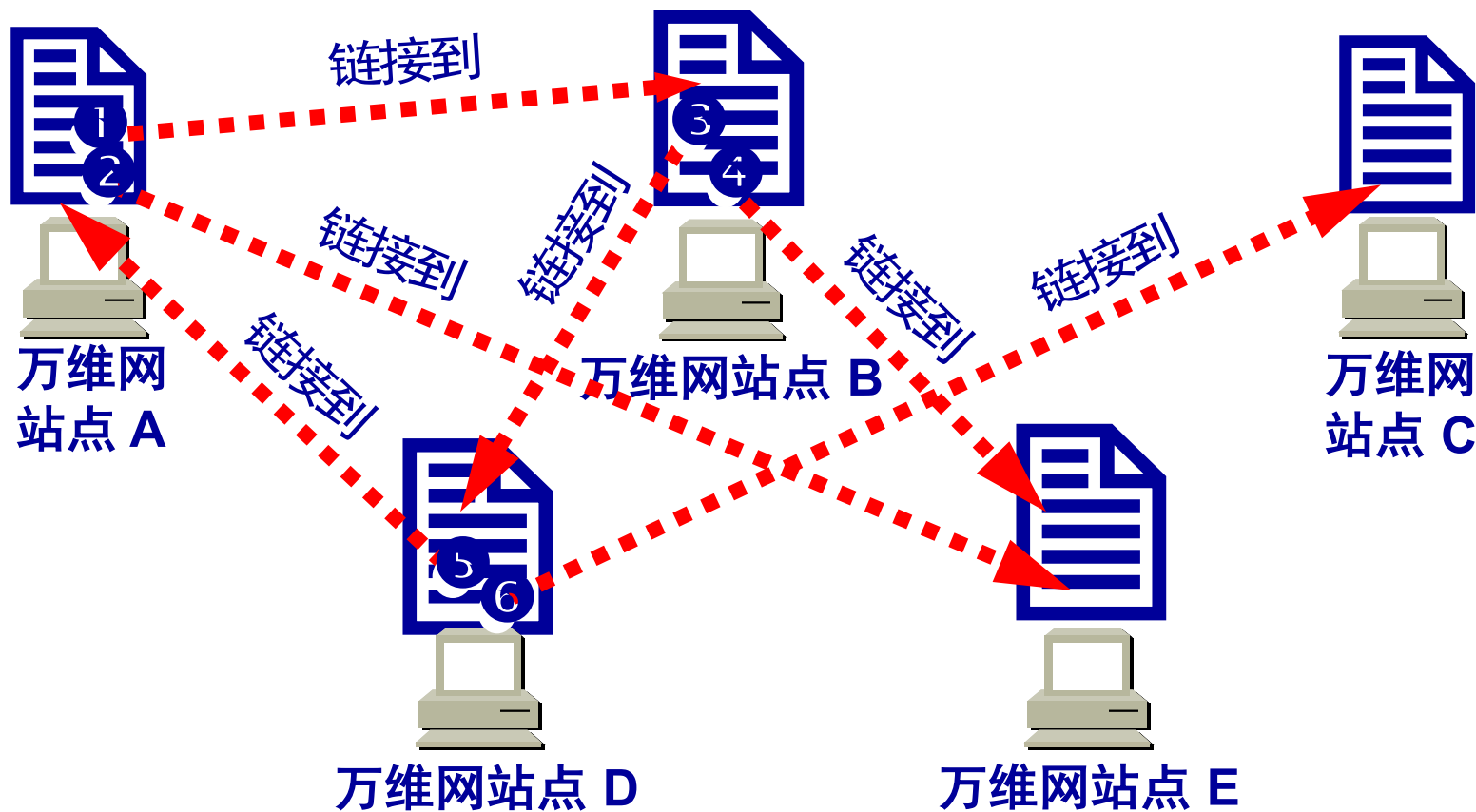
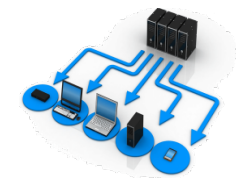
- 6.4.1 万维网概述
- 6.4.2 统一资源定位符 URL
- 6.4.3 超文本传送协议 HTTP
- 6.4.4 万维网的文档
- 6.4.5 万维网的信息检索系统
- 6.4.6 博客和微博
- 6.4.7 社交网络



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内容具有唯一的标识符 **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

- 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



2. 使用 HTTP 的 URL

- 使用 HTTP 的 URL 的一般形式

`http://<主机>:<端口>/<路径>`



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。

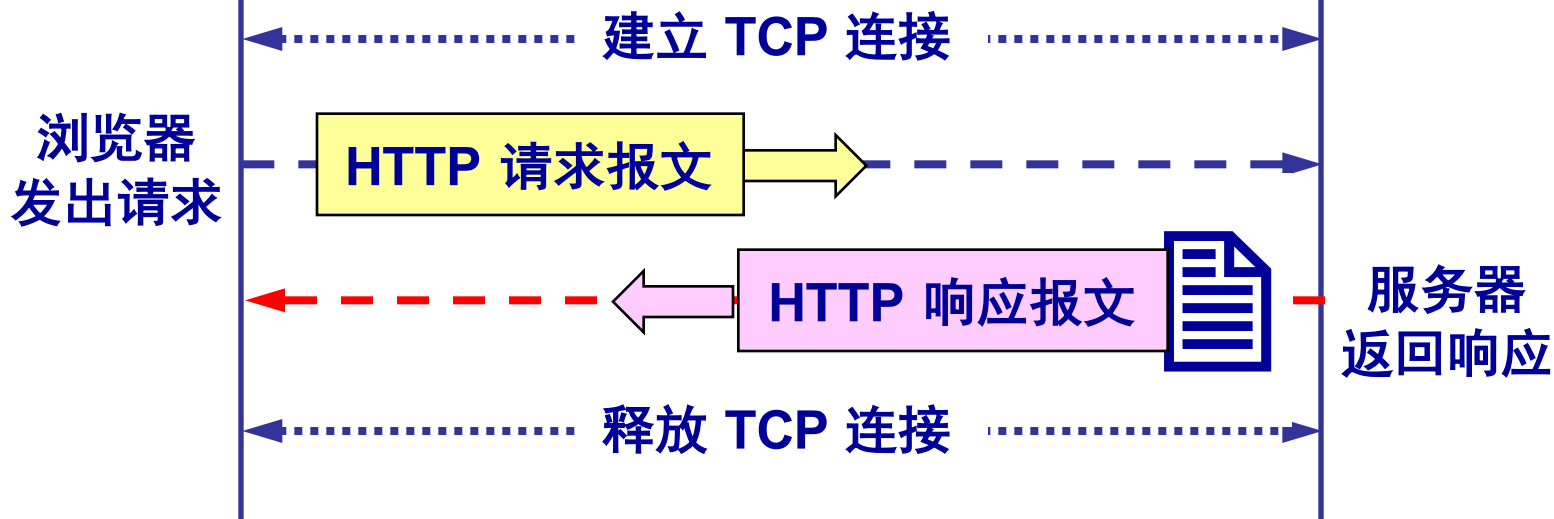
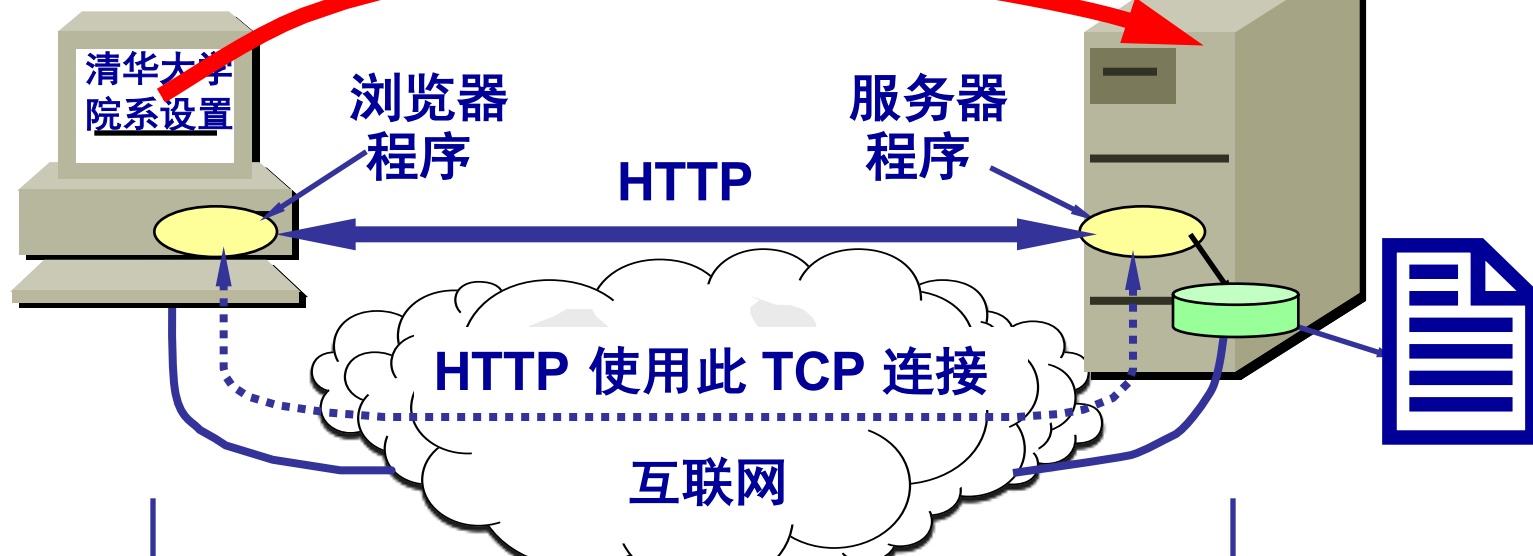
万维网的工作过程



服务器

链接到URL的超链 www.tsinghua.edu.cn

客户





用户点击 URL

<http://www.tsinghua.edu.cn/chn/yxsx/index.htm>

后所发生的事件

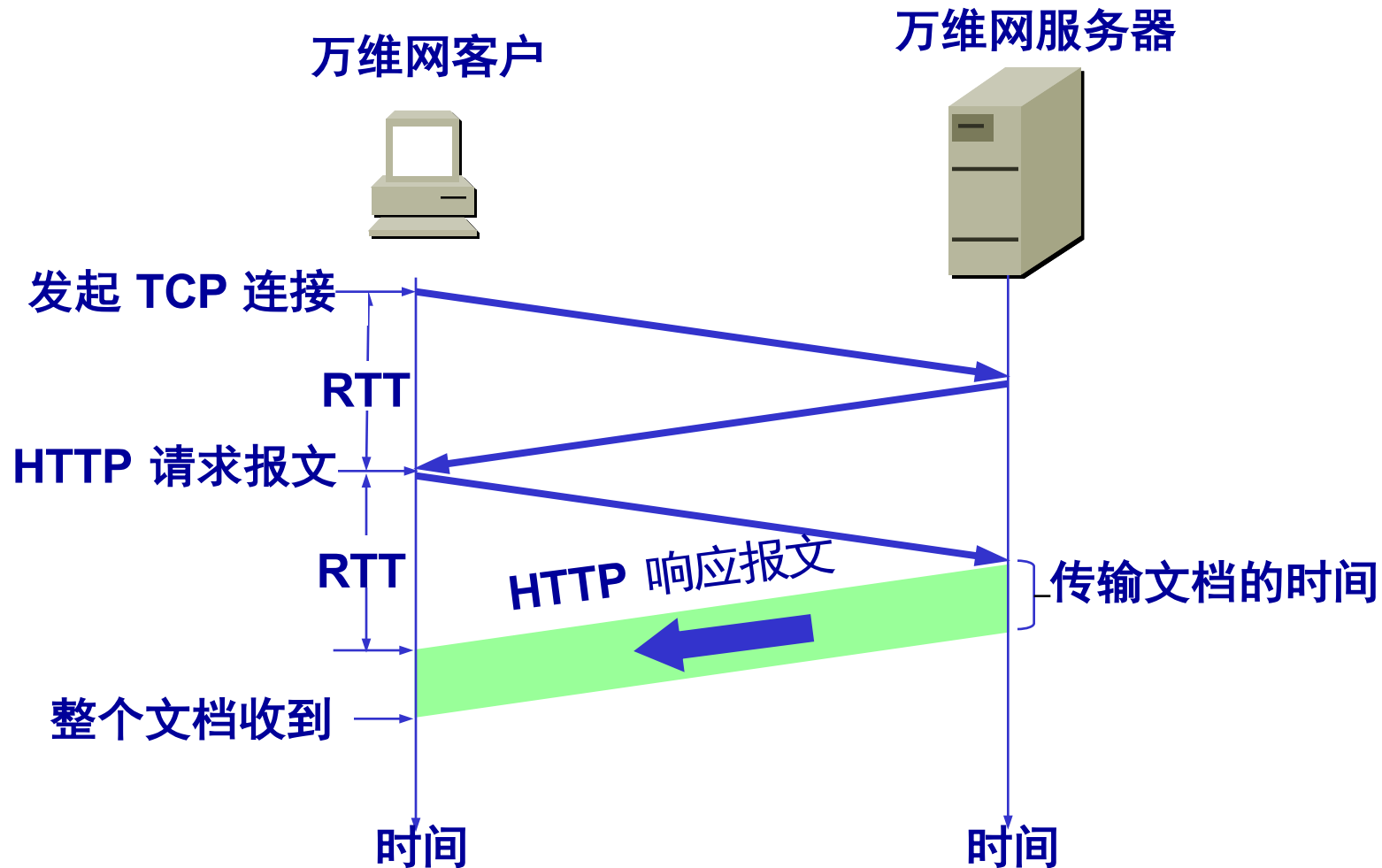
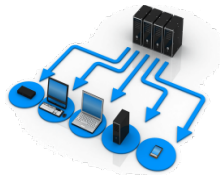
- (1) 浏览器分析超链指向页面的 URL。
- (2) 浏览器向 DNS 请求解析 www.tsinghua.edu.cn 的 IP 地址。
- (3) 域名系统 DNS 解析出清华大学服务器的 IP 地址。
- (4) 浏览器与服务器建立 TCP 连接。
- (5) 浏览器发出取文件命令：GET /chn/yxsx/index.htm。
- (6) 服务器给出响应，把文件 index.htm 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示“清华大学院系设置”文件 index.htm 中的所有文本。



HTTP 的主要特点

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

请求一个万维网文档所需的时间





持续连接

- 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。



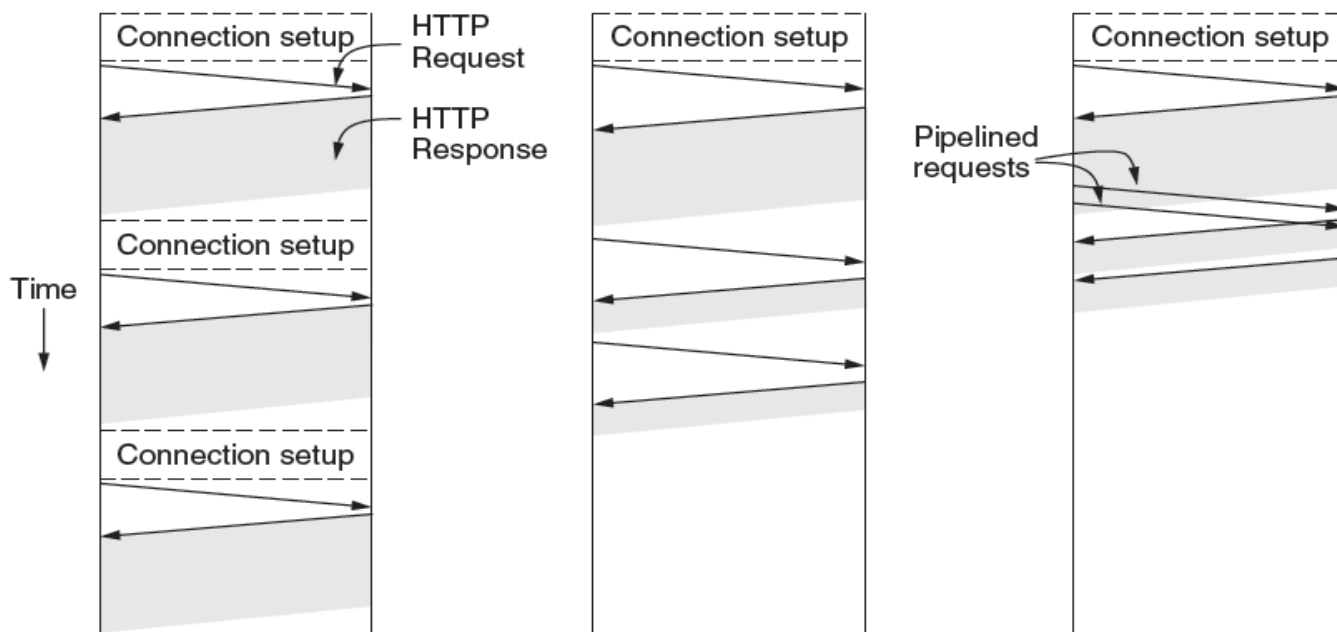
持续连接的两种工作方式

- **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。



HTTP

- 在发送HTTP请求之前，需要建立TCP连接
- 三种HTTP请求：如图



一个请求一个连接

非流水线方式

流水线方式



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。

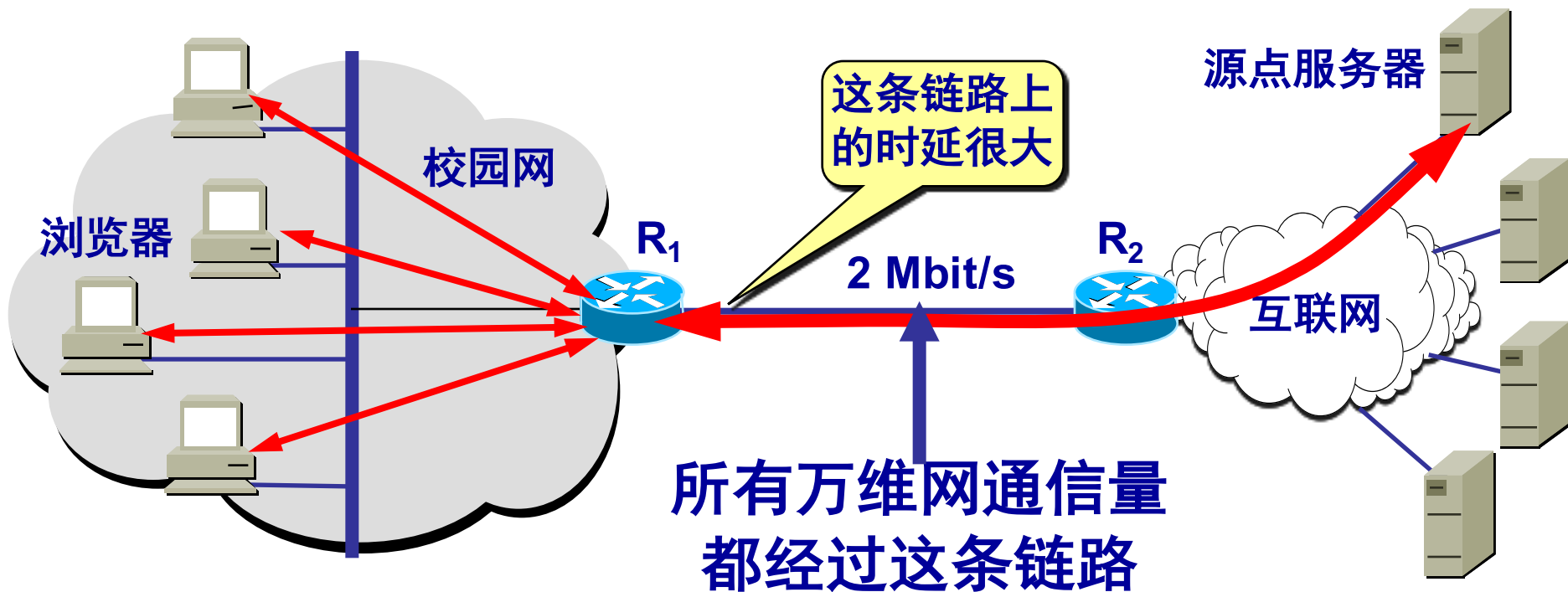


武汉大学

WUHAN UNIVERSITY

使用高速缓存可减少 访问互联网服务器的时延

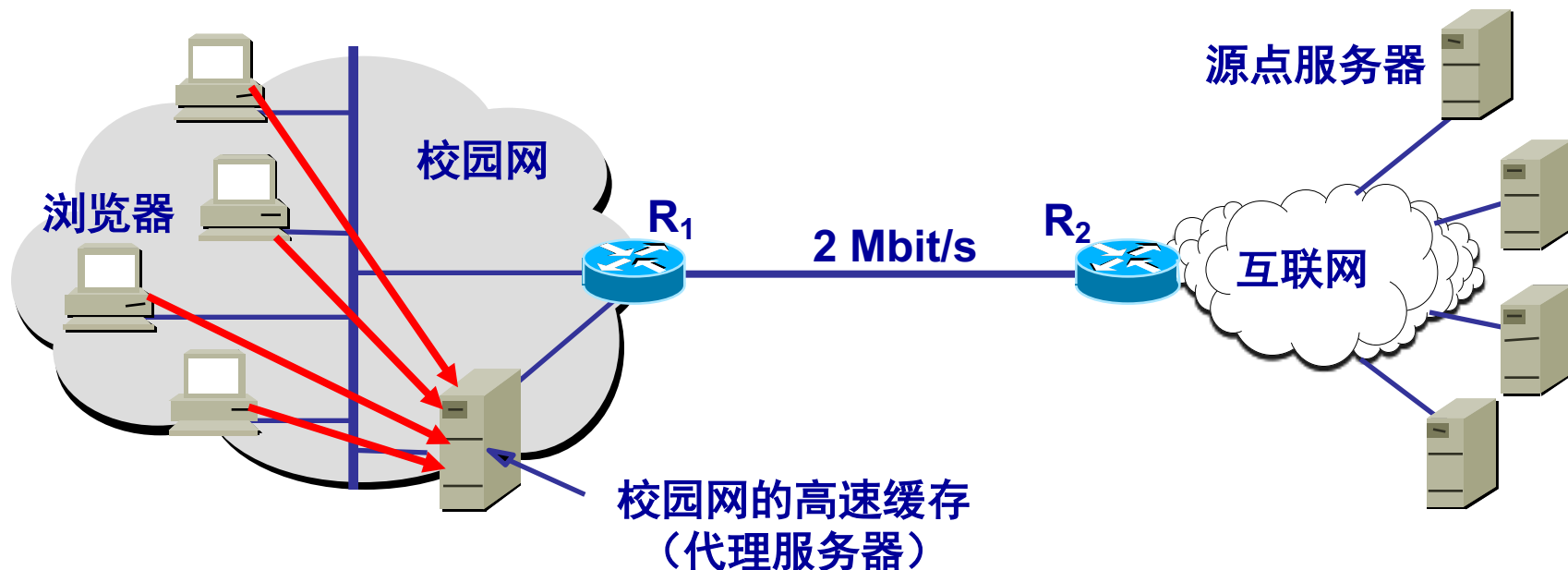
没有使用高速缓存的情况





使用高速缓存的情况

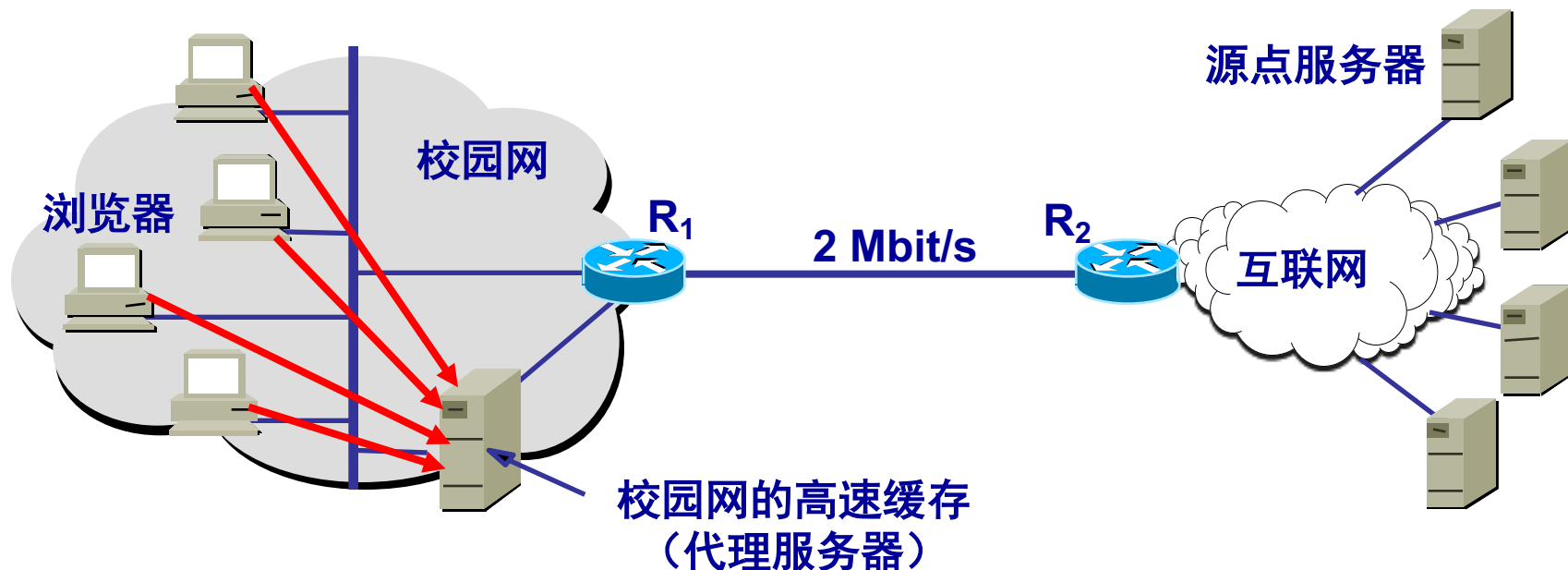
(1) 浏览器访问互联网的服务器时，要先与校园网的高速缓存建立 TCP 连接，并向高速缓存发出 HTTP 请求报文。



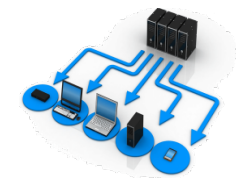


使用高速缓存的情况

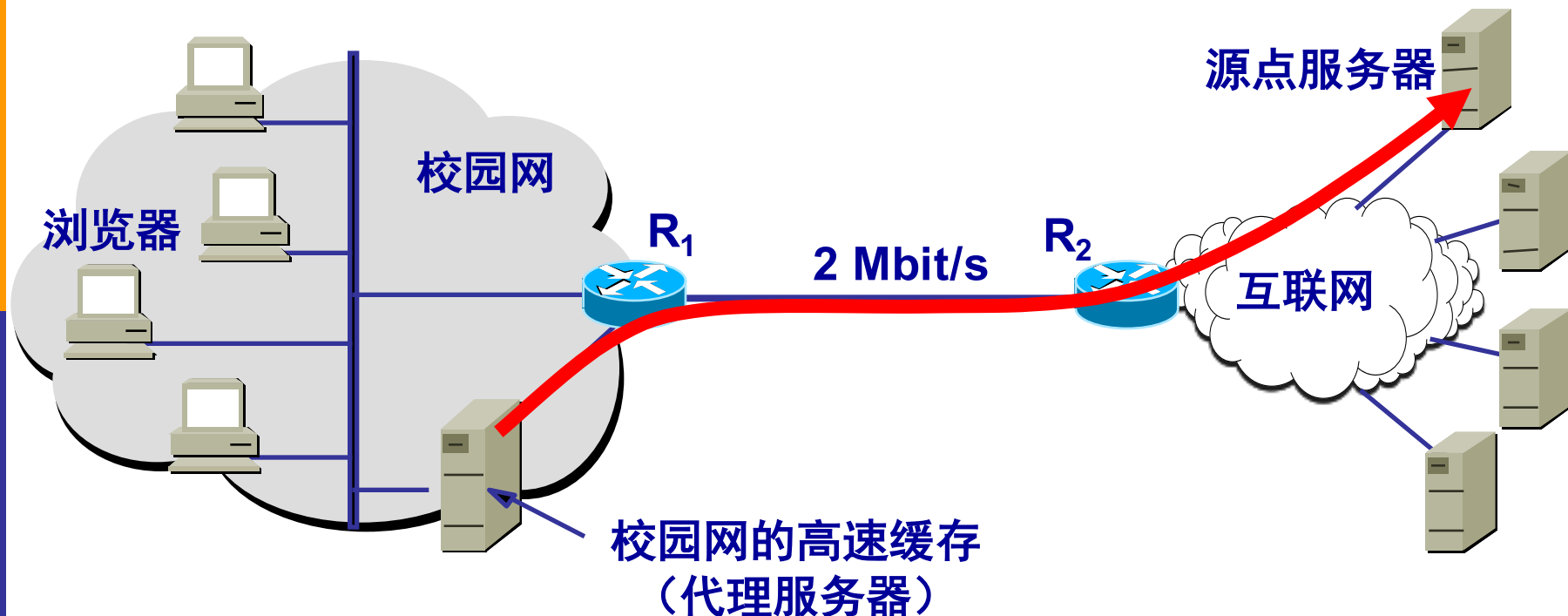
(2) 若高速缓存已经存放了所请求的对象，则将此对象放入 HTTP 响应报文中返回给浏览器。



使用高速缓存的情况



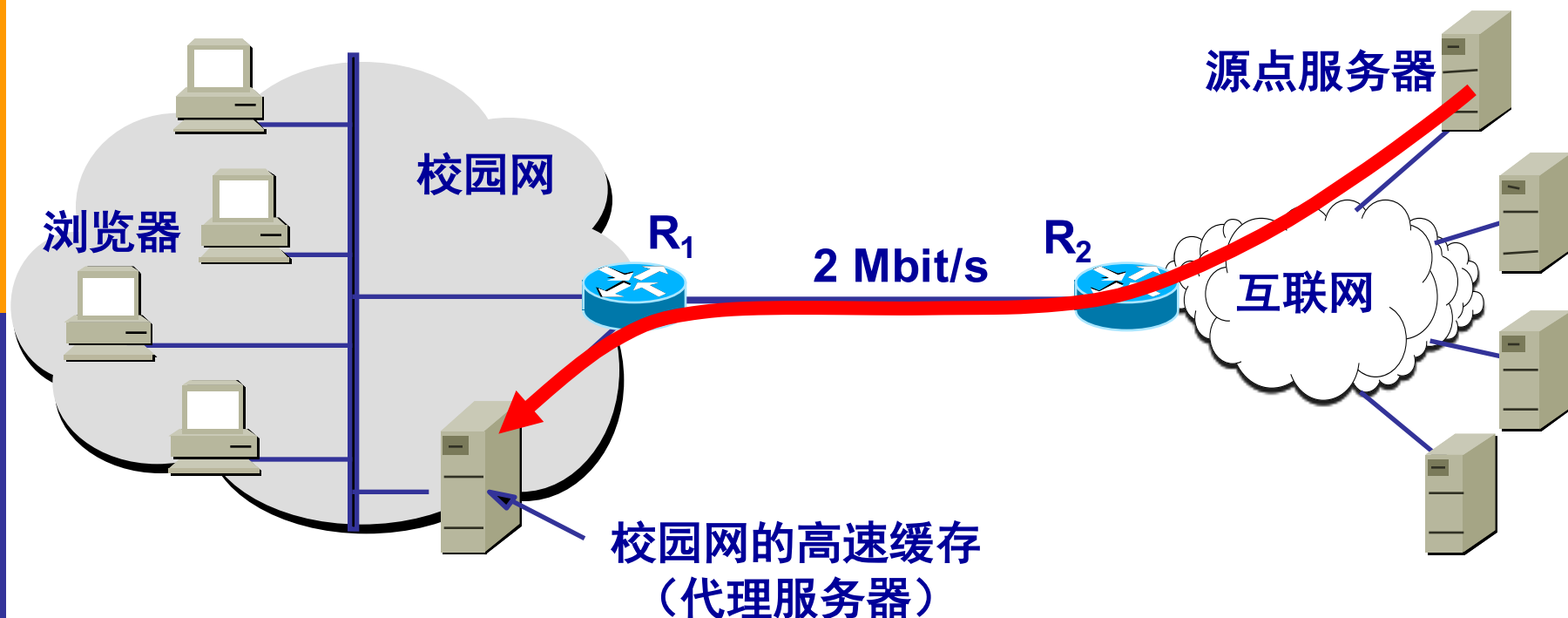
(3) 否则，高速缓存就代表发出请求的用户浏览器，与互联网上的源点服务器建立 TCP 连接，并发送 HTTP 请求报文。



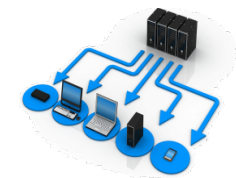
使用高速缓存的情况



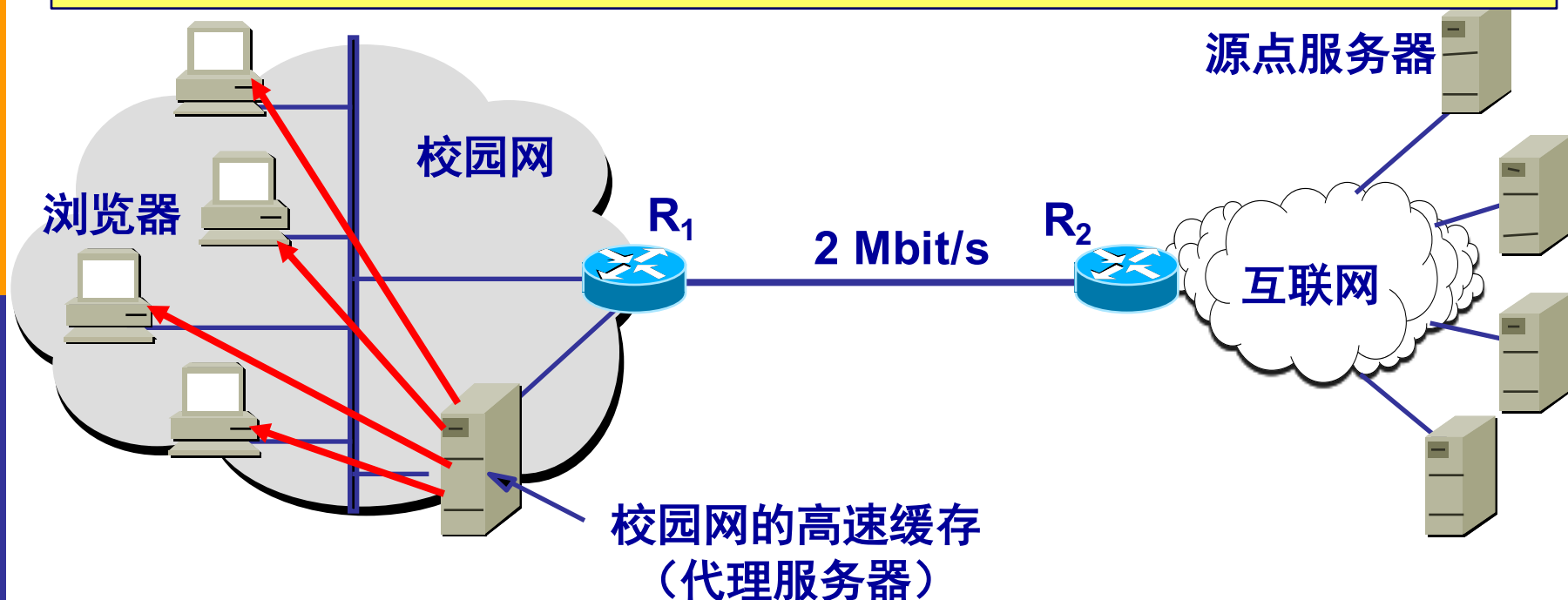
(4) 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存。



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。



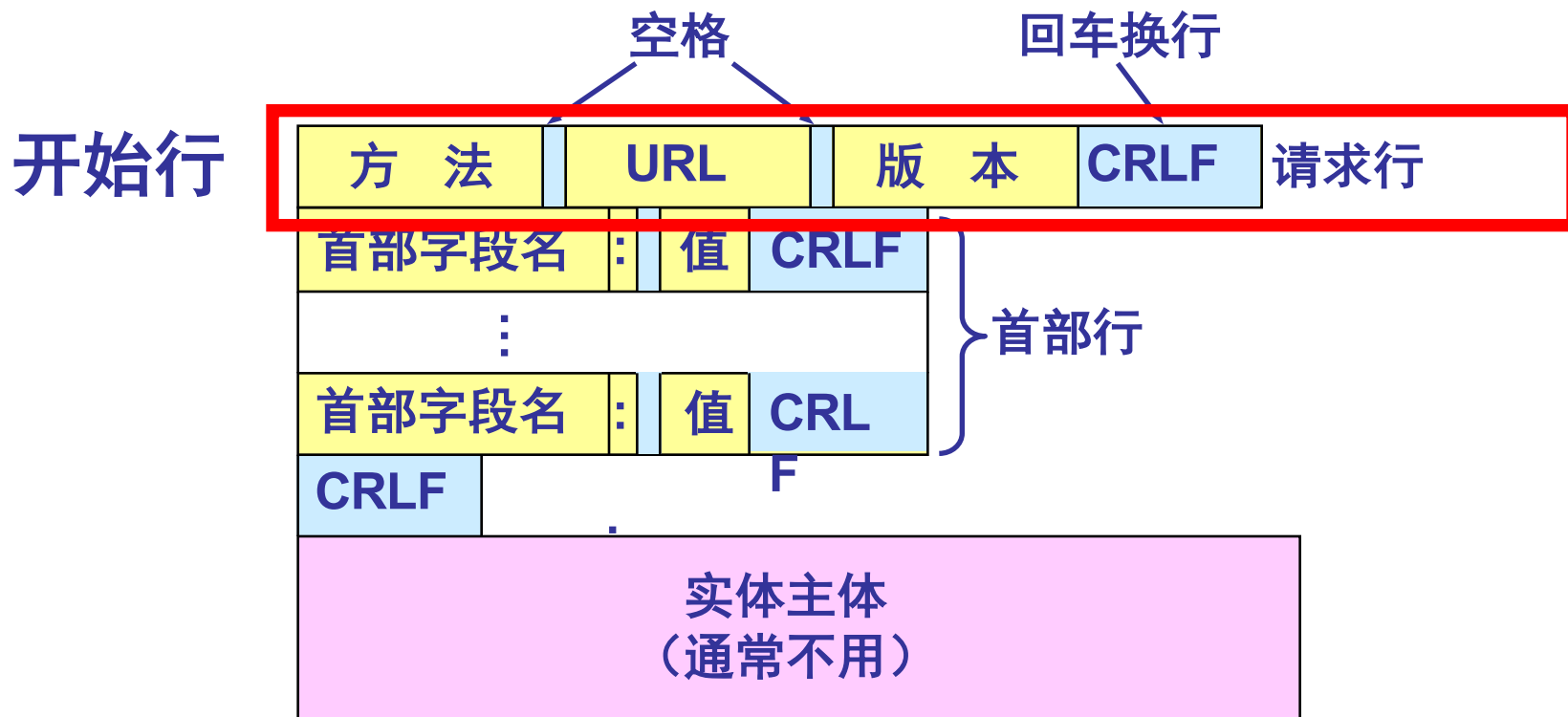


3. HTTP 的报文结构

HTTP 有两类报文：

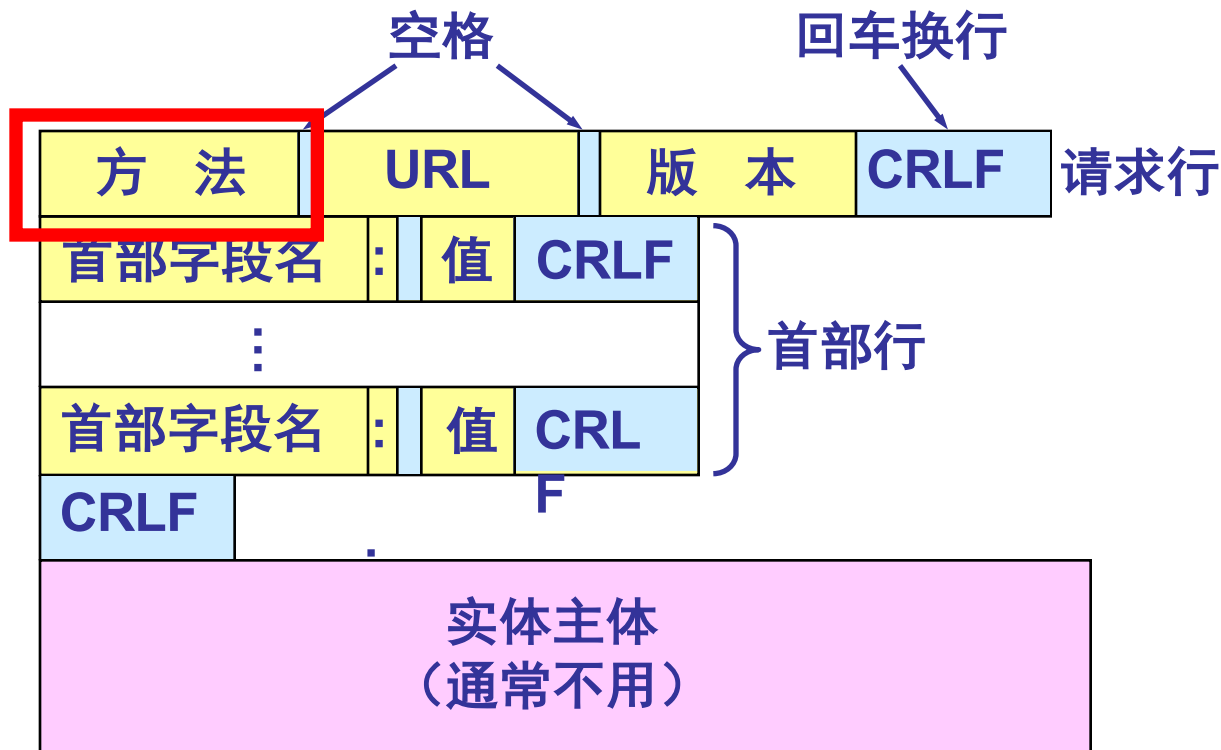
- **请求报文**——从客户向服务器发送请求报文。
- **响应报文**——从服务器到客户的回答。
- 由于 HTTP 是面向正文的 (text-oriented), 因此在报文中的每一个字段都是一些 **ASCII 码串**, 因而每个字段的长度都是不确定的。

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）



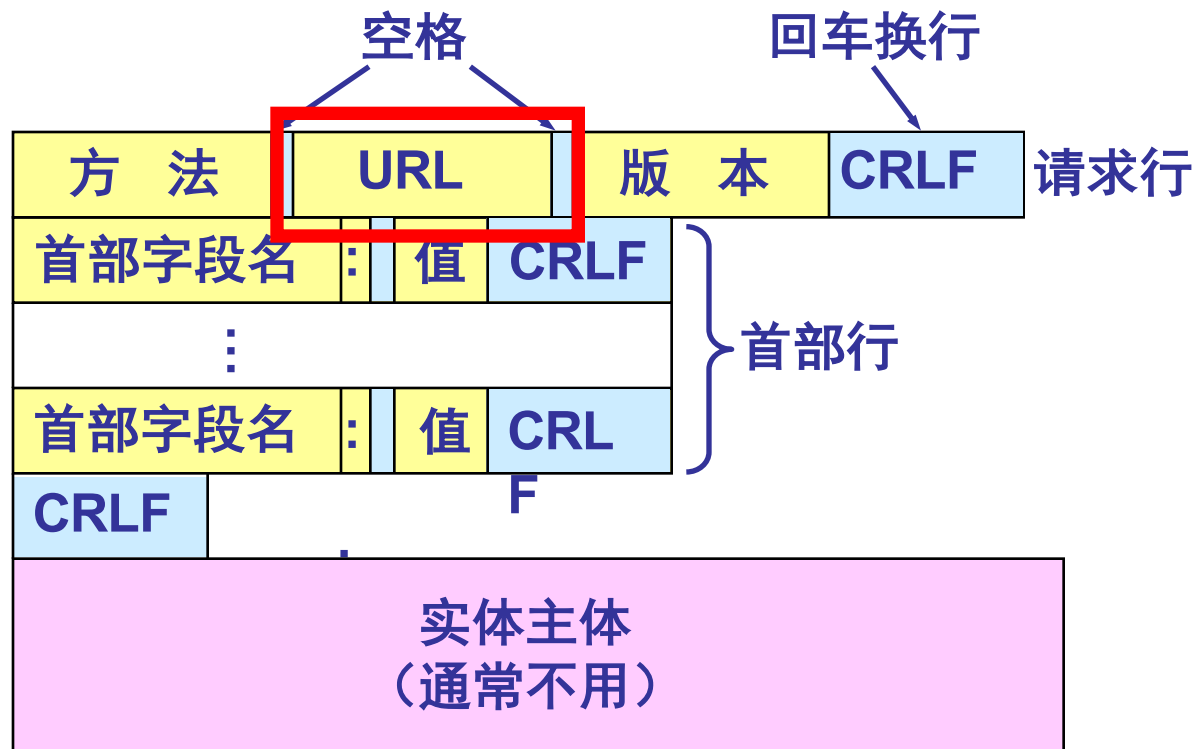
“方法” 是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的。

HTTP 请求报文的一些方法



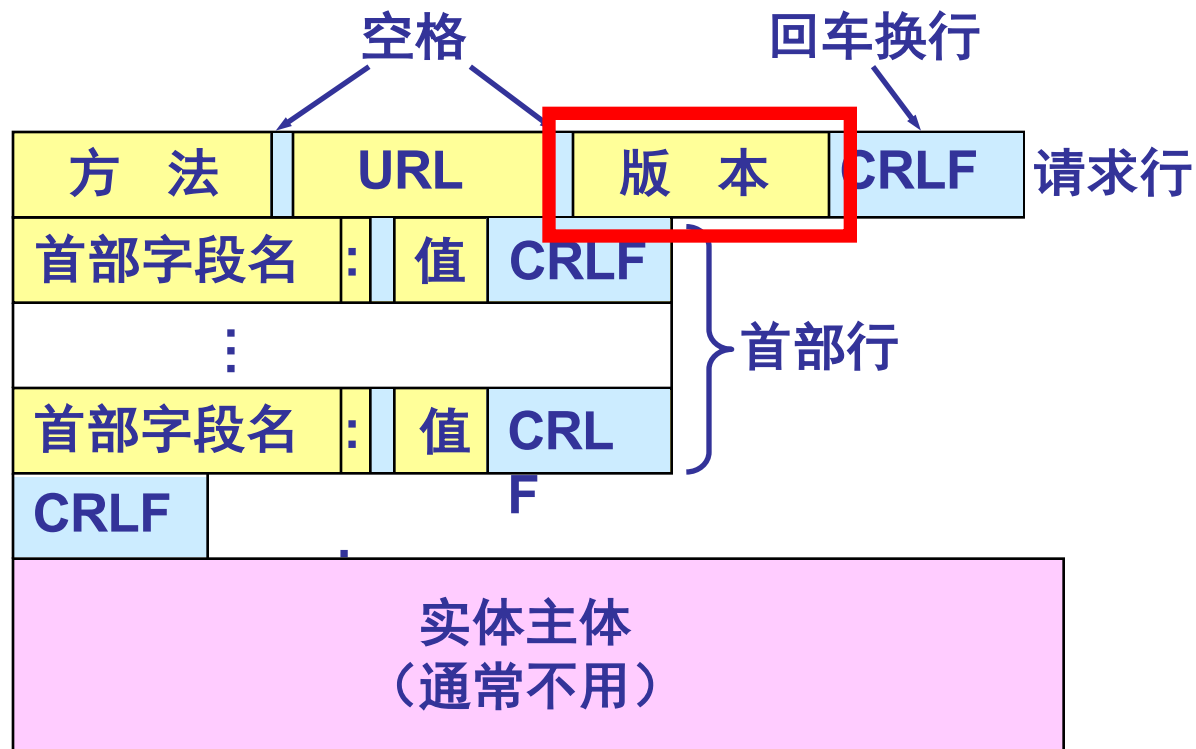
方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL 所标志的信息
HEAD 首部	请求读取由 URL 所标志的信息的
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL 下存储一个文档
DELETE	删除指明的 URL 所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

HTTP 的报文结构（请求报文）



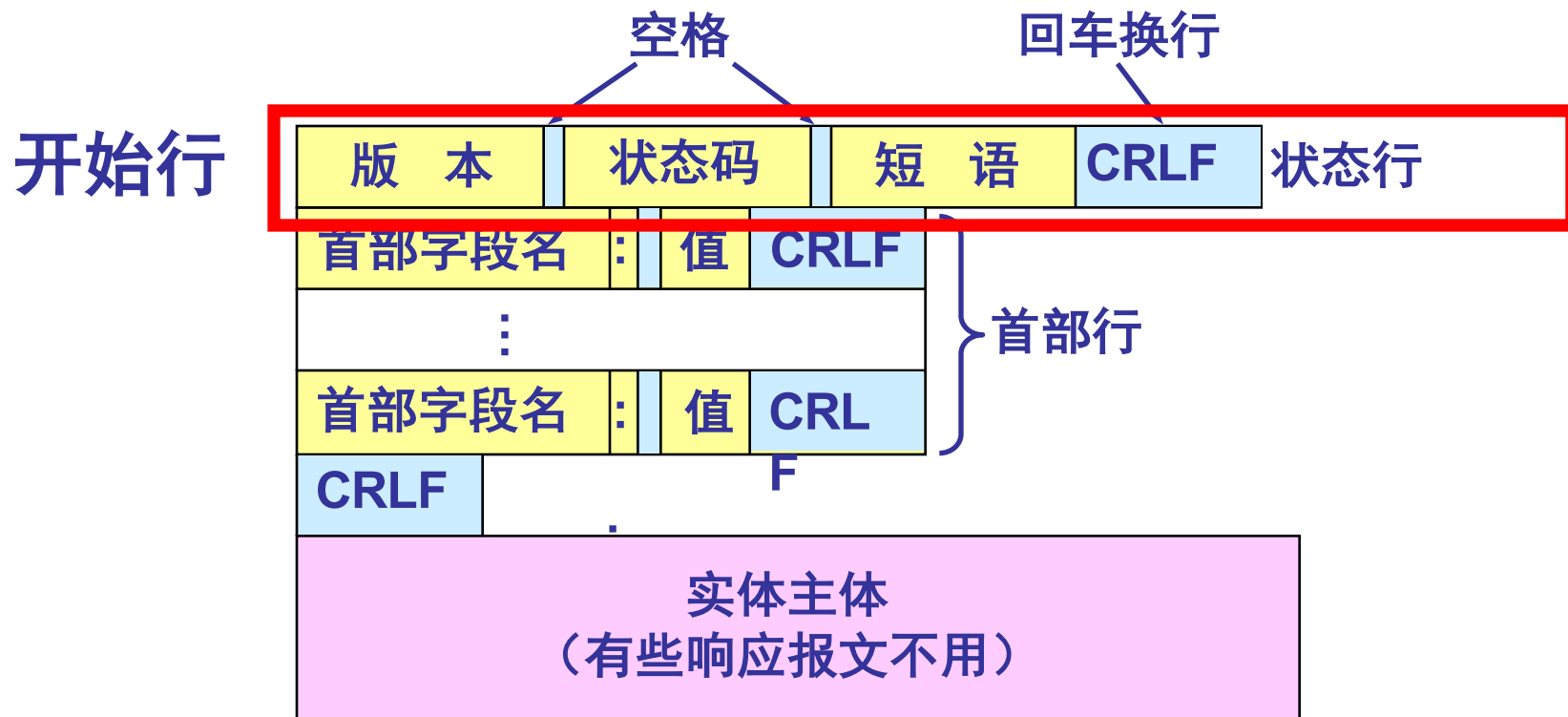
“URL”是所请求的资源的 URL。

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。

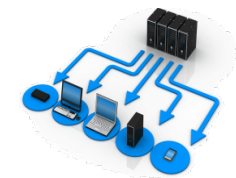


6.4.4 万维网的文档

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（ASCII 码文件）。
- .html 或 .htm 为后缀

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>



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。



XHTML

- 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

- 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。



2. 动态万维网文档

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



6.5 电子邮件

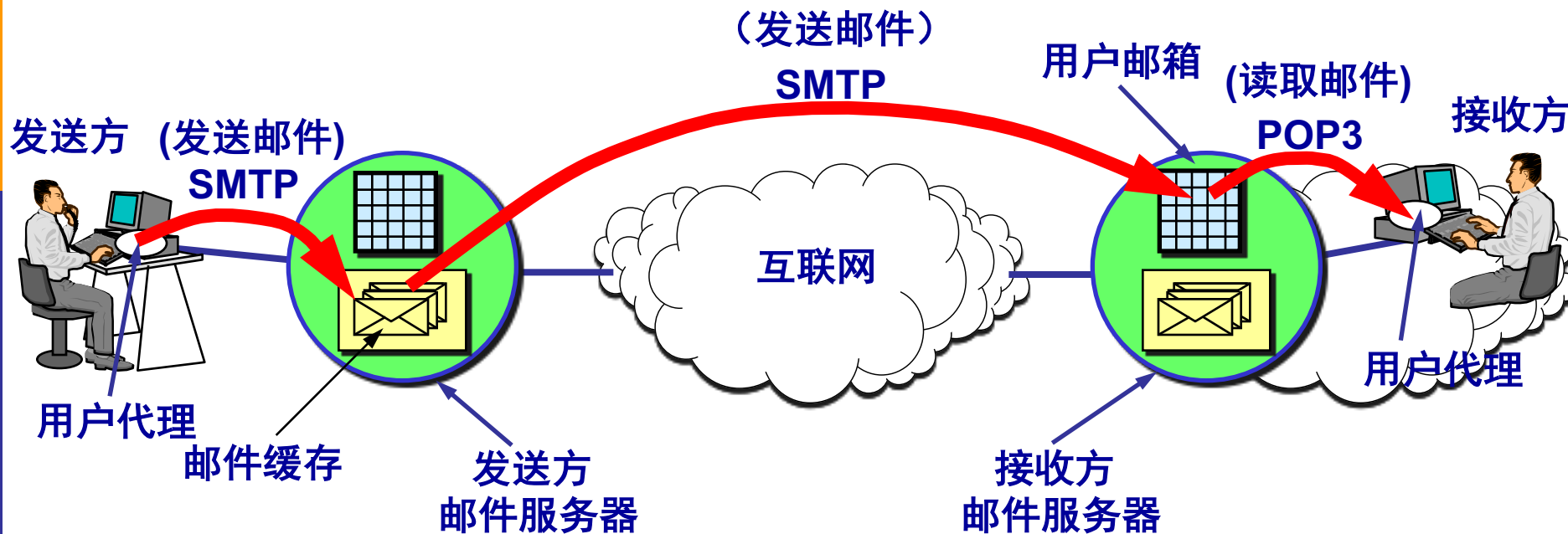
- 6.5.1 电子邮件概述
- 6.5.2 简单邮件传送协议 **SMTP**
- 6.5.3 电子邮件的信息格式
- 6.5.4 邮件读取协议 **POP3** 和 **IMAP**
- 6.5.5 基于万维网的电子邮件
- 6.5.6 通用互联网邮件扩充 **MIME**



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。



SMTP

- SMTP 使用**客户服务器**方式
- 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。



邮件读取协议POP3 和 IMAP

■ POP3协议

- 允许电子邮件客户端下载服务器上的邮件，
- 但是在客户端的操作（如移动邮件、标记已读等），不会反馈到服务器上，比如通过客户端收取了邮箱中的3封邮件并移动到其他文件夹，邮箱服务器上的这些邮件是没有同时被移动的。



邮件读取协议POP3 和 IMAP

■ IMAP

- 提供webmail 与电子邮件客户端之间的双向通信，客户端的操作都会反馈到服务器上，对邮件进行的操作，服务器上的邮件也会做相应的动作。
- 同时，IMAP和POP3都提供了方便的邮件下载服务，让用户能进行离线阅读。



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。



邮件内容的首部

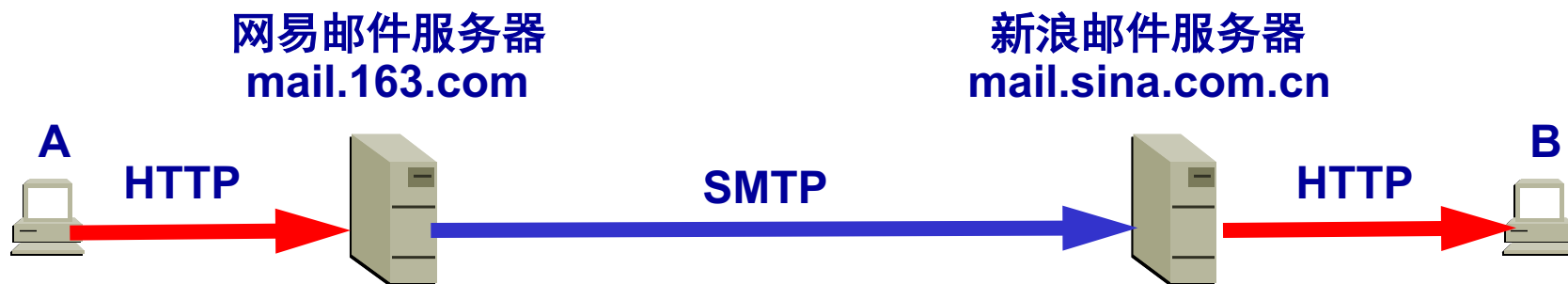
- **"To:"** 后面填入一个或多个收件人的电子邮件地址。用户只需打开地址簿，点击收件人名字，收件人的电子邮件地址就会自动地填入到合适的位置上。
- **"Subject:"** 是邮件的主题。它反映了邮件的主要内容，便于用户查找邮件。
- **"Cc:"** 表示应给某某人发送一个邮件副本。
- **"From"** 和 **"Date"** 表示发信人的电子邮件地址和发信日期。
- **"Reply-To"** 是对方回信所用的地址。



6.5.5 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

- 为什么需要**MIME**：因为邮件可携带的内容不仅仅是文本，还可以是图片、视频等，**MIME**用于描述这些不同的内容。
- **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。



邮件

邮件例子:

- 包含HTML和audio的一个邮件

One part
(HTML)

Another
(audio)

From: alice@cs.washington.edu
To: bob@ee.uwa.edu.au
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@cs.washington.edu>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/html

<p>Happy birthday to you

Happy birthday to you

Happy birthday dear Bob

Happy birthday to you</p>

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
access-type="anon-ftp";
site="bicycle.cs.washington.edu";
directory="pub";
name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--



6.6 动态主机配置协议 DHCP

- 主机需要配置什么项目
 - (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
- 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

1. 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，客户端）。
 - 原地址0.0.0.0；目的地址:255.255.255.255
2. 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(MAC地址)。
 - 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）。
 - 主机的以太网地址作为目的地址



DHCP 中继代理(relay agent)

- 如果网络没有DHCP服务器，则需要一个DHCP 中继代理
 - 配置了 DHCP 服务器的 IP 地址信息。
 - 以单播方式向 DHCP 服务器转发客户端的DHCP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。

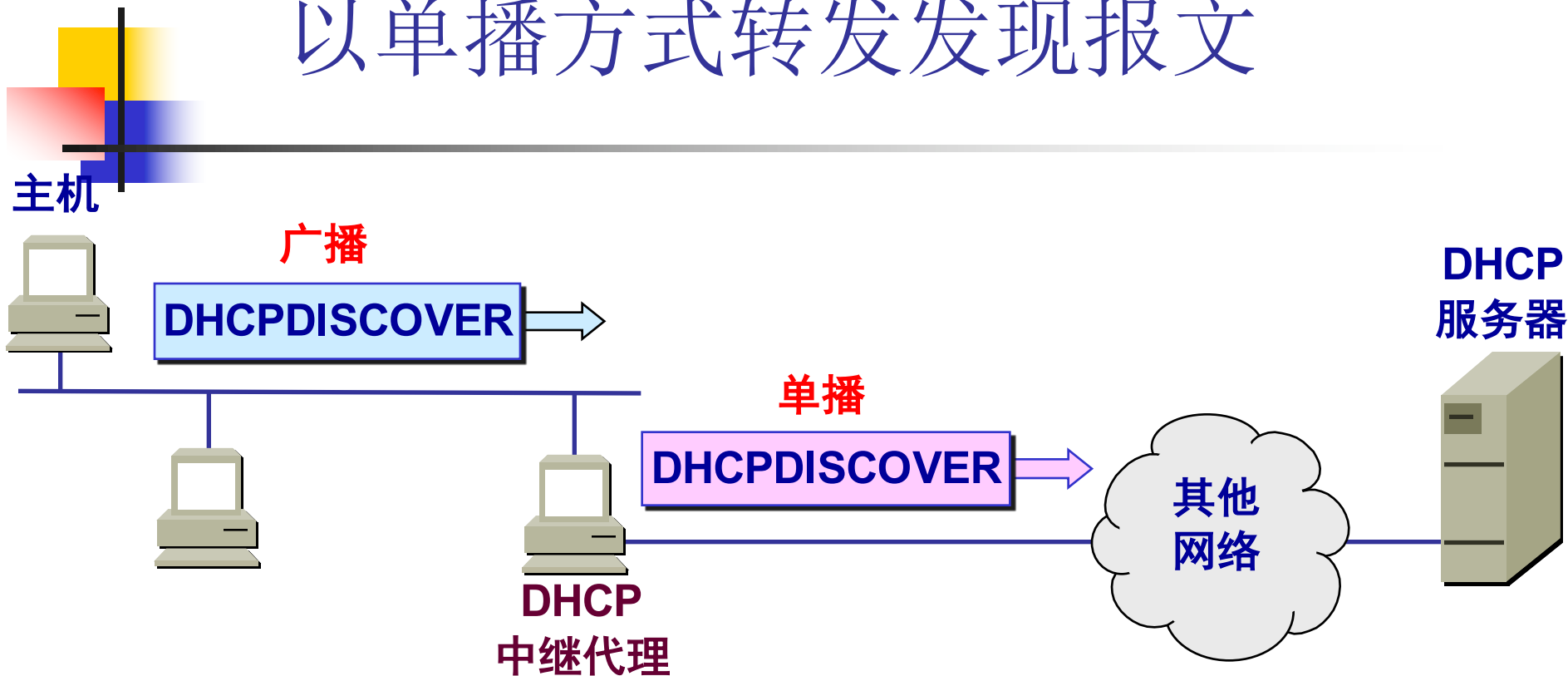


武汉大学

WUHAN UNIVERSITY

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。



DHCP 使用客户-服务器方式

3. 客户端选择一个DHCP服务（可能收到多个），并向其发送DHCP request
 - 宣告自己所选的DHCP
4. DHCP 服务器确认
 - 回复DHCPack
 - 包含了所有的网络参数



租用期 (lease period)

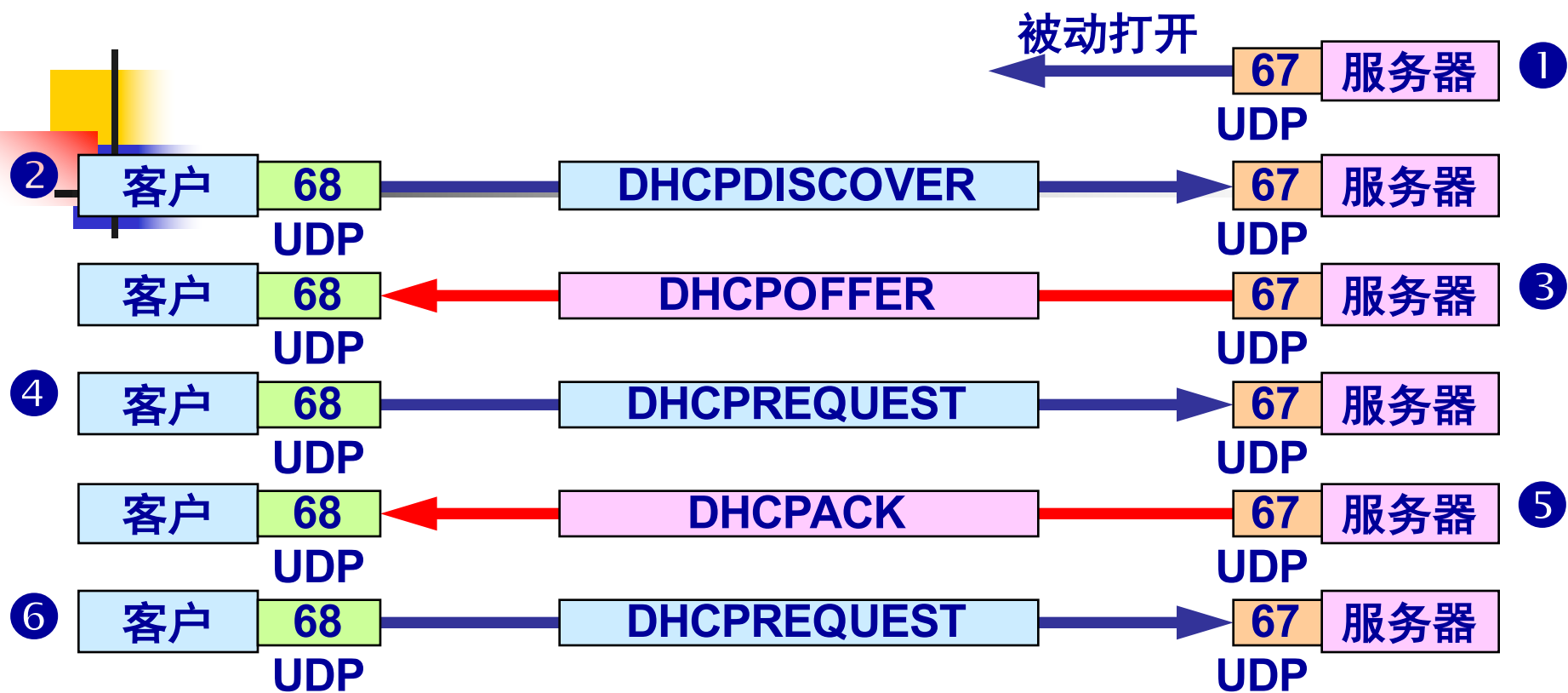
- DHCP 服务器分配给 DHCP 客户的 IP 地址的**临时的**，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。DHCP 协议称这段时间为**租用期**。
- 租用期的数值应由 DHCP 服务器自己决定。
- DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。



武汉大学

WUHAN UNIVERSITY

DHCP 协议的工作过程



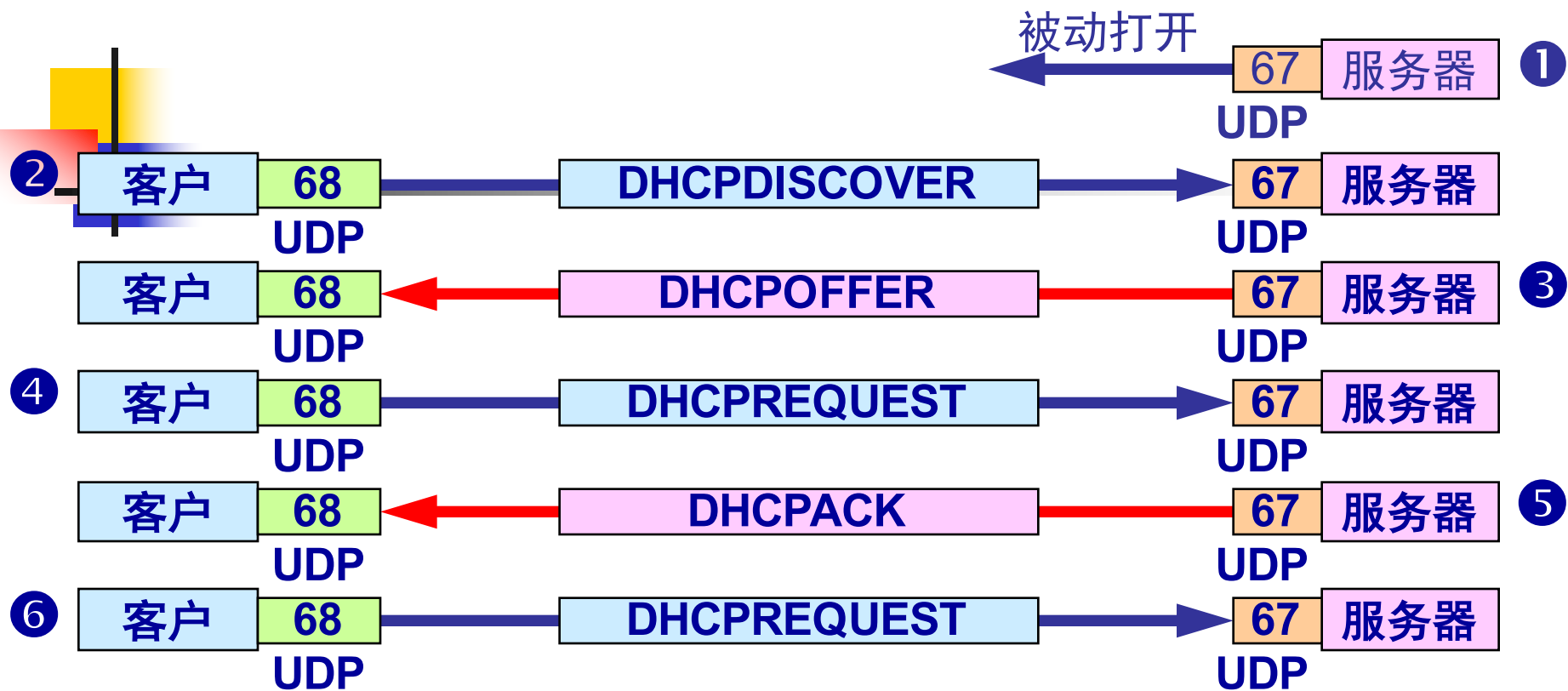
①：DHCP 服务器被动打开 UDP 端口 67，等待客户端发来的报文。



武汉大学

WUHAN UNIVERSITY

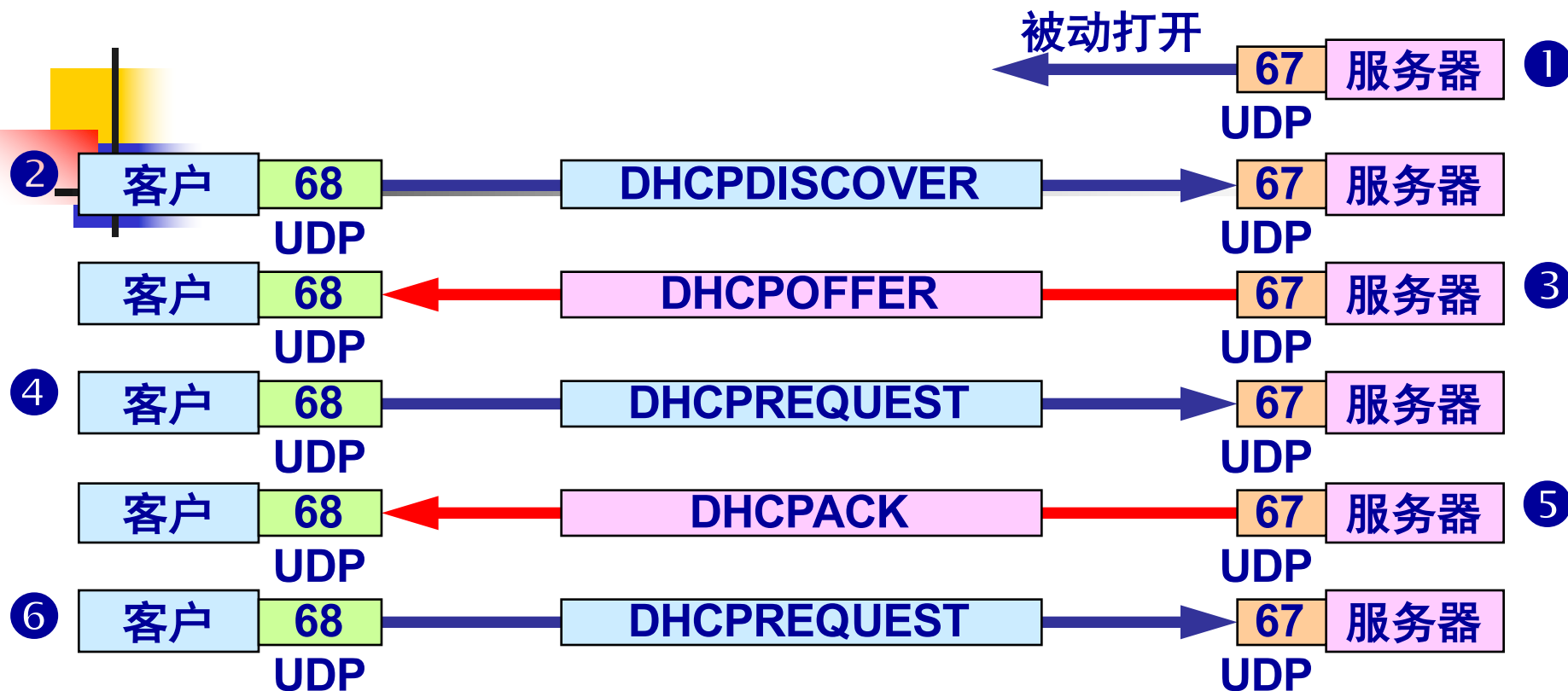
DHCP 协议的工作过程



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。

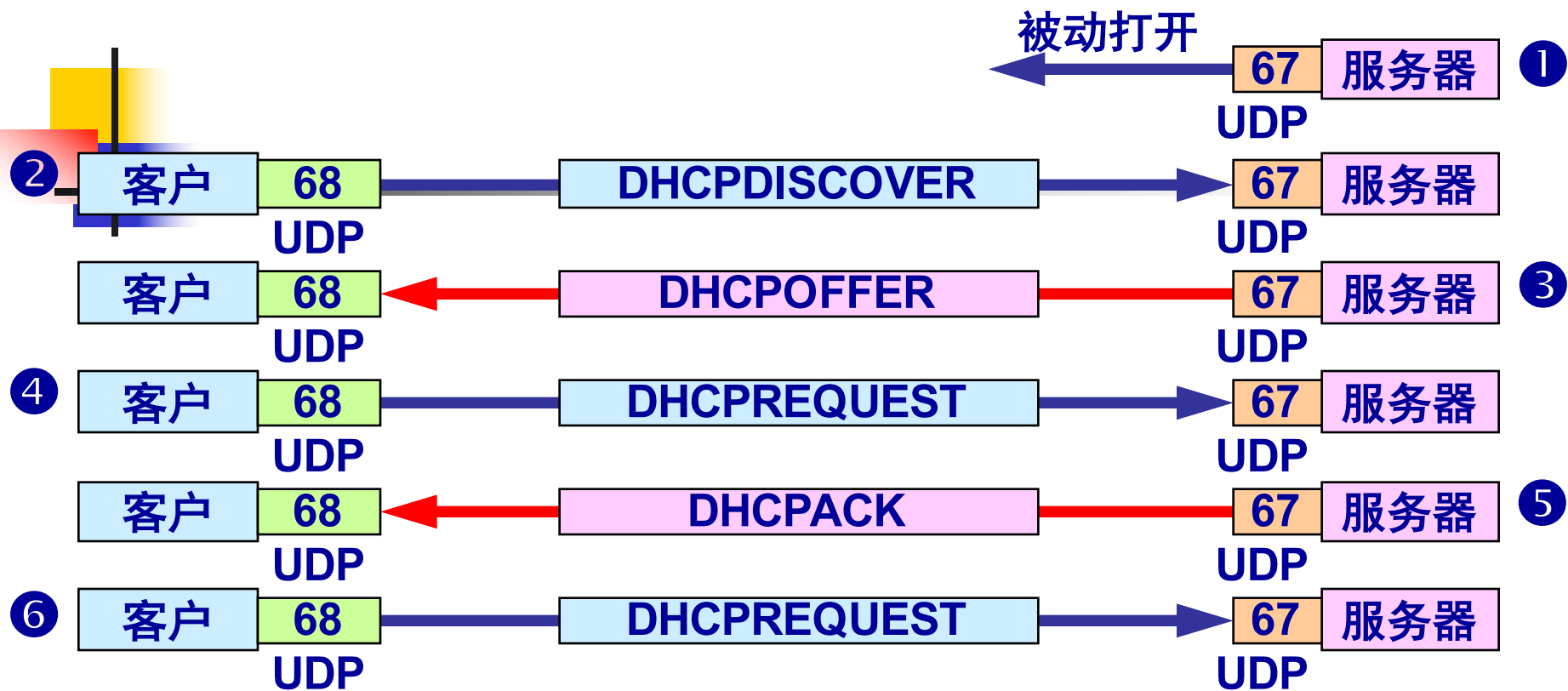


武汉

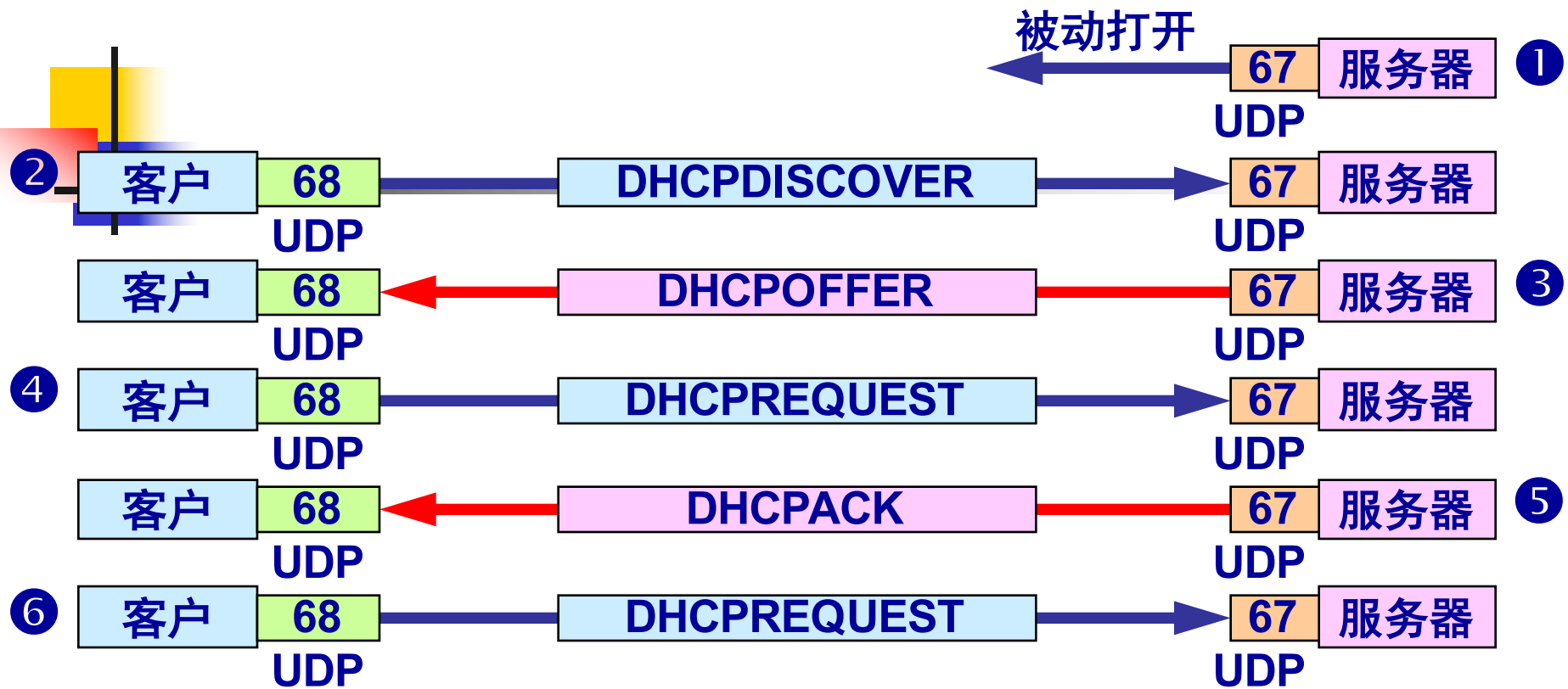
大学

WUHAN UNIVERSITY

DHCP 协议的工作过程

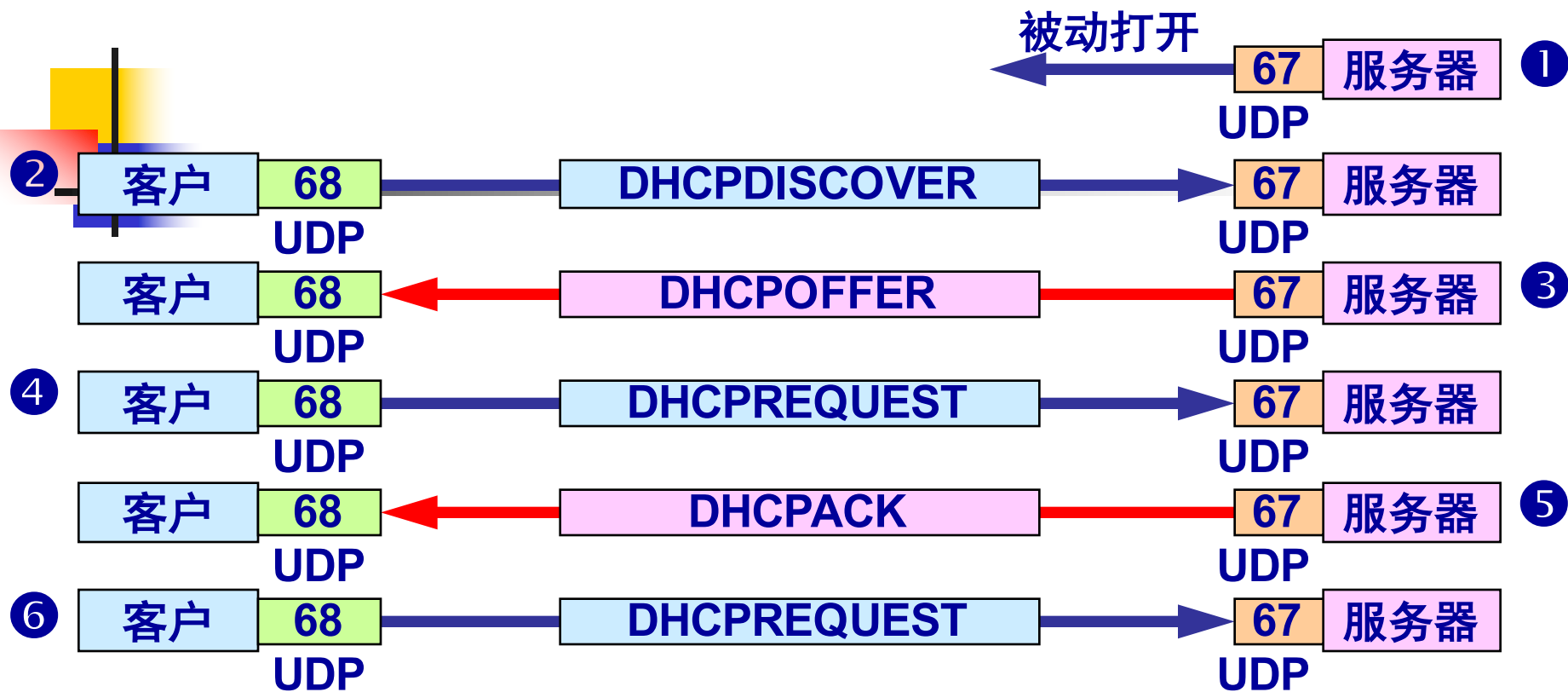


④： DHCP 客户从几个 DHCP 服务器中选择其中的一个，并向所选择的 DHCP 服务器发送 DHCP 请求报文。

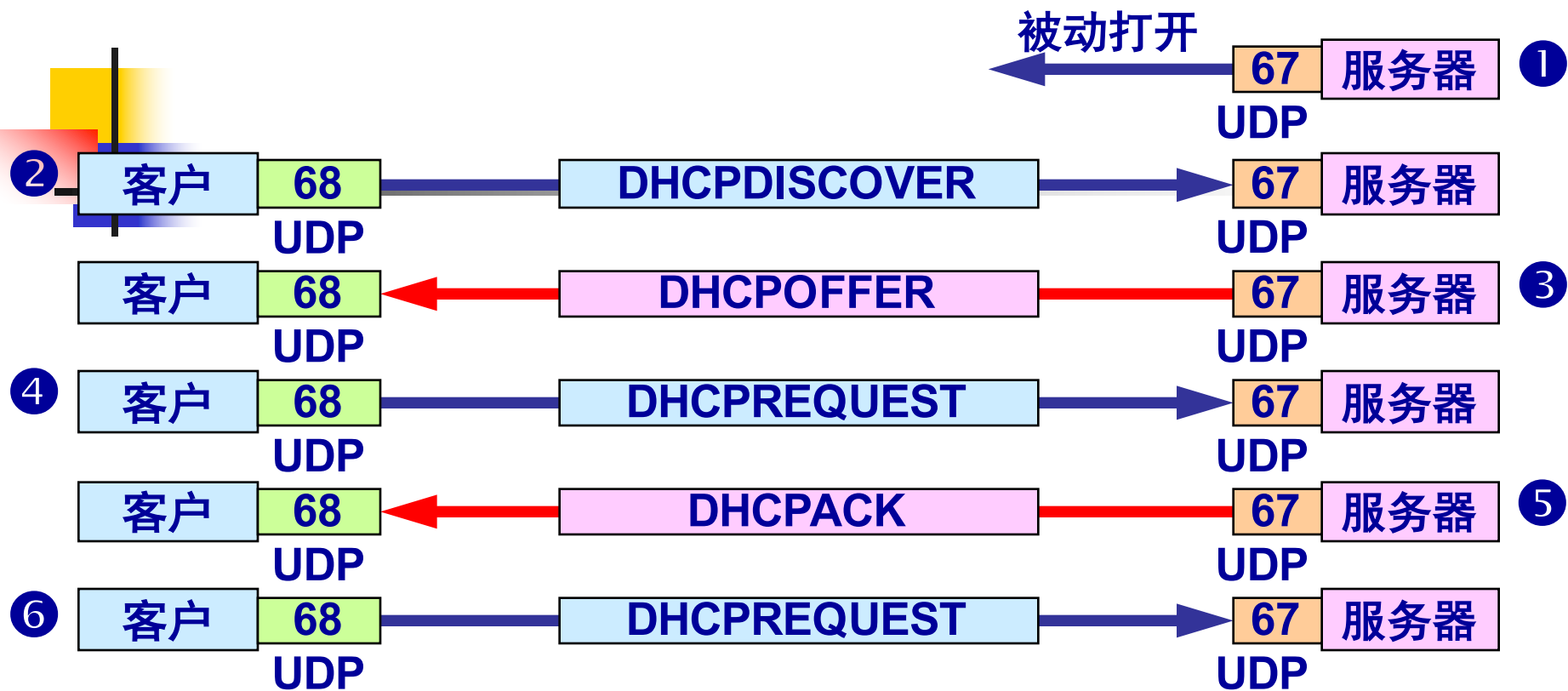


⑤： 被选择的DHCP服务器发送确认报文DHCPACK，进入已绑定状态，并可开始使用得到的临时IP地址了。

武汉大学 WUHAN UNIVERSITY DHCP 协议的工作过程

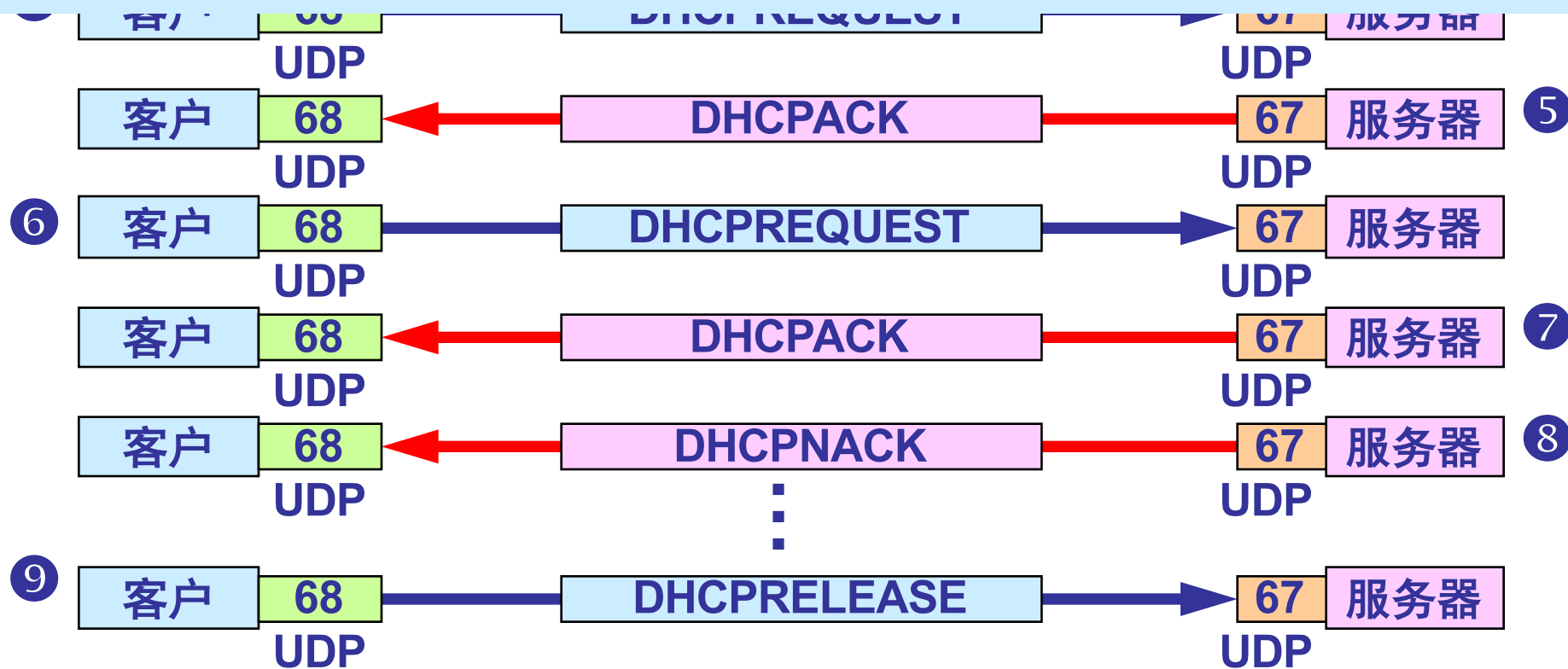


DHCP 客户现在要根据服务器提供的租用期 T 设置两个计时器 T_1 和 T_2 ，它们的超时时间分别是 $0.5T$ 和 $0.875T$ 。当超时时间到就要请求更新租用期。

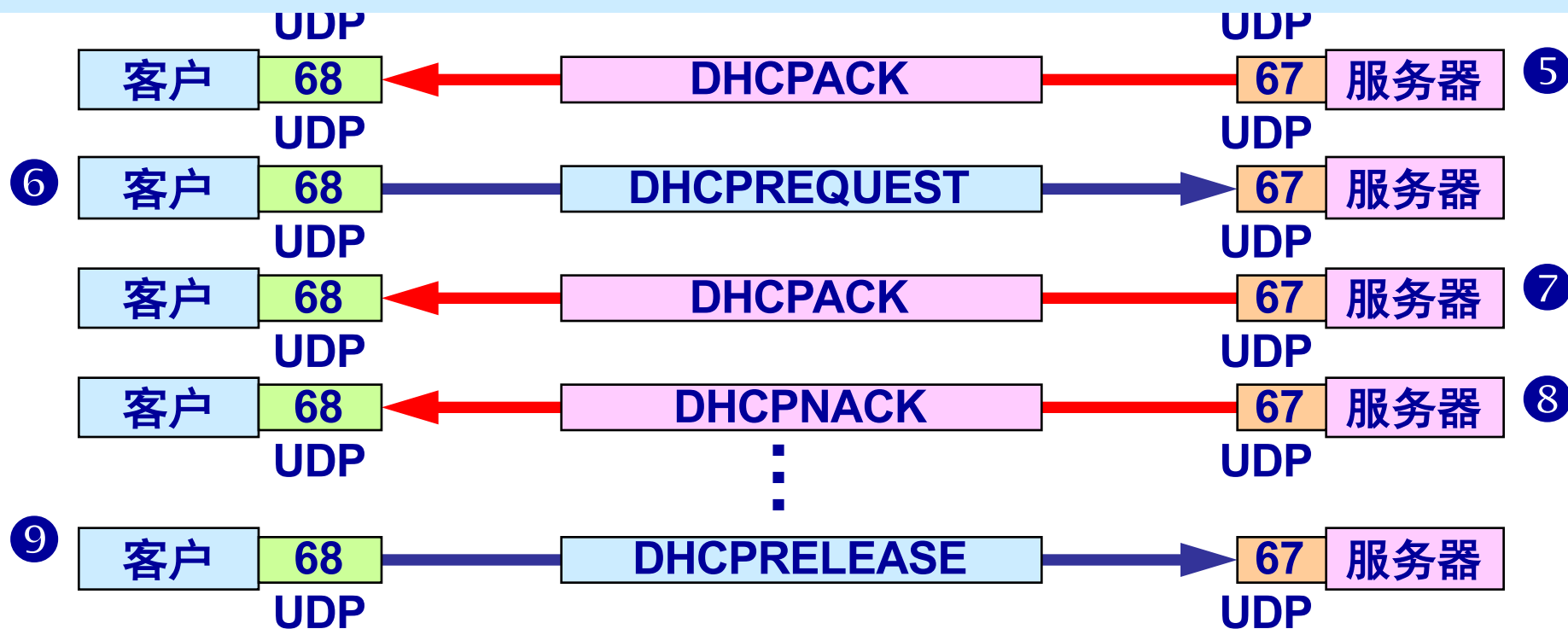


⑥： 租用期过了一半（T1 时间到），DHCP 发送请求报文 **DHCPREQUEST** 要求更新租用期。

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。

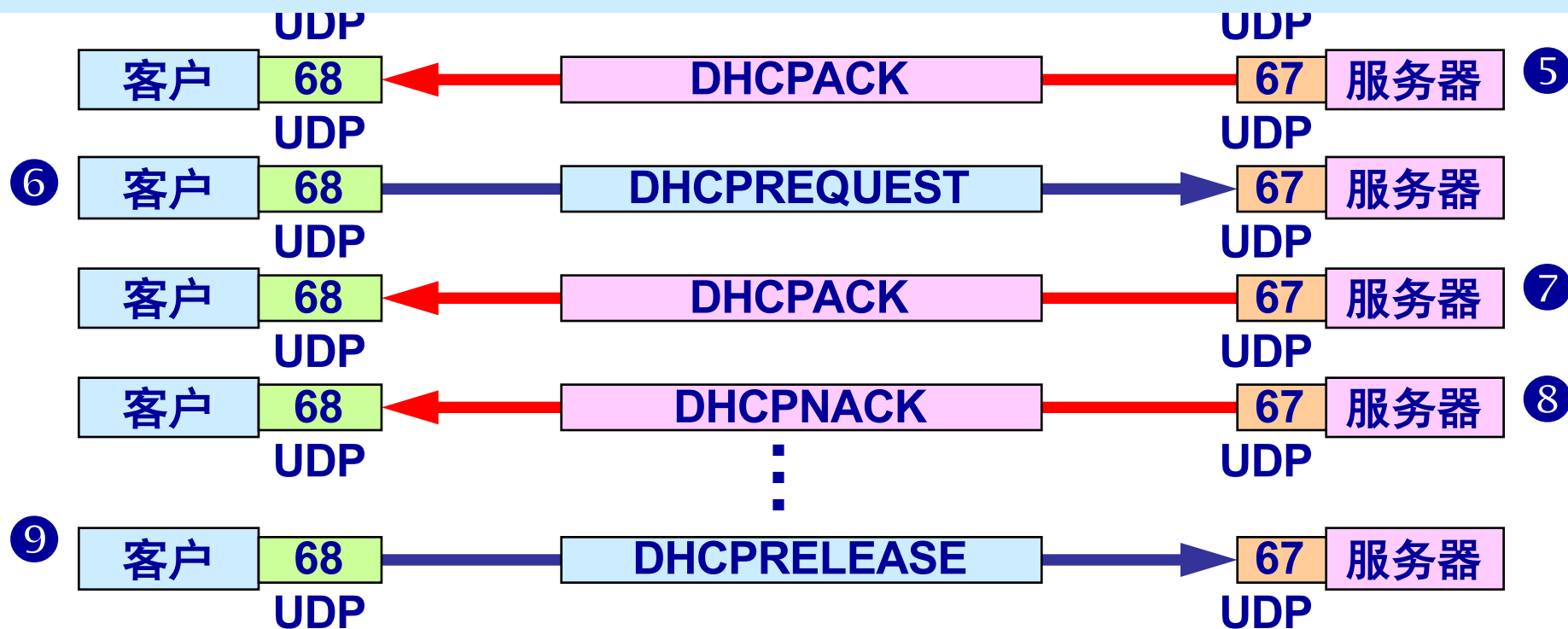


⑧：DHCP 服务器若不同意，则发回否认报文DHCPNACK。
 这时 DHCP 客户必须立即停止使用原来的 IP 地址，而必须重新申请 IP 地址（回到步骤 ②）。



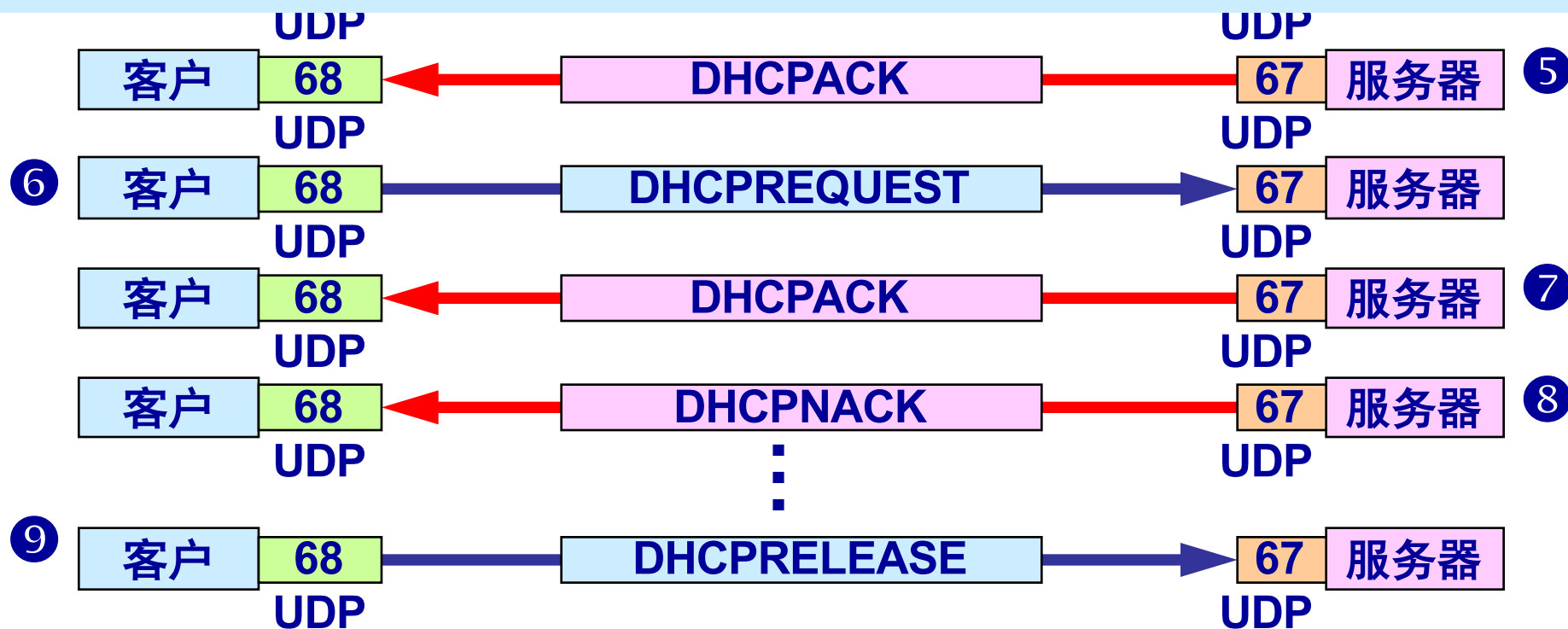


若DHCP服务器不响应步骤 ⑥ 的请求报文DHCPREQUEST，则在租用期过了 87.5% 时，DHCP 客户必须重新发送请求报文DHCPREQUEST（重复步骤 ⑥），然后又继续后面的步骤。





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。





6.9 P2P 应用

- 6.9.1 具有集中目录服务器的 P2P 工作方式
- 6.9.2 具有全分布式结构的 P2P 文件共享程序
- 6.9.3 P2P 文件分发的分析
- 6.9.4 在 P2P 对等方中搜索对象



P2P 工作方式概述

- 特点
 - 没有服务器
 - 终端节点直接连接
 - 连接可中断（终端离开），终端可改变IP地址
- 应用
 - 文件共享
 - 在线视频
 - VoIP（Skype）



6.9.1 集中目录服务器 P2P: Napster

- 最早出现的 P2P 技术，可提供免费下载 MP3 音乐。
- Napster 能够搜索音乐文件，能够提供检索功能。所有的音乐文件地址集中存放在一个 Napster 目录服务器中。使用者可很方便地下载需要的 MP3 文件。
- 用户要及时向 Napster 的目录服务器报告自己存有的音乐文件。当用户想下载某个 MP3 文件时，就向目录服务器发出询问。目录服务器检索出结果后向用户返回存放此文件的 PC 机的 IP 地址。Napster 的文件传输是分散的，但文件的定位则是**集中**的。
- **这种集中式目录服务器的缺点就是可靠性差。**Napster 被判决属于“间接侵害版权”，因此在 2000 年 7 月底 Napster 网站就被迫关闭了。



6.9.2 全分布式结构的 P2P: Gnutella

- Gnutella 是第二代 P2P 文件共享程序，采用**全分布方法**定位内容的P2P 文件共享应用程序。
- Gnutella 与 Napster 最大的区别就是不使用集中式的目录服务器，而是使用洪泛法在大量 Gnutella 用户之间进行查询。
- 为了不使查询的通信量过大，Gnutella 设计了一种有限范围的洪泛查询。这样可以减少倾注到互联网的查询流量，但由于查询的范围受限，因而这也影响到查询定位的准确性。



6.9.2 全分布式结构的 P2P：电骡 eMule

- 电骡 eMule 使用分散定位和分散传输技术，把每一个文件划分为许多小文件块，并使用多源文件传输协议 MFTP 进行传送。因此用户可以同时从很多地方下载一个文件中的不同文件块。由于每一个文件块都很小，并且是并行下载，所以下载可以比较快地完成。
- eMule 用户在下载文件的同时，也在上传文件，因此，互联网上成千上万的 eMule 用户在同时下载和上传一个个小的文件块。



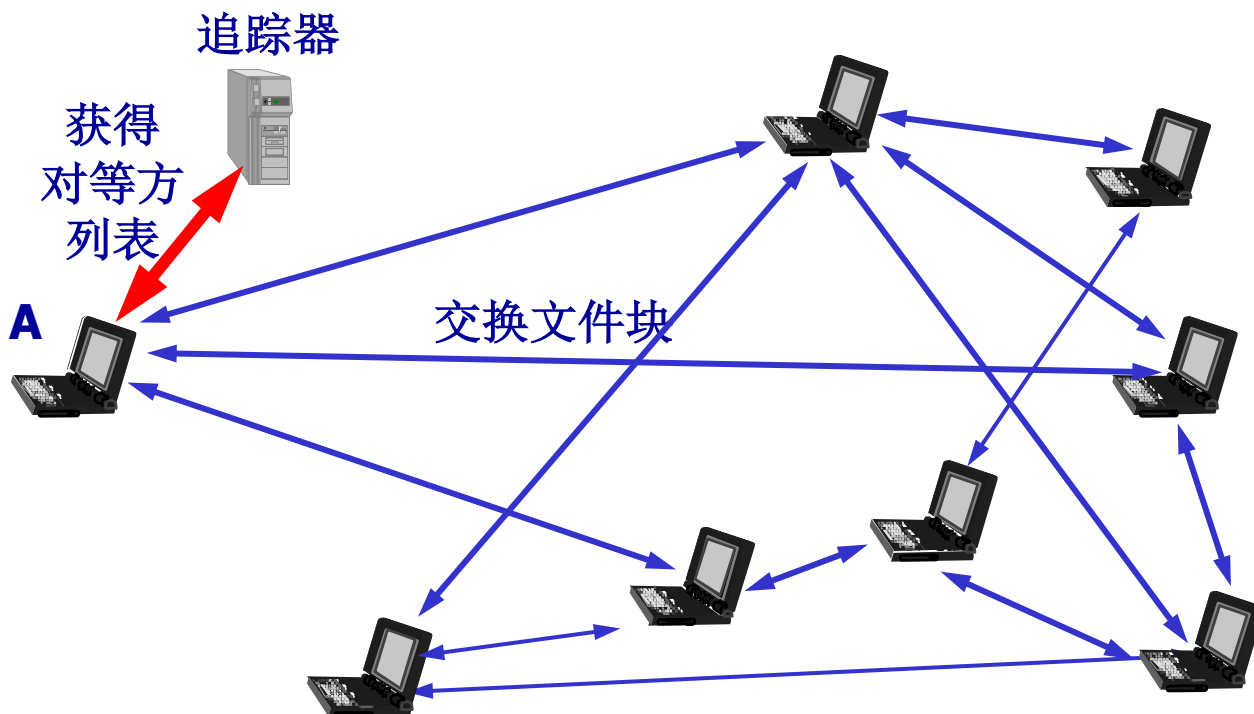
eMule 的其他特点

- eMule 使用了一些服务器。这些服务器并不是保存音频/视频文件，而是保存用户的有关信息，因而可以告诉用户从哪些地方可以下载到所需的文件。
- eMule 使用了专门定义的文件夹，让用户存放可以和其他用户共享的文件。
- eMule 的下载文件规则是鼓励用户向其他用户上传文件。用户上传文件越多，其下载文件的优先级就越高（因而下载就越快）。



使用 P2P 的比特洪流 BT

- BitTorrent 所有对等方集合称为一个**洪流 (torrent)**，下载文件的数据单元为长度固定的**文件块 (chunk)**。基础设施结点，叫做**追踪器 (tracker)**。





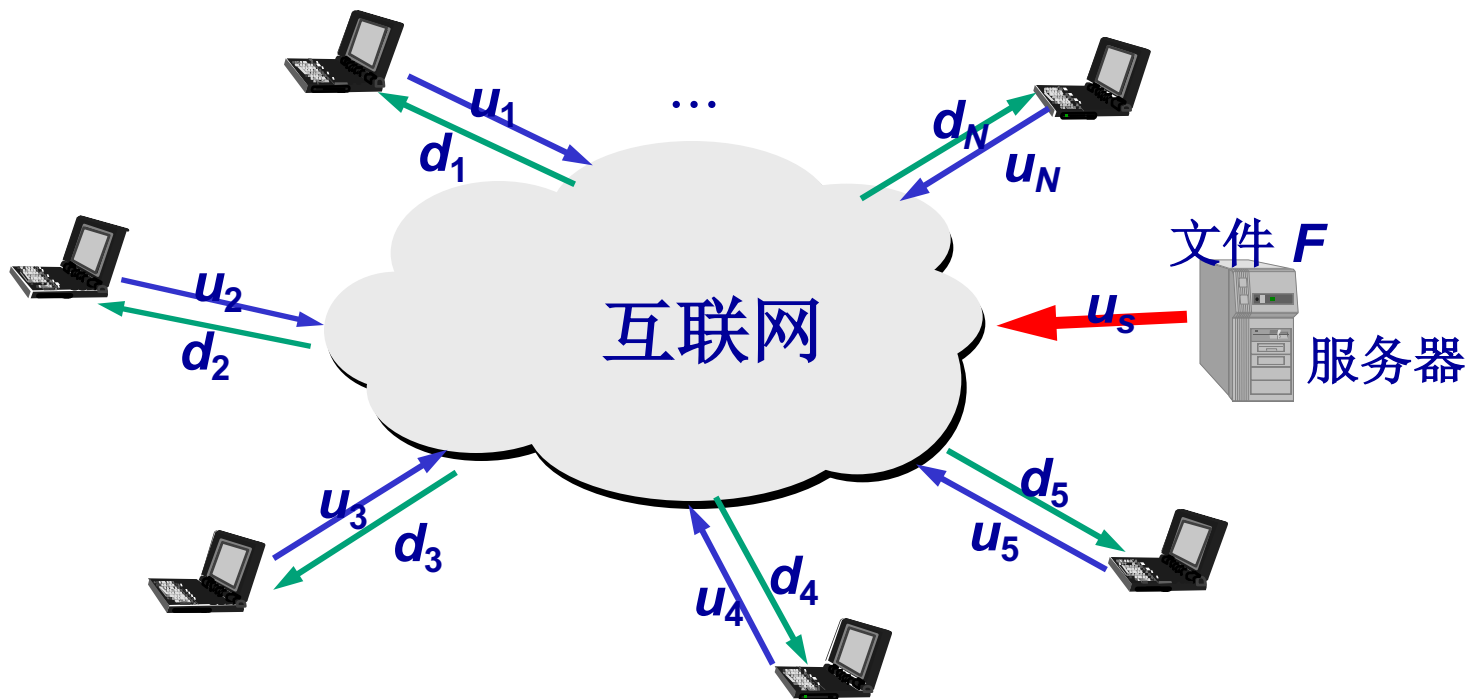
BT 协议

- 当一个新的对等方 **A** 加入洪流时，追踪器就随机地从参与的对等方集合中选择若干个告诉 **A**。于是 **A** 就和这些对等方建立了 **TCP** 连接。
- 应该先下载哪些文件块？
 - 最稀有的优先 (rarest first)。
- 应该给哪些对等方发送文件
 - 给哪些给**A**最多数据的对等分发送
 - 也随机选择一个其他对等方进行发送



6.9.3 P2P 文件分发的分析

- 有 N 台主机从服务器下载一个大文件，其长度为 F bit。假定主机与互联网连接的链路的上传速率和下载速率分别为 u_i 和 d_i ，单位都是 bit/s。





C/S 方式下分发的最短时间

- 从服务器端考虑，所有主机分发完毕的最短时间 T_{cs} 不可能小于 NF/u_s ;
- 下载速率最慢的主机的下载速率为 d_{\min} ，则 T_{cs} 不可能小于 F/d_{\min} 。
- 由此可得出所有主机都下载完文件 F 的最少时间是：

$$T_{cs} = \max (NF/u_s, F/d_{\min})$$



P2P 方式下分发的最短时间

- 服务器：分发文件的最少时间不可能小于 F/u_s ;
- 单个客户端：下载文件分发的最少时间不可能小于 F/d_{\min} ;
- 所有客户端：总共需要下载 NF ，也就是总共需要上传 NF ，有全部的客户端和服务端提供，即 u_T 是上传速率之和。
- 所有主机都下载完文件 F 的最少时间的下限是：

$$T_{p2p} \geq \max (F/u_s , F/d_{\min} , NF/u_T)$$



6.9.4 在 P2P 对等方中搜索对象

- 分布式散列表 DHT (Distributed Hash Table)。
 - 分布式的p2p数据库
- 数据库存放 (key, value) 对
 - 如key:身份证号, value: 人名
 - Key: 电影; value: IP地址
- 将这些数据分布存储在对等点上
- 当一个用户 (对等点) 需要某数据时 (即通过key查询value)
 - DHT返回value

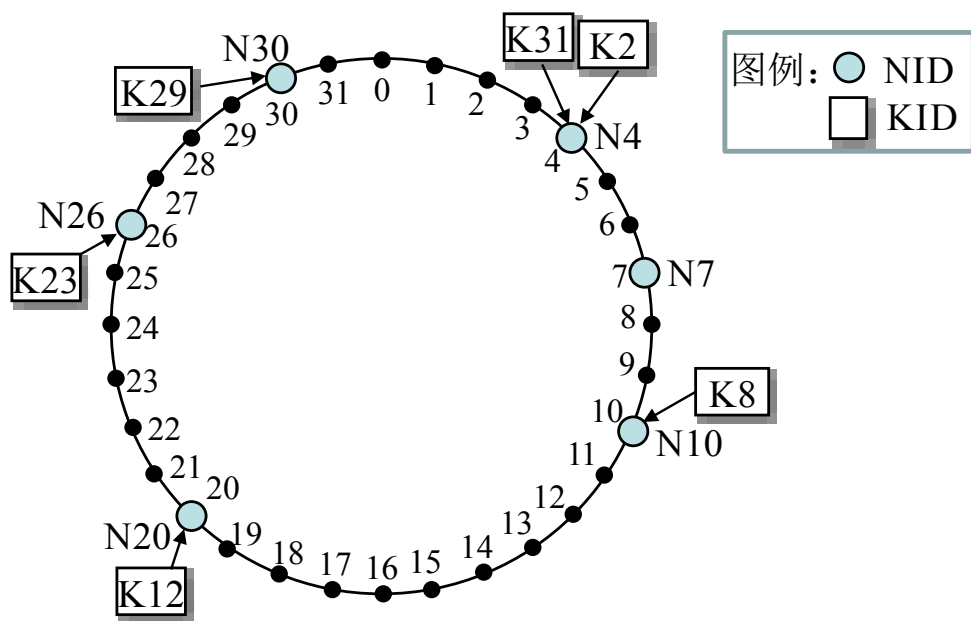


6.9.4 在 P2P 对等方中搜索对象

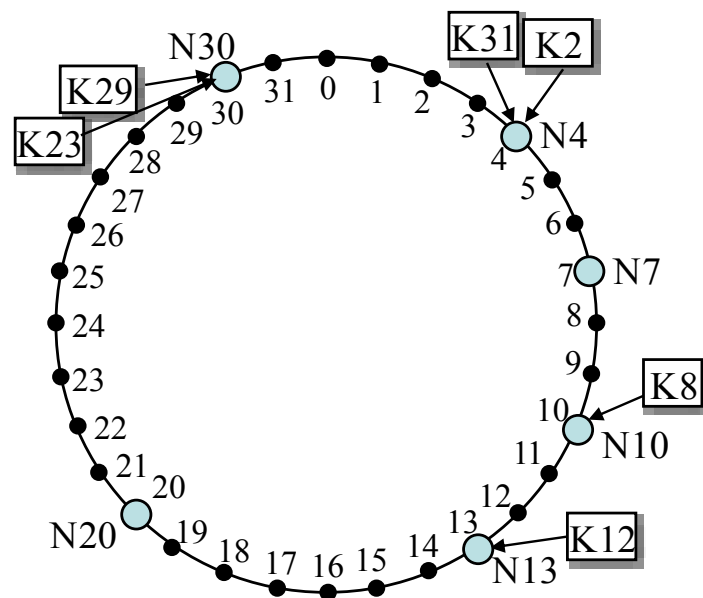
- 关键问题。
 - 如何将 these 数据进行分布式存储，即哪些对等点应该存储哪些数据
- 解决方案
 - 将每个key转换为不同的整数（hash）
 - 每个对等点赋一个编号（整数）
 - 将key对应的数据存放在离其最近的对等点

基于 DHT 的 Chord 环

- 每个资源由 Chord 环上与其标识符值最接近的下一个结点提供服务。



(a) KID 和 NID 映射在环上



(b) N13加入， N26退出



通过指针表加速 Chord 表查找

- 为了加速查找，在 Chord 环上可以增加一些**指针表** (finger table)，它又称为**路由表**或**查找器表**。
- 第 1 列第 i 行计算 $(N4 + 2^{i-1})$ ，得出后继结点。

