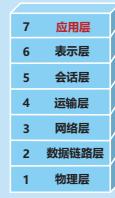


# 第6章

# 应用层

## 计算机网络体系结构

OSI 的七层协议体系结构



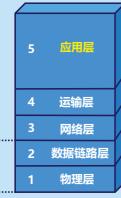
(a)

TCP/IP 的四层协议体系结构



(b)

五层协议的体系结构



(c)

- 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

## 应用层协议

- 精确定义不同主机中的多个应用进程之间的通信规则。

- 包括:

- ◆ 应用进程交换的报文类型, 如请求报文和响应报文。
- ◆ 各种报文类型的语法, 如报文中的各个字段及其详细描述。
- ◆ 字段的语义, 即包含在字段中的信息的含义。
- ◆ 进程何时、如何发送报文, 以及对报文进行响应的规则。

## 应用层协议

### 不同于网络应用

万维网	HTTP
电子邮件	SMTP, POP3
文件传输	FTP

### 许多都基于客户服务器方式

客户 (client)	服务器 (server)
服务请求方	服务提供方
应用进程	应用进程

## 6.1 域名系统 DNS

### 6.1.1 域名系统概述

### 域名系统概述

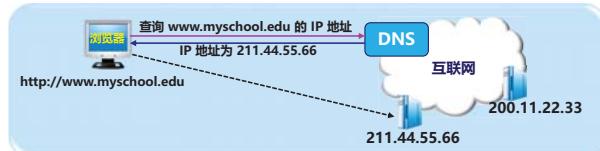
### 互联网的域名结构

### 域名服务器

### 6.1.1 域名系统概述

#### ● 域名系统 DNS (Domain Name System) :

- ◆ 互联网使用的命名系统。
- ◆ 用来把人们使用的机器名字（域名）转换为 IP 地址。
- ◆ 为互联网的各种网络应用提供了核心服务。



### 6.1.1 域名系统概述

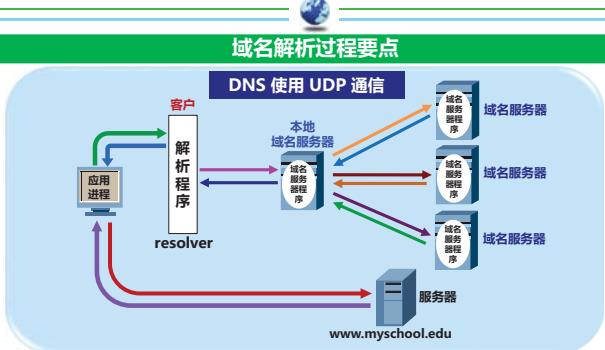
#### ● 域名采用层次树状结构的命名方法: www.myschool.edu。

- DNS 是一个联机分布式数据库系统, 采用客户服务器方式。

- 域名到 IP 地址的解析是由若干个域名服务器程序共同完成。

- 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

- **命名方法：**层次树状结构方法。
  - 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**的层次结构的名字，即**域名** (domain name)。
  - **域 (domain)：**
    - ◆ 名字空间中一个**可被管理**的划分。
    - ◆ 可以划分为**子域**，而子域还可继续划分为子域的子域，这样就形成了顶级域、二级域、三级域，等等。

### 6.1.2 互联网的域名结构

- **域名结构：**层次结构。由标号(label)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。

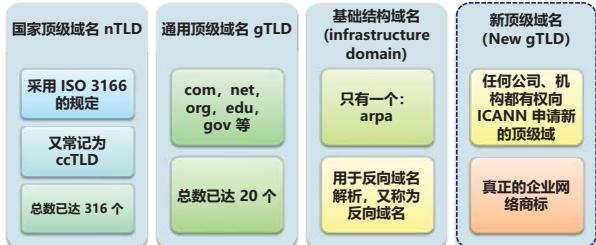
The diagram illustrates the structure of a domain name: **mail.cctv.com**. The text is arranged as follows:

- 每一个标号不超过 63 个字符** (Each label has no more than 63 characters) is positioned above the first part.
- 不区分大小写字母** (Case-insensitive) is positioned above the second part.
- 完整域名总共不超过 255 个字符** (The total length of the full domain name does not exceed 255 characters) is positioned above the third part.
- 注意：域名只是个逻辑概念，并不代表计算机所在的物理地点。** (Note: A domain name is just a logical concept and does not represent the physical location of the computer.) is enclosed in a blue box on the right.

Below the main text, it is broken down into three levels:  
**三级域名 . 二级域名 . 顶级域名**  
(级别最低) (级别中间) (级别最高)

**注意：域名称号中除连字符（-）外不能使用其他的标点符号！**

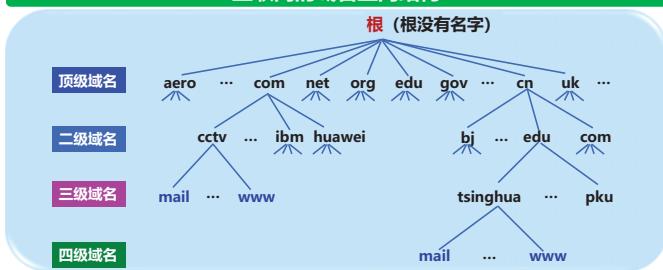
全球顶级域名 TLD (Top Level Domain)



在国家顶级域名下注册的二级域名均由该国家自行确定。

我国把二级域名划分为“类别域名”和“行政区域名”两大类。

互联网的域名空间结构



域名树的树叶就是计算机的名字，它不能再继续往下划分子域了。

## 互联网的域名空间结构

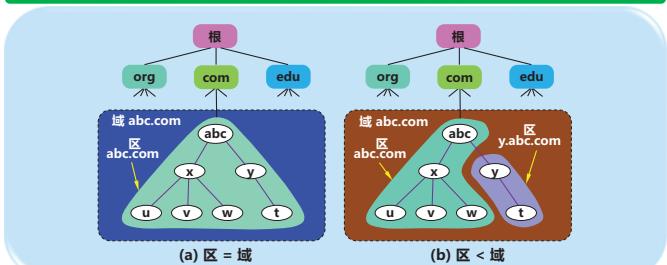
**注意：**互联网的名字空间是按照机构的组织来划分的，与物理的网络无关，与 IP 地址中的“子网”也没有关系。

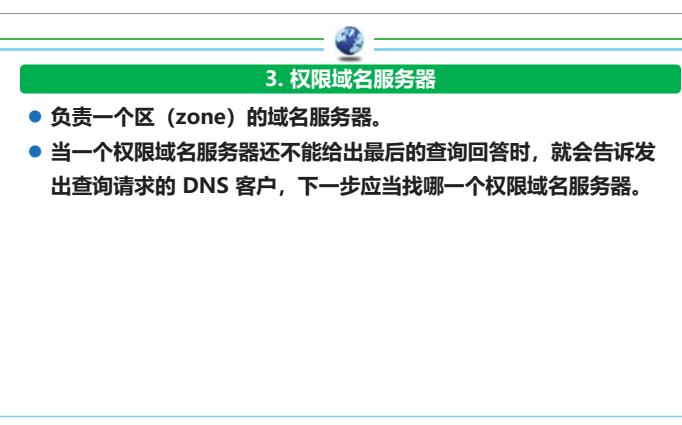
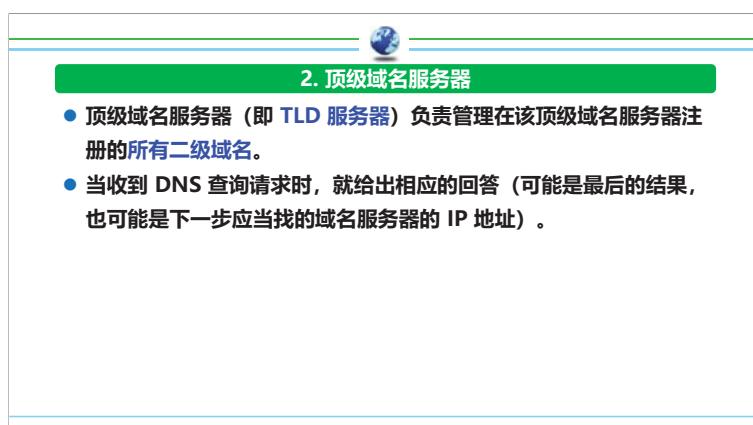
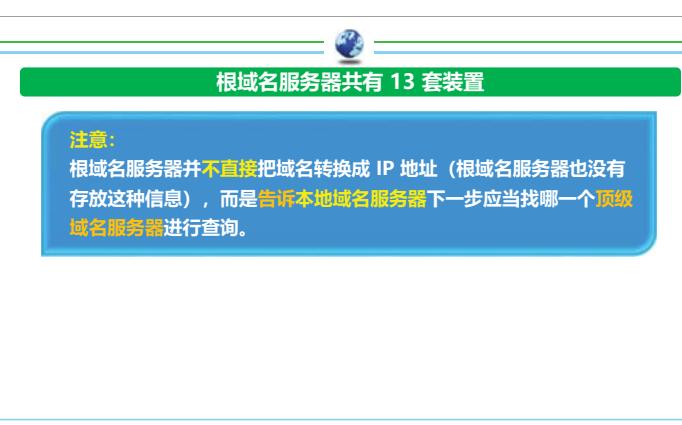
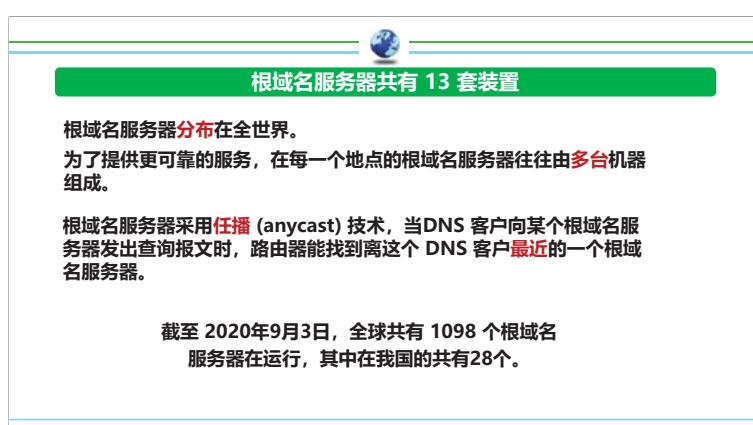
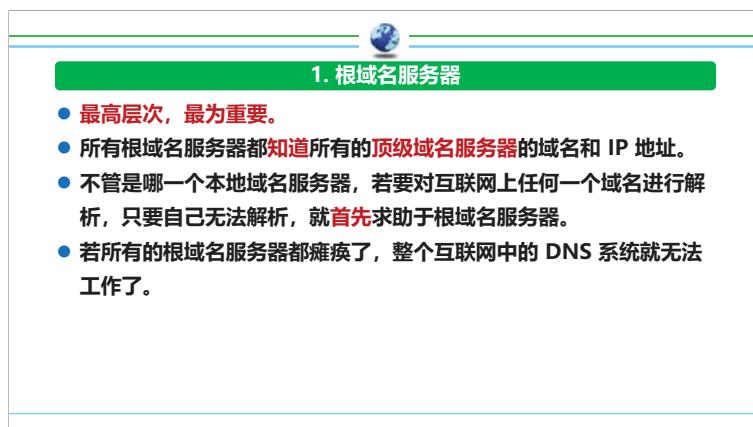
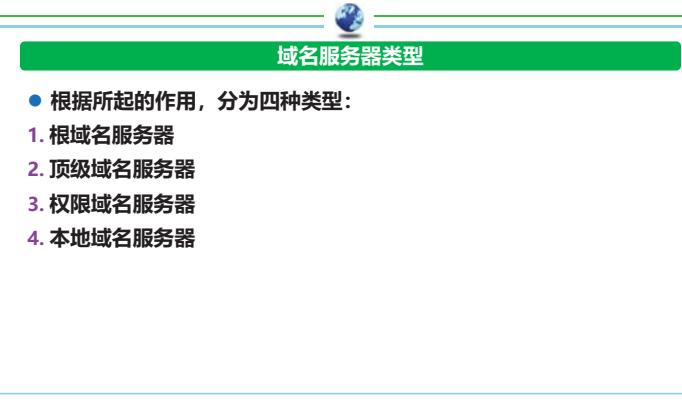
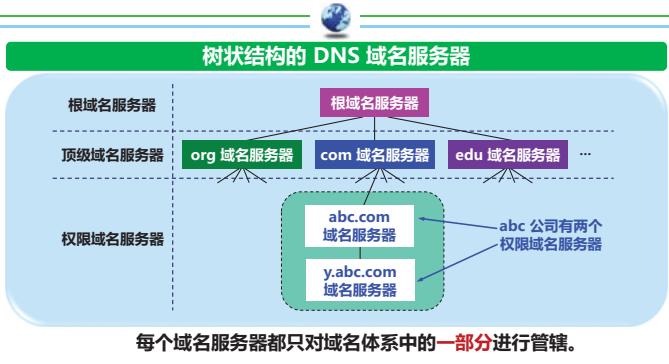
### 6.1.3 域名服务器

- 实现域名系统使用分布在各地的**域名服务器**（DNS 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





#### 4. 本地域名服务器

- 非常重要。
- 当一个主机发出 DNS 查询请求时，该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，都可以拥有一个本地域名服务器。
- 当所要查询的主机也属于同一个本地 ISP 时，该本地域名服务器立即就能将所查询的主机名转换为它的 IP 地址，而不需要再去询问其他的域名服务器。
- 本地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，保证了数据的一致性。

#### 域名的解析过程

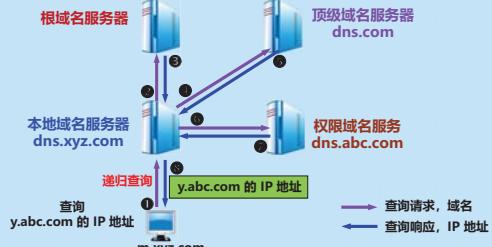
##### 递归查询

- 通常，主机向本地域名服务器查询时使用。
- 若不知道，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。

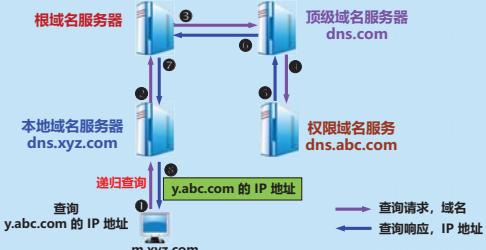
##### 迭代查询

- 本地域名服务器向根域名服务器查询时使用。
- 要么给出所要查询的 IP 地址，要么告诉下一个要查询的域名服务器的 IP 地址。
- 本地域名服务器继续后续查询。

#### 迭代查询（本地域名服务器向根域名服务器）



#### 递归查询（主机向本地域名服务器）



#### 高速缓存

- 也称为高速缓存域名服务器。
- 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。
- 作用：大大减轻根域名服务器的负担，使 DNS 查询请求和回答报文的数量大为减少。
- 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。
- 当权限域名服务器回答一个查询请求时，在响应中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。
- 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。
- 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。
- 是文件共享协议的一个大类。

## 文件共享协议

- **文件传送协议:** FTP, TFTP 等。
  - ◆ 复制整个文件。对文件副本进行访问。
    - 若要存取一个文件，就必须先获得一个本地文件副本。
    - 若要修改文件，只能对文件副本进行修改，然后再将修改后的文件副本传回到原节点。
- **联机访问 (on-line access) 协议:** NFS 等。
  - ◆ 允许同时对一个文件进行存取。
  - ◆ 远地共享文件访问，如同对本地文件的访问一样。
  - ◆ 透明存取，不需要对该应用程序作明显的改动。
  - ◆ **由操作系统负责。**

## 6.2.2 FTP 的基本工作原理

网络环境下复制文件的**复杂性**:

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

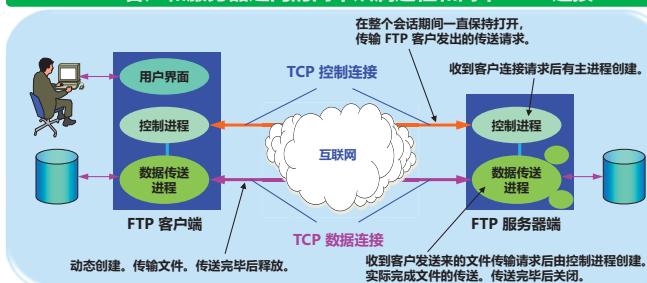
## FTP 特点

- 只提供文件传送的一些**基本服务**，它使用 **TCP 可靠的运输服务**。
- **主要功能:** 减少或消除在不同操作系统下处理文件的不兼容性。
- **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；
    - 若干个**从属进程**，负责处理单个请求。

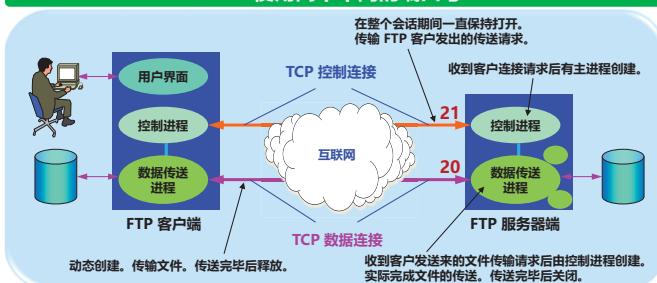
## FTP 主进程的工作步骤

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。

## FTP 客户和服务器之间的两个从属进程和两个 TCP 连接



## FTP 使用两个不同的端口号



## NFS 采用的思路

- **FTP 并非对所有的数据传输都是最佳的：仅能访问副本。**
- **NFS 允许应用进程打开一个远地文件，并能在该文件的某一个特定的位置上开始读写数据。**
- **NFS 可使用户只复制一个大文件中的一个很小的片段，而不需要复制整个大文件。**
- **NFS 在网络上传送的只是少量的修改数据。**

## 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。
- 使用**客户服务器方式**和使用 **UDP 数据报**，因此 TFTP 需要有自己的差错改正措施。
- 只支持文件传输，**不支持交互**。
- 没有庞大的命令集，没有列目录的功能，也不能对用户进行身份鉴别。
- **优点：**(1) 可用于 UDP 环境；(2) 代码所占的内存较小。

### TFTP 的主要特点

- 每次传送的数据报文中有 512 字节的数据，但最后一次可不足 512 字节。
- 数据报文按序编号，从 1 开始。
- 支持 ASCII 码或二进制传送。
- 可对文件进行读或写。
- 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未收到下一个文件块，需重发确认 PDU，保证文件的传送不致因某一个数据报的丢失而告失败。

### TFTP 的工作过程

- 开始工作时，TFTP 客户进程发送一个读请求或写请求报文给 TFTP 服务器进程，其 UDP 端口号为 69。
- TFTP 服务器进程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报文。
- 若文件长度不是 512 字节的整数倍，则最后传送数据报文的数据字段一定不满 512 字节，作为文件结束的标志。

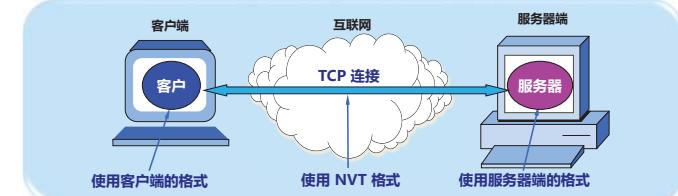
### 6.3 远程终端协议 TELNET

- 是一个简单的远程终端协议，是互联网的正式标准。
- 允许用户在其所在地通过 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）。
- 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。
- 服务是透明的。
- 又称为终端仿真协议。

### TELNET 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。
- 服务器中的主进程等待新的请求，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



TELNET 的选项协商 (Option Negotiation) 使客户和服务器可商定使用更多的终端功能，协商的双方是平等的。

### NVT (Network Virtual Terminal) 格式

两个字符集：数据，控制

标准 ASCII 字符

数据字符：

0				
---	--	--	--	--

控制字符：

1				
---	--	--	--	--

- 客户端把用户的击键和命令转换成 NVT 格式，并送交服务器。
- 服务器端把收到的数据和命令，从 NVT 格式转换成远地系统所需的形式。
- 向客户返回数据时，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形式。

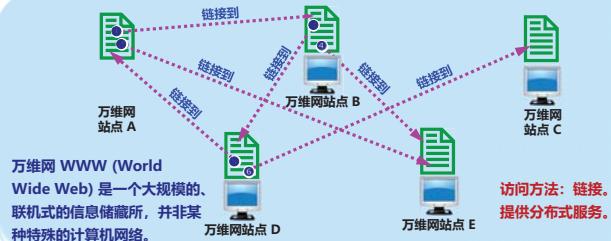
### 6.4 万维网 WWW

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	社交网站

### 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**”。

### 6.4.1 万维网概述



### 万维网是分布式超媒体 (hypermedia) 系统

- 是超文本 (hypertext) 系统的**扩充**。
- **超文本**: 由多个信息源**链接成**。是万维网的基础。
- 超媒体与超文本的区别: 文档内容不同。
  - ◆ 超文本文档仅包含文本信息。
  - ◆ 超媒体文档还包含其他信息, 如图形、图像、声音、动画, 甚至活动视频图像等。
- **分布式系统**
  - ◆ 信息分布在整个互联网上。每台主机上的文档都独立进行管理。

### 万维网的工作方式

- 以**客户服务器**方式工作。
- 客户程序: 浏览器。
- 服务器程序: 在万维网文档所驻留的主机上运行。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 (1/2)

- (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)。
  - 使每一个文档在整个互联网的范围内具有**唯一的**标识符 URL。
- (2) 用什么**协议**来实现万维网上的各种链接?
  - 使用**超文本传送协议 HTTP** (HyperText Transfer Protocol)。
  - HTTP 是一个应用层协议, 使用 **TCP** 连接进行可靠的传送。

### 万维网必须解决的问题 (2/2)

- (3) 怎样使不同作者**创作**的不同风格的万维网文档都能在互联网上的各种主机上**显示出来**, 同时让用户清楚地知道在什么地方存在着**链接**?
  - 使用**超文本标记语言 HTML** (HyperText Markup Language)。
- (4) 怎样使用户能够很方便地**找到**所需的信息?
  - 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

- 是对互联网上资源的位置和访问方法的一种**简洁表示**。
- 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**。
- 实际上就是在互联网上的**资源的地址**。
- 显然, 互联网上的所有资源, 都有一个**唯一确定的URL**。
- **资源**: 指在互联网上可以被访问的任何对象, 包括文件目录、文件、文档、图像、声音等, 以及与互联网相连的**任何形式**的数据。

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。

### 1. URL 的格式

- 由以冒号 (:) 隔开的两大部分组成, 对字符大写或小写没有要求。
- 一般形式:

<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

## 1. URL 的格式

- 由以冒号 (:) 隔开的两大部分组成，对字符大写或小写没有要求。
- 一般形式：

<协议>://<主机>:<端口>/<路径>

规定的格式

## 1. URL 的格式

- 由以冒号 (:) 隔开的两大部分组成，对字符大写或小写没有要求。
- 一般形式：

<协议>://<主机>:<端口>/<路径>

存放资源的主机在互联网中的域名，也可以是用点分十进制的 IP 地址。

## 1. URL 的格式

- 由以冒号 (:) 隔开的两大部分组成，对字符大写或小写没有要求。
- 一般形式：

<协议>://<主机>:<端口>/<路径>

端口号。省略时使用默认端口号。

## 1. URL 的格式

- 由以冒号 (:) 隔开的两大部分组成，对字符大写或小写没有要求。
- 一般形式：

<协议>://<主机>:<端口>/<路径>

资源所在目录位置。  
区分大小写。省略时使用所定义的默认路径。后面可能还有一些选项。

## 2. 使用 HTTP 的 URL

使用 HTTP 协议。

规定的格式。

主机域名或 IP 地址。

HTTP 的默认端口号是 80，可省略。

http://<主机>:<端口>/<路径>

省略时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。

http://www.tsinghua.edu.cn/publish/newthu/newthu\_cnt/faculties/index.html

主机域名

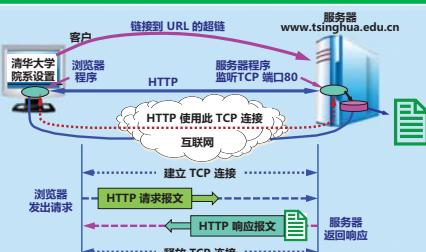
路径名

## 6.4.3 超文本传送协议 HTTP

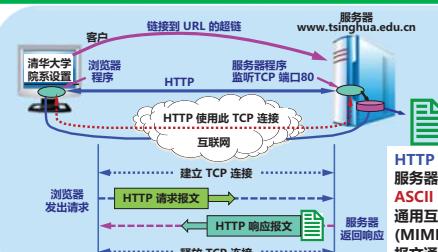
- HTTP 是面向事务的 (transaction-oriented) 应用层协议。
- 使用 TCP 连接进行可靠的传送。
- 定义了浏览器与万维网服务器通信的格式和规则。
- 是万维网上能够可靠地交换文件 (包括文本、声音、图像等各种多媒体文件) 的重要基础。

HTTP 不仅传送完成超文本跳转所必需的信息，而且也传送任何可从互联网上得到的信息，如文本、超文本、声音和图像等。

## 1. HTTP 的操作过程



## 1. HTTP 的操作过程



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。

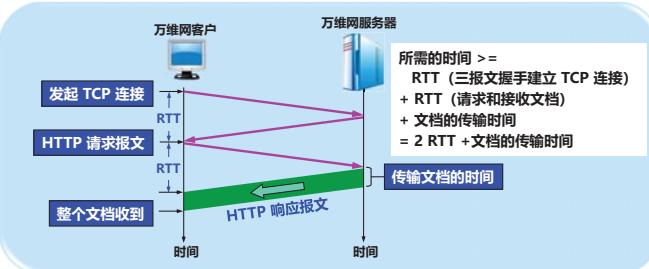
## 用户浏览页面的两种方法

- 在浏览器的地址窗口中键入所要找的页面的 URL。
- 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

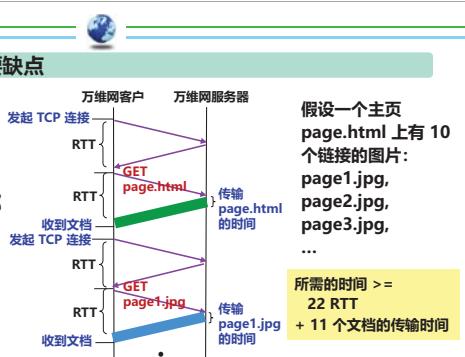
- HTTP 使用了面向连接的 TCP 作为运输层协议，保证了数据的可靠传输。
- HTTP 协议本身也是无连接的。
- HTTP 是无状态的 (stateless)，服务器并不记得曾经访问过的客户，简化了服务器的设计，使服务器更容易支持大量并发的 HTTP 请求。

## 请求一个万维网文档所需的时间



## 协议 HTTP/1.0 的主要缺点

- 每请求一个文档就要有两倍 RTT 的开销。
- 客户和服务器每一次建立新的 TCP 连接都要分配缓存和变量。
- 这种非持续连接使服务器的负担很重。

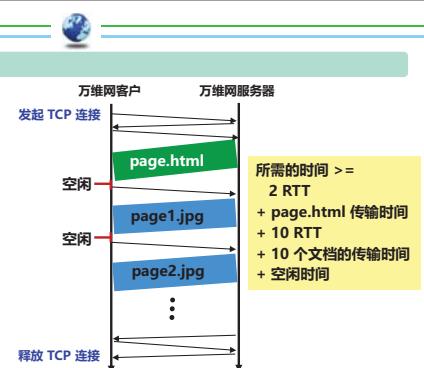


## 协议 HTTP/1.1 使用持续连接

- 持续连接 (persistent connection)**：服务器在发送响应后仍然在一段时间内保持这条连接（不释放），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。
- 只要文档都在同一个服务器上，就可以继续使用该 TCP 连接。
- 两种工作方式：**
  - 非流水线方式 (without pipelining)
  - 流水线方式 (with pipelining)。

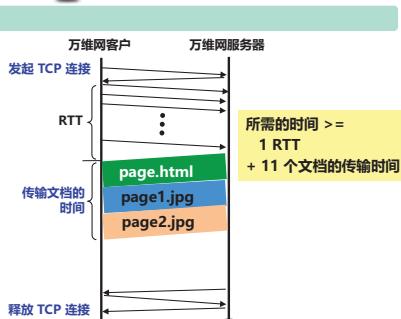
## 持续连接：非流水线方式

- 客户在收到前一个响应之后才能发出下一个请求。
- 缺点：**TCP 连接空闲状态。



## 持续连接：流水线方式

- 客户在收到响应报文之前就能够接着发送新的请求报文。
- 连续的多个请求报文到达服务器后，服务器就可连续发回响应报文。
- 下载效率提高。



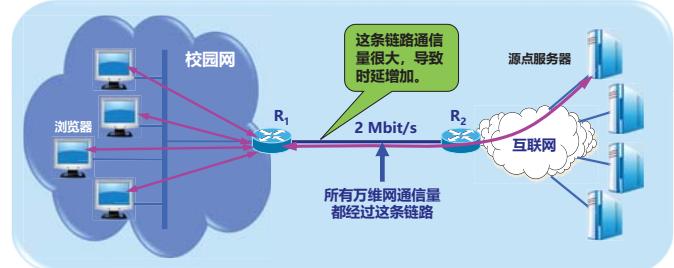
## 协议 HTTP/2

- 是协议 HTTP/1.1 的升级版本。
- 服务器可以并行发回响应（使用同一个 TCP 连接）。
- 允许客户复用 TCP 连接进行多个请求。
- 把所有的报文都划分为许多较小的二进制编码的帧，并采用了新的压缩算法，不发送重复的首部字段，大大减小了头部的开销，提高了传输效率。
- 向后兼容。

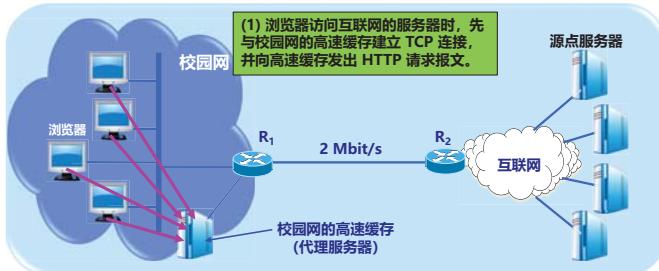
## 2. 代理服务器

- 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。
- 使用高速缓存可减少访问互联网服务器的时延。

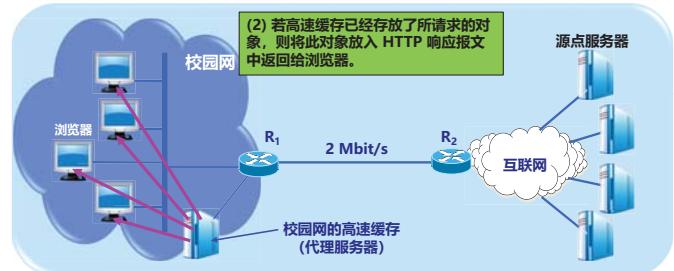
### 不使用高速缓存的情况



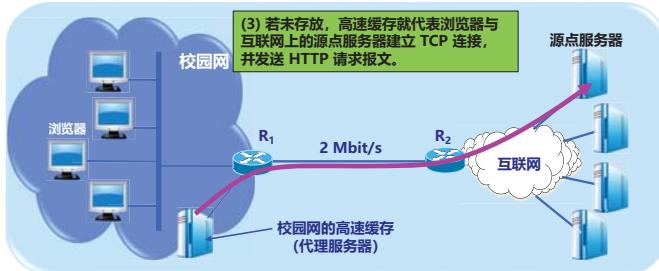
### 使用高速缓存的情况



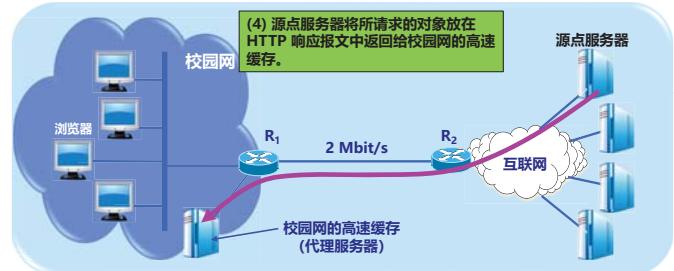
### 使用高速缓存的情况



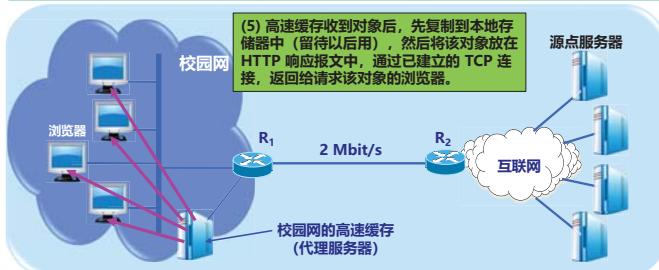
### 使用高速缓存的情况



### 使用高速缓存的情况



### 使用高速缓存的情况



## 3. HTTP 的报文结构

### 两类报文：

- **请求报文：**从客户向服务器的请求。
- **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)，因此报文中每一个字段的值都是一些 ASCII 码串，每个字段的长度都是不确定的。

### 三个组成部分：

- **开始行：**用于区分是请求报文还是响应报文。
- **首部行：**说明浏览器、服务器或报文主体的一些信息。可以有多行，也可以不使用。
- **实体主体：**请求报文中一般不用，响应报文中也可能没有该字段。

## HTTP 的报文结构 (请求报文)



在请求报文中，开始行是请求行。

## HTTP 的报文结构 (请求报文)



方法：对所请求的对象进行的操作，实际上就是一些命令。

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：所请求的资源的 URL。

## HTTP 的报文结构 (请求报文)



版本：HTTP 的版本。

## HTTP 的报文结构 (响应报文)



在响应报文中，开始行是状态行。

版本：HTTP 的版本。状态码：服务器操作完成的状态。短语：解释状态码。

## 状态码：都是三位数字，分为5大类

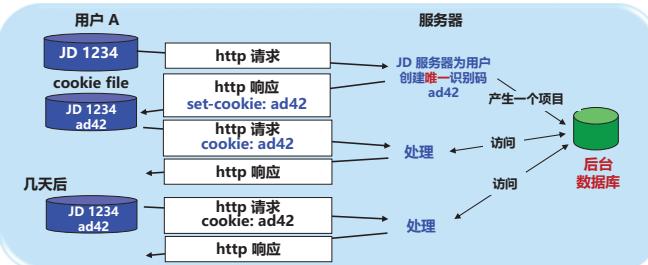
- 1xx 表示通知信息，如请求收到了或正在进行处理。
- 2xx 表示成功，如接受或知道了。
- 3xx 表示重定向，表示要完成请求还必须采取进一步的行动。
- 4xx 表示客户的差错，如请求中有错误的语法或不能完成。
- 5xx 表示服务器的差错，如服务器失效无法完成请求。

响应报文中常见的三种状态行：

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

## 4. 在服务器上存放用户的信息

- 万维网使用 **Cookie** 跟踪在 HTTP 服务器和客户之间传递的**状态信息**。



## HTTPS协议与HTTP协议

- 传输信息安全性不同

http协议中的信息是明文传输，而https协议是具有安全性的ssl加密传输协议，为浏览器和服务器之间的通信加密，确保数据传输的安全。

- 连接方式不同

http协议的连接很简单，是无状态的，而https协议是由SSL + HTTP协议构建的可进行加密传输、身份认证的网络协议。

- 端口不同

http协议使用的端口是80，而https协议使用的端口是443。

- 证书申请方式不同

http协议的证书是免费的，而https协议的证书需要到CA申请。

## 6.4.4 万维网的文档

- 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)。

- 页面制作的标准语言：**HTML**。

- 分为：

- ◆ **静态**万维网文档。内容不会改变。简单。
- ◆ **动态**万维网文档。文档的内容由应用程序动态创建。
- ◆ **活动**万维网文档。由浏览器端改变文档的内容。

### 1. 超文本标记语言 HTML

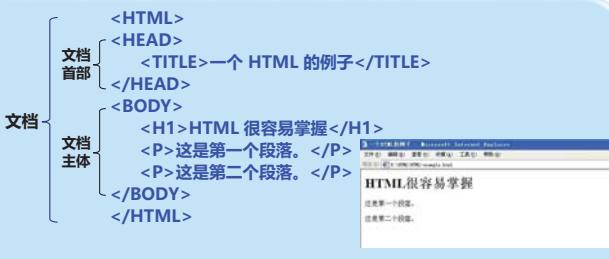
- 超文本标记语言 **HTML** (HyperText Markup Language) 是一种制作万维网页面的**标准语言**，它消除了不同计算机之间信息交流的障碍，是万维网的重要基础 [RFC 2854]。
- 最新 **HTML 5.0** 增加了<audio>和<video>两个标签，实现对多媒体中的音频、视频使用的支持，增加了能够在移动设备上支持多媒体功能。

注意：HTML 不是应用层的协议，它只是万维网浏览器使用的一种语言。

### 1. 超文本标记语言 HTML

- 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，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 文档的**后缀**：.html 或 .htm。

### HTML 文档中标签的用法



### HTML 文档中标签的用法

- <img>：图像标签。
  - ◆ 允许在万维网页面中插入图像。
  - ◆ 一个页面本身带有的图像称为**内含图像** (inline image)。
  - ◆ HTML 标准没有规定该图像的格式。
  - ◆ 大多数浏览器都支持 GIF 和 JPEG 文件。

### HTML 文档中标签的用法

- <a>：链接标签。
  - ◆ 每个链接都有一个起点和终点。
  - ◆ **起点**：说明在万维网页面中的什么地方可**引出**一个链接。可以是一个字或几个字，或是一幅图，或是一段文字。例如：  
`<a href="http://www.tsinghua.edu.cn">清华大学主页</a>`
  - ◆ **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**。这时必须指明链接到的**网站的 URL**。
    - 可以指向本计算机中的某一个文件或本文件中的某处，这种链接方式叫做**本地链接**。这时必须指明链接的**路径**。

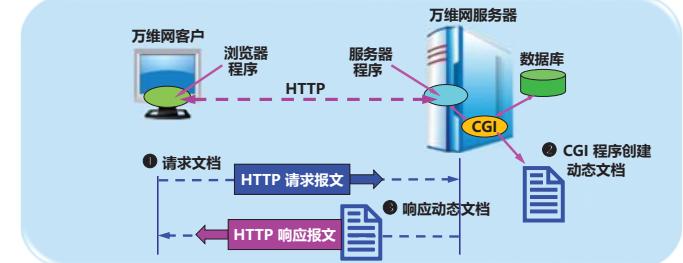
### 2. 动态万维网文档

- **静态文档**：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**：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**服务器端**：文档内容的生成方法不同。从浏览器的角度看，这两种文档并没有区别。

## 万维网服务器功能的扩充

- (1) 增加一个**应用程序**: 处理浏览器发来的数据，并创建动态文档。
- (2) 增加一个**机制**: 使万维网服务器把浏览器发来的数据**传递给这个应用程序**，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。

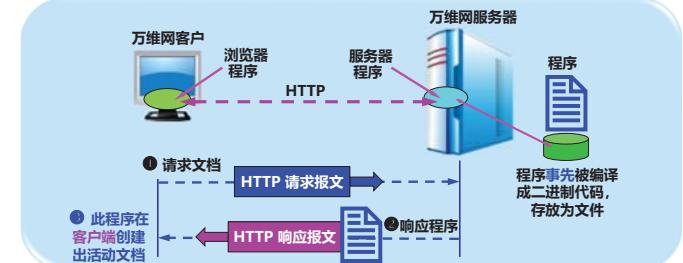
## 扩充了功能的万维网服务器



## 3. 活动万维网文档

- 活动文档 (active document) 技术**: 把屏幕连续更新的工作转移给浏览器端。
- 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



## 6.5 电子邮件

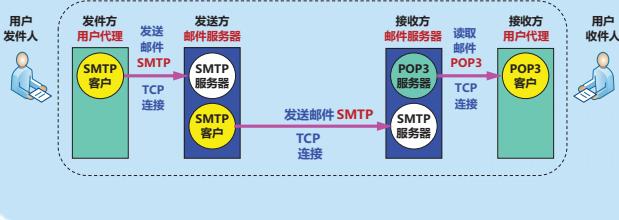
6.5.1	电子邮件概述
6.5.2	简单邮件传送协议 SMTP
6.5.3	电子邮件的信息格式
6.5.4	邮件读取协议 POP3 和 IMAP
6.5.5	基于万维网的电子邮件
6.5.6	通用互联网邮件扩充 MIME

### 6.5.1 电子邮件概述

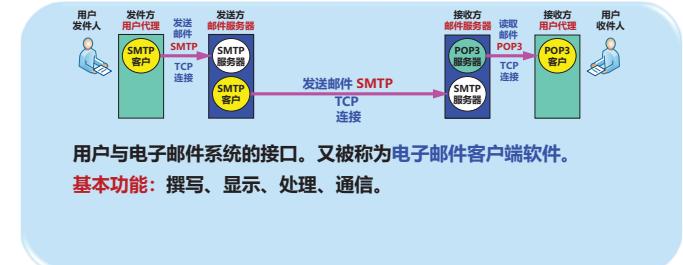
- 电子邮件 (e-mail)**: 指使用电子设备交换的邮件及其方法。
- 优点**: 使用方便，传递迅速，费用低廉，可以传送多种类型的信息（包括：文字信息，声音和图像等）。
- 重要标准**:
  - 简单邮件发送协议: SMTP
  - 互联网文本报文格式
  - 通用互联网邮件扩充 MIME
  - 邮件读取协议: POP3 和 IMAP

## 电子邮件系统的组成: 三个主要构件

### 用户代理, 邮件服务器, 以及邮件发送和读取协议。



### 用户代理 UA (User Agent)



## 邮件服务器 (Mail Server)



- 又被称为邮件传输代理。
- 功能：发送和接收邮件，同时还要向发信人报告邮件传送的情况。
- 按照客户服务器方式工作。

## 邮件发送和读取协议



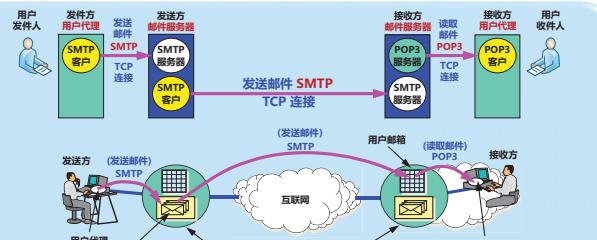
- 邮件发送和读取使用不同的协议。
- 简单邮件发送协议 SMTP：用于在用户代理向邮件服务器或邮件服务器之间发送邮件。
- 邮局协议 POP3：用于用户代理从邮件服务器读取邮件。

## 应当注意

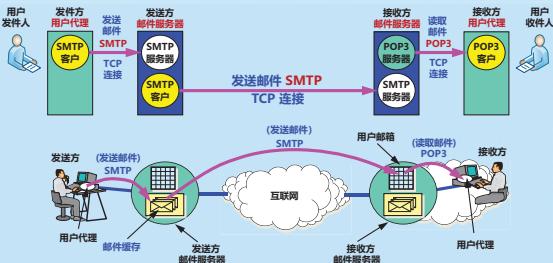


- 邮件服务器必须能够同时充当客户和服务器。
- SMTP 和 POP3（或 IMAP）都使用 TCP 连接可靠地传送邮件。

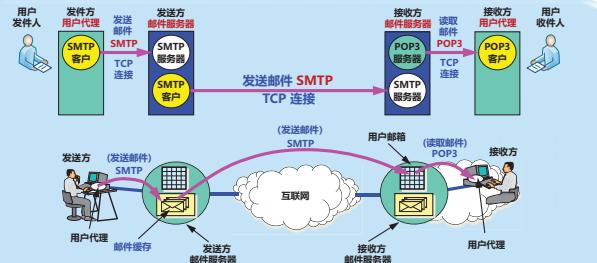
## 发送和接收电子邮件的重要步骤



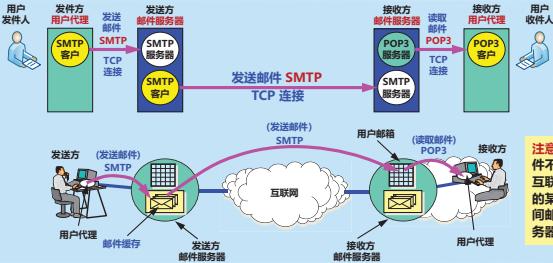
## 发送和接收电子邮件的重要步骤



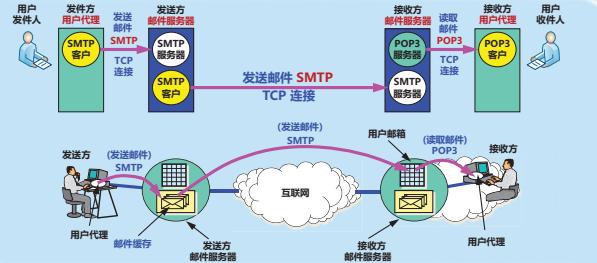
## 发送和接收电子邮件的重要步骤

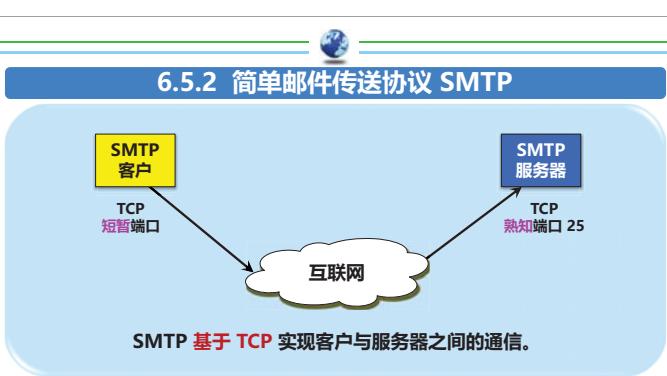
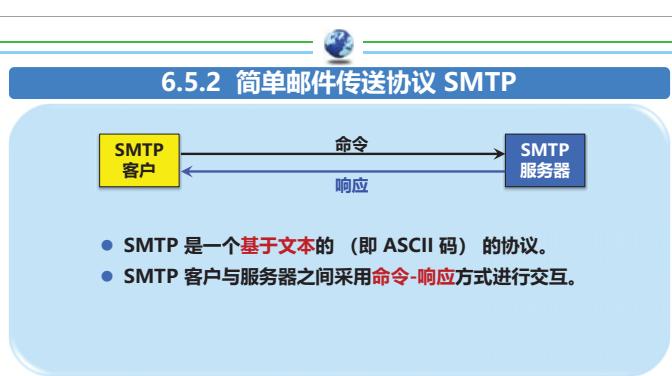
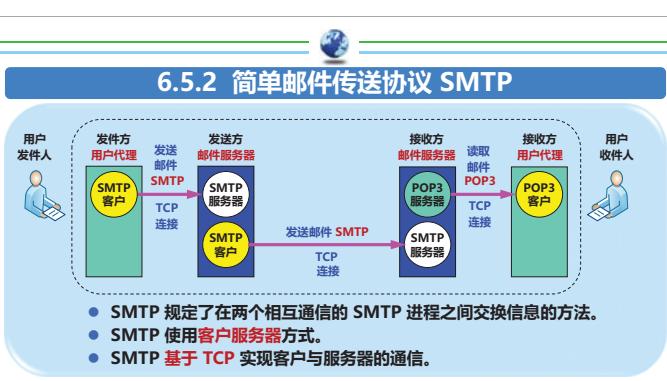
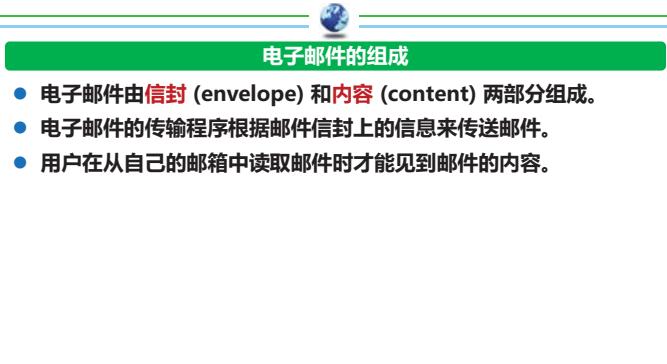
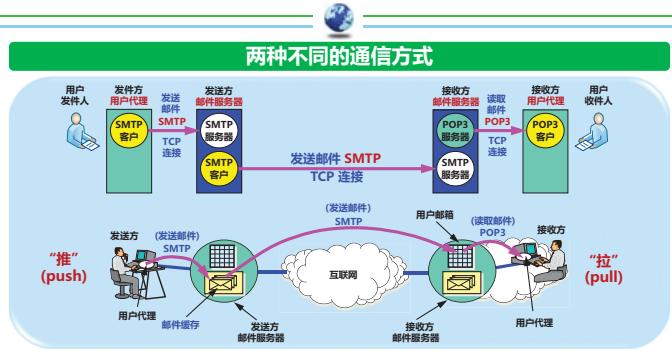


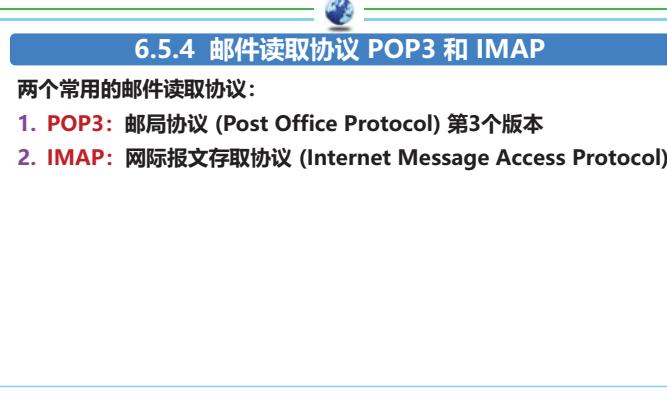
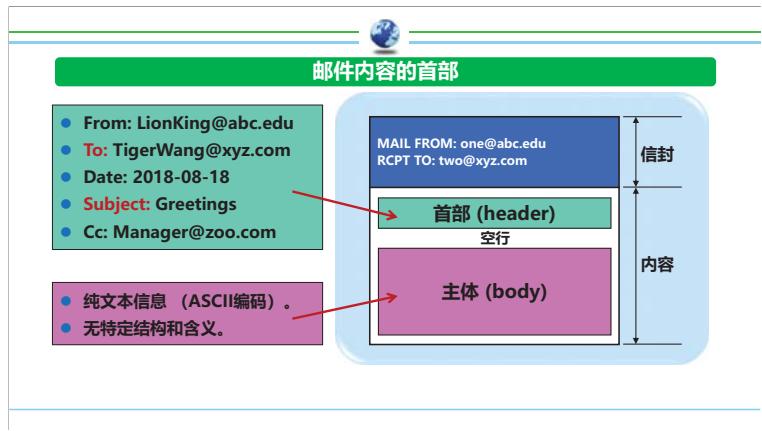
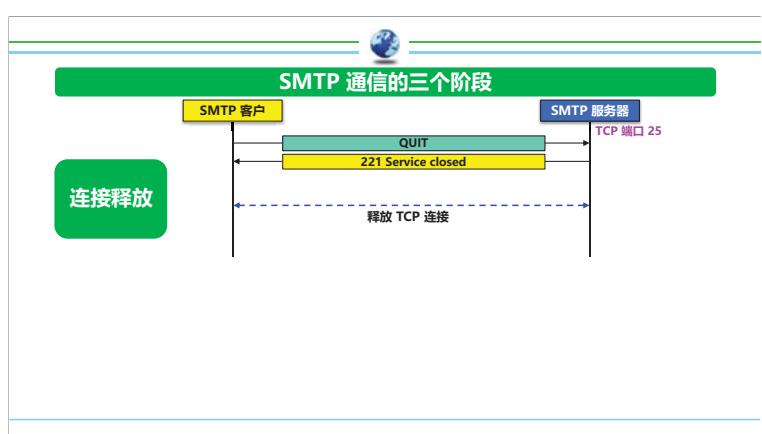
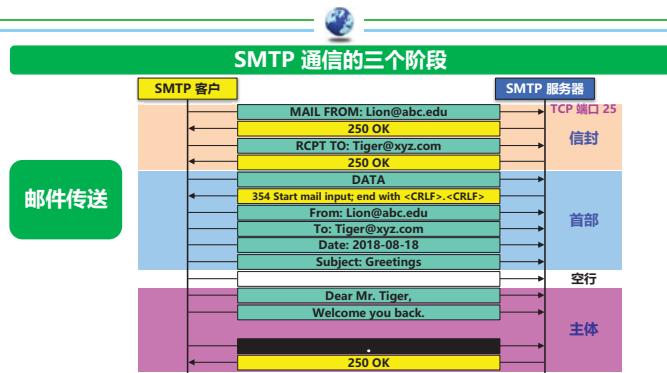
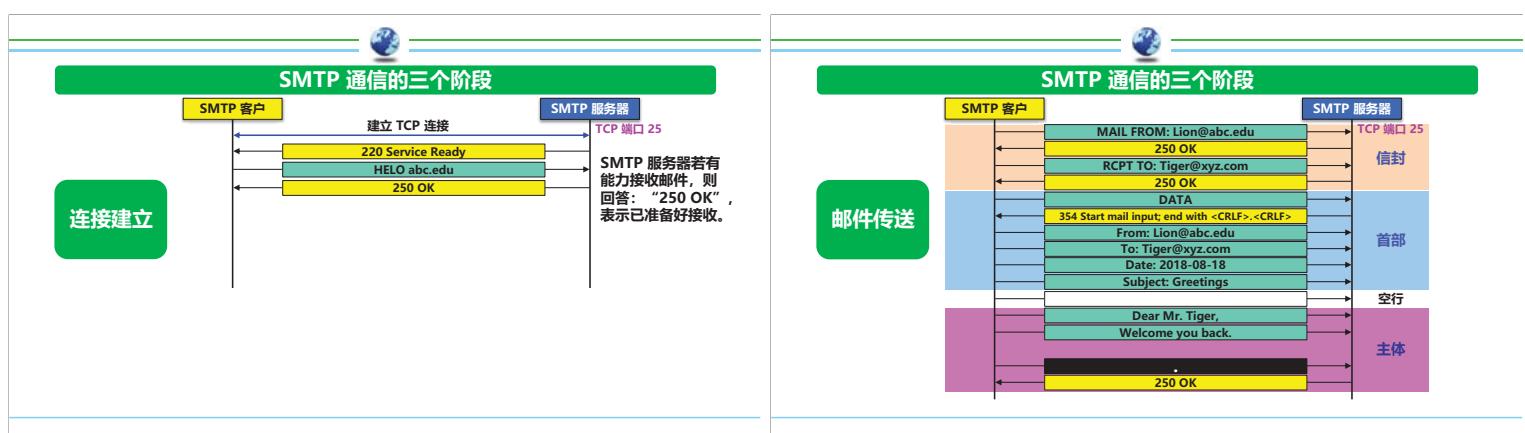
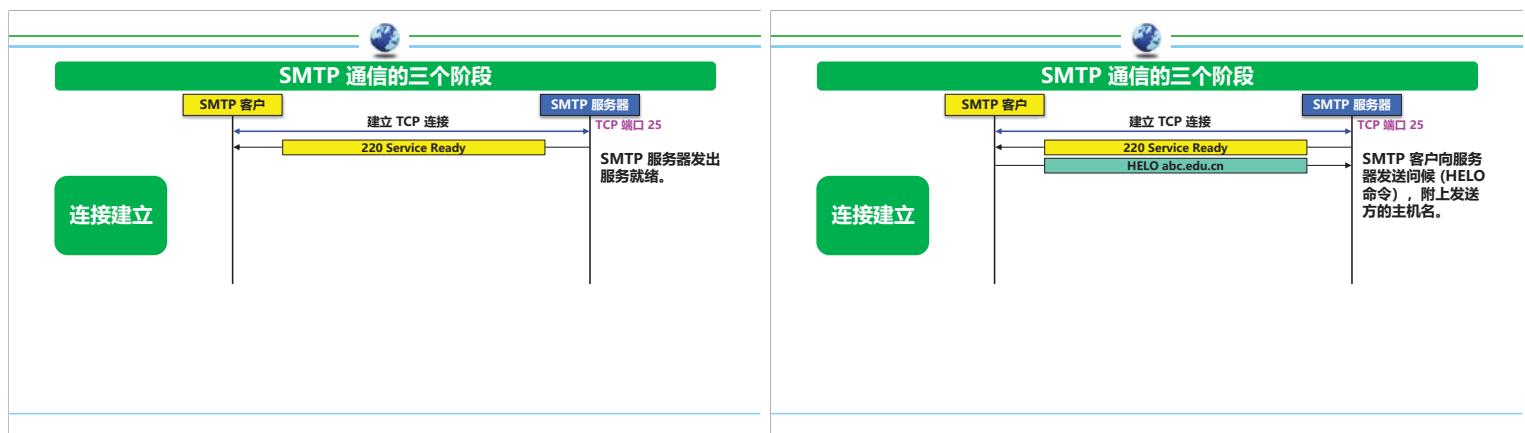
## 发送和接收电子邮件的重要步骤



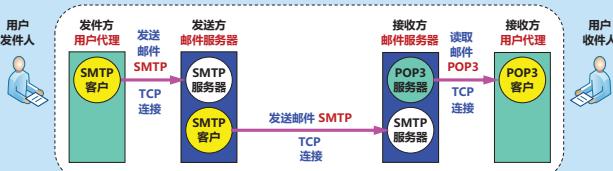
## 发送和接收电子邮件的重要步骤







## POP3 协议



- POP3 使用**客户服务器方式**。
- POP3 基于**TCP** 实现客户与服务器的通信。

## POP3 协议



- POP3 支持**用户鉴别**。
- POP3 服务器**删除**被用户读取了的邮件。

## IMAP 协议



- IMAP 使用**客户服务器方式**。
- IMAP 基于**TCP** 实现客户与服务器的通信。
- IMAP 是一个**联机协议**。

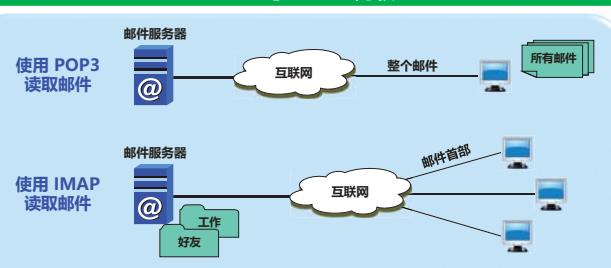
## IMAP 的特点

- 连接后只下载**邮件首部** (部分下载)。
- 用户直接在 IMAP 服务器上**创建和管理文件夹**。
- 用户可以**搜索邮件内容**。
- 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。
- 允许收信人**只读取**邮件中的某一个部分。
- 缺点：要想查阅邮件，必须先联网。

## IMAP 与 POP3 比较

操作位置	操作内容	IMAP	POP3
收件箱	阅读、标记、移动、删除邮件等	客户端与邮箱更新同步	仅在客户端内
发件箱	保存到已发送	客户端与邮箱更新同步	仅在客户端内
创建文件夹	新建自定义的文件夹	客户端与邮箱更新同步	仅在客户端内
草稿	保存草稿	客户端与邮箱更新同步	仅在客户端内
垃圾文件夹	接收并移入垃圾文件夹的邮件	支持	不支持
广告邮件	接收并移入广告邮件夹的邮	支持	不支持

## IMAP 与 POP3 比较



## 必须注意

- 邮件读取协议 POP 或 IMAP 与邮件传送协议 SMTP 完全不同。
- 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

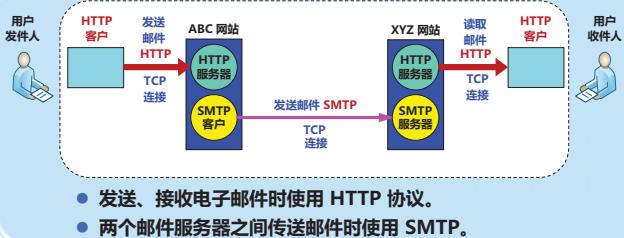
### 用户代理 (UA) 的缺点：

- 必须在计算机中安装用户代理软件。
- 收发邮件不方便。

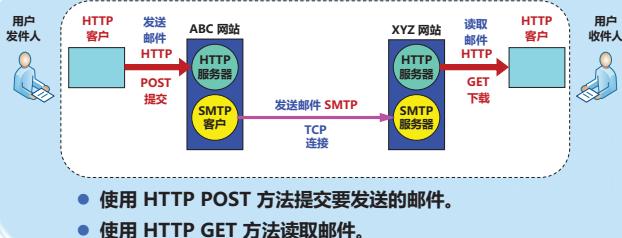
### 万维网电子邮件优点：

- 不需要在计算机中再安装用户代理软件。
- 计算机能联网，就能非常方便地收发电子邮件。
- 界面非常友好。

## 万维网电子邮件



## 万维网电子邮件



## 6.6 动态主机配置协议 DHCP

- 在协议软件中，给协议参数赋值的动作叫做**协议配置**。
- 一个协议软件在使用之前必须是已正确配置的。
- 具体的配置信息取决于协议栈。
- 连接到互联网的计算机的协议软件需要正确配置的参数包括：
  - IP 地址
  - 子网掩码
  - 默认路由器的 IP 地址
  - 域名服务器的 IP 地址

## 动态主机配置协议 DHCP

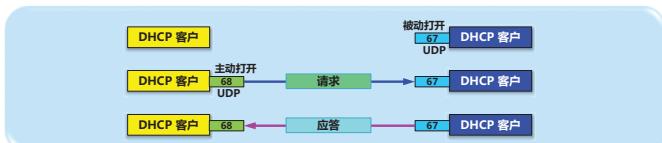
- 动态主机配置协议 DHCP (Dynamic Host Configuration Protocol)** 提供了**即插即用连网 (plug-and-play networking)** 的机制，允许一台计算机加入网络和获取 IP 地址，而不用手工配置。
- DHCP 给运行**服务器**软件、且位置固定的计算机指派一个**永久地址**，给运行**客户端**软件的计算机分配一个**临时地址**。

## DHCP 使用客户服务器方式

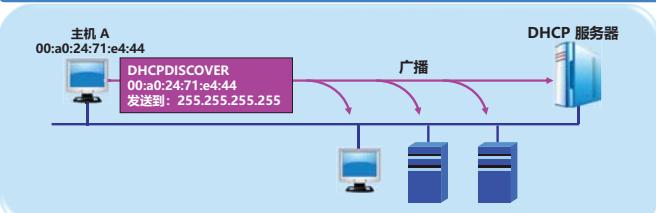
- 需要 IP 地址的主机在启动时就向 DHCP 服务器**广播发送**发现报文 (DHCPDISCOVER)，这时该主机就成为 DHCP 客户。
- 本地网络上所有主机都能收到此广播报文，但只有 DHCP 服务器才回答此广播报文。
- DHCP 服务器先在其数据库中查找该计算机的配置信息。若找到，则返回找到的信息。若找不到，则从服务器的 IP 地址池 (address pool) 中取一个地址分配给该计算机。DHCP 服务器的回答报文叫做**提供报文 (DHCPoffer)**。

## DHCP 工作方式

- DHCP 使用**客户服务器**方式，采用**请求/应答**方式工作。
- DHCP 基于 UDP 工作，DHCP 服务器运行在 67 号端口，DHCP 客户运行在 68 号端口。

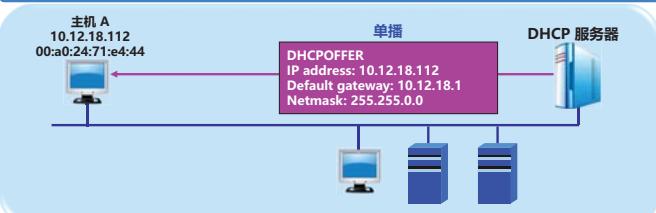


## DHCP 工作方式



主机 A 00:a0:24:71:e4:44  
DHCPDISCOVER  
00:a0:24:71:e4:44  
发送到: 255.255.255.255

## DHCP 工作方式

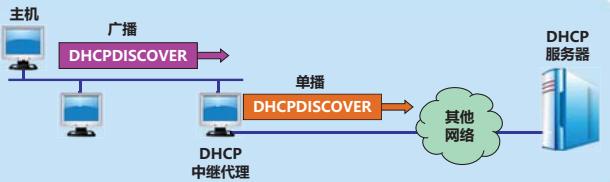


DHCP 服务器回答提供报文 (DHCPOFFER) (**单播**)，提供 IP 地址等配置信息。

## DHCP 中继代理 (relay agent)

- 问题：每个网络上都需要有 DHCP 服务器吗？
- 答案：不需要，因为会使 DHCP 服务器的数量太多。
- 问题：若没有 DHCP 服务器，如何自动获得地址？
- 解决：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。

## DHCP 中继代理以单播方式转发发现报文

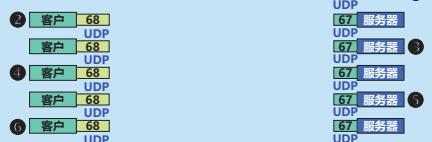


- DHCP 中继代理收到主机广播发送的发现报文后，就以单播方式向 DHCP 服务器转发此报文，并等待其回答。
- 收到 DHCP 服务器回答的提供报文后，DHCP 中继代理再将其发回给主机。

## 租用期 (lease period)

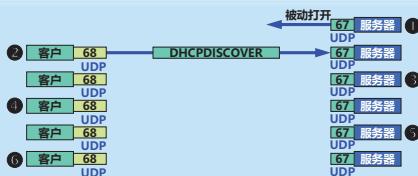
- DHCP 服务器分配给 DHCP 客户的 IP 地址的临时的，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。DHCP 协议称这段时间为租用期。
- 租用期的数值应由 DHCP 服务器自己决定。
- DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。

## DHCP 协议的工作过程



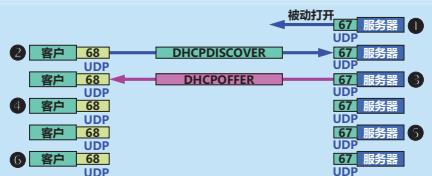
①: DHCP 服务器被动打开 UDP 端口 67，等待客户端发来的报文。

## DHCP 协议的工作过程



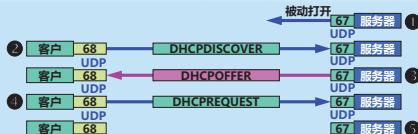
②: DHCP 客户从 UDP 端口 68 发送 DHCP 发现报文 DHCPDISCOVER。

## 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 DHCPOFFER，因此 DHCP 客户可能收到多个 DHCPOFFER 报文。

## DHCP 协议的工作过程



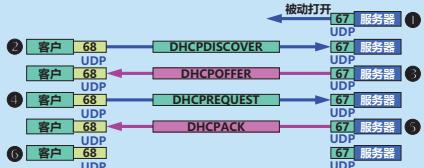
④: DHCP 客户从几个 DHCP 服务器中选择其中一个，并向所选择的 DHCP 服务器发送 DHCP 请求报文 DHCPREQUEST。

## DHCP 协议的工作过程



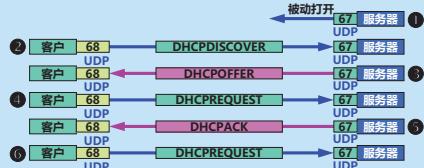
⑤: 被选择的 DHCP 服务器发送确认报文 DHCPACK，DHCP 客户可开始使用得到的临时 IP 地址了，进入已绑定状态。

## 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  $0.875T$ 。当超时时间到时，就要请求更新租用期。

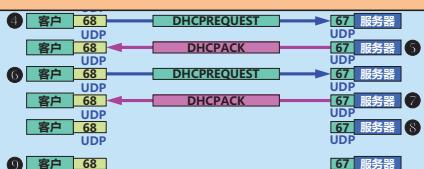
## DHCP 协议的工作过程



①：租用期过了一半 ( $T_1$  时间到)，DHCP 发送请求报文 DHCPREQUEST，要求更新租用期。

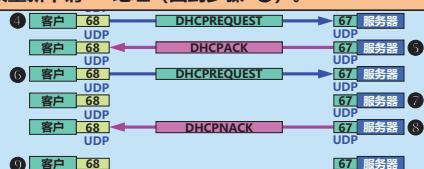
## DHCP 协议的工作过程

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。



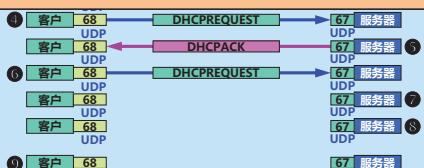
## DHCP 协议的工作过程

⑧：DHCP 服务器若不同意，则发回否认报 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）。



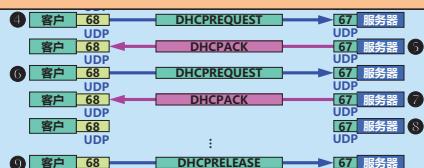
## DHCP 协议的工作过程

若 DHCP 服务器不响应步骤 ① 的请求报文 DHCPREQUEST，则在租用期过了  $87.5\%$  时 ( $T_2$  时间到)，DHCP 客户必须重新发送请求报文 DHCPREQUEST（重复步骤 ①），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。



DHCP客户机在网络中广播一个DHCPDiscover包以请求IP地址，所以此过程也称为DHCPDiscover。

DHCPDiscover包的源IP地址为0.0.0.0，目的IP地址为255.255.255.255，该包还包含客户机的MAC地址和计算机名，以使DHCP服务器能够确定哪个客户机发送该请求。



服务器响应：当DHCP服务器接收到客户机请求IP地址的信息时，就在自己的IP地址库中查找是否有合法的IP地址提供给客户机，如果有，DHCP服务器就将此IP地址做上标记，广播一个DHCPOffer包（即DHCPOOffer）。

因为DHCP客户机还没有IP地址，所以由DHCP服务器发送广播消息



客户机选择IP地址：DHCP客户机从收到的第一个DHCPOffer包中选择IP地址，并将DHCPRequest包广播到所有DHCP服务器，表明它接受提供的内容（即DHCPRequest）。如果客户机接受了IP地址，则发出IP地址的DHCP服务器将该地址保留，该地址就不能提供给另一个DHCP客户机；如果那些DHCPOffer包被拒绝，DHCP服务器则取消提供并保留其IP地址以用于下一个IP租约请求。

**注意：**在客户机选择IP地址的过程中，虽然客户机选择了IP地址，但是还没有配置IP地址，所以源地址仍为0.0.0.0，在一个网络中可能有几个DHCP服务器，所以DHCP客户机任然广播发出DHCPRequest包。



## 6.7 简单网络管理协议 SNMP

6.7.1	网络管理的基本概念
6.7.2	管理信息结构 SMI
6.7.3	管理信息库 MIB
6.7.4	SNMP 的协议数据单元和报文

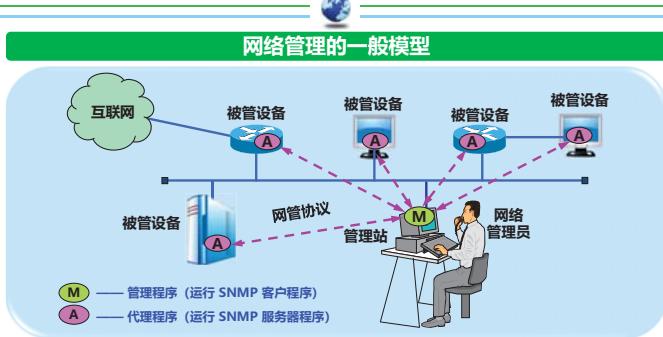
### 6.7.1 网络管理的基本概念

- **网络管理**包括对硬件、软件和人力的使用、综合与协调，以便对网络资源进行监视、测试、配置、分析、评价和控制，这样就能以合理的价格满足网络的一些需求，如实时运行性能，服务质量等。
- 网络管理常简称为**网管**。

### 网络管理的五大功能

- 故障管理：故障检测、隔离和纠正。
- 配置管理：初始化网络、并配置网络。
- 计费管理：记录网络资源的使用。
- 性能管理：估价系统资源的运行状况及通信效率等。
- 网络安全管理：对授权机制、访问控制、加密和加密关键字的管理。

### 网络管理的一般模型

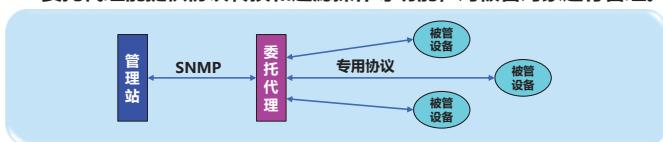


### SNMP 的基本功能

- 最重要的指导思想：**尽可能简单**。
- 基本功能：
  1. 监视网络性能
  2. 检测分析网络差错
  3. 配置网络设备等。

### SNMP 的管理站和委托代理

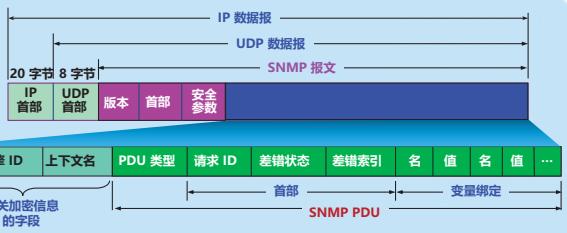
- 整个系统必须有一个**管理站**。
- 若网络元素使用的不是 SNMP 而是另一种网络管理协议，SNMP 协议就无法控制该网络元素。这时可使用**委托代理 (proxy agent)**。
- 委托代理能提供协议转换和过滤操作等功能，对被管对象进行管理。



### SNMP 使用无连接的 UDP

- 运行**代理程序的服务器端**用 UDP 熟知端口 161 接收 get 或 set 报文，发送响应报文。与熟知端口通信的**客户端**使用**临时端口**。
- 运行**管理程序的客户端**则使用 UDP 熟知端口 162 来接收来自各代理的 trap 报文。

## SNMP 的报文格式



### 6.8 应用进程跨越网络的通信

#### 6.8.1

#### 系统调用和应用编程接口

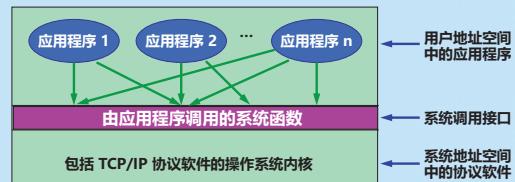
#### 6.8.2

#### 几种常用的系统调用

### 6.8.1 系统调用和应用编程接口

- 大多数操作系统使用**系统调用 (system call)**的机制在应用程序和操作系统之间传递控制权。
- 对程序员来说，每一个系统调用和一般程序设计中的函数调用非常相似，只是系统调用是将控制权传递给了操作系统。

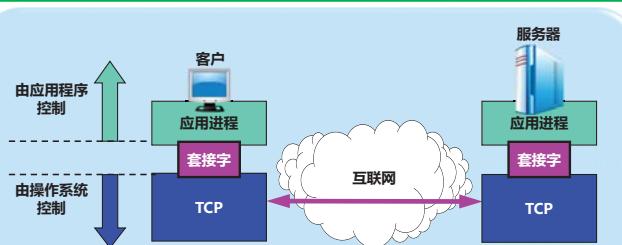
### 多个应用程序使用系统调用的机制



## 应用编程接口 API

- 系统调用接口**实际上就是应用进程的控制权和操作系统的控制权进行转换的一个接口。
- 使用系统调用之前要编写一些程序，特别是需要设置系统调用中的许多参数，因此这种系统调用接口又称为**应用编程接口 API** (Application Programming Interface)。

## 应用进程通过套接字接入到网络



### 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

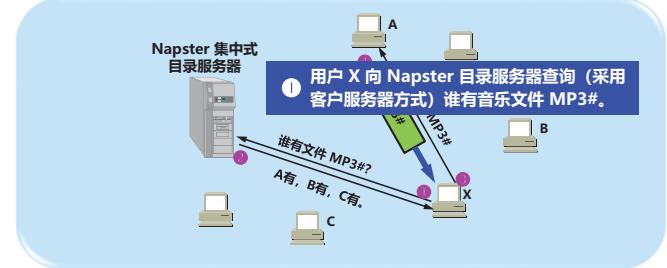
在 P2P 工作方式下，所有的音频/视频文件都是在普通的互联网用户之间传输。



### 6.9.1 具有集中目录服务器的 P2P 工作方式

- Napster 最早使用 P2P 技术，提供免费下载 MP3 音乐。
- Napster 将所有音乐文件的索引信息都集中存放在 Napster 目录服务器中。
- 使用者只要查找目录服务器，就可知道应从何处下载所要的 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。
- Napster 的文件传输是分散的，文件的定位则是集中的。

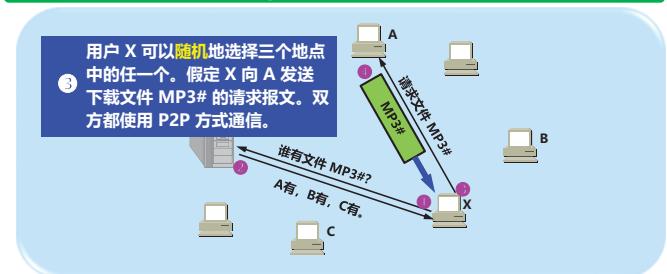
### Napster 的工作过程



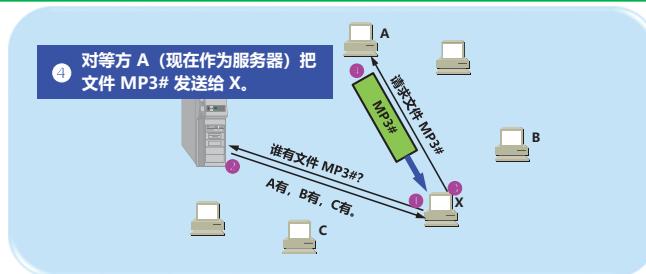
### Napster 的工作过程



### Napster 的工作过程



### Napster 的工作过程



### 集中式目录服务器的缺点

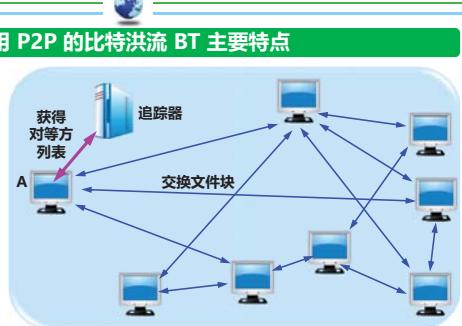
- 可靠性差。
- 会成为性能的瓶颈。

### 6.9.2 具有全分布式结构的 P2P 文件共享程序

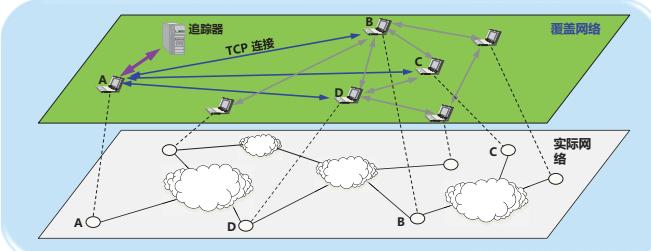
- Gnutella 是第二代 P2P 文件共享程序，采用全分布方法定位内容的 P2P 文件共享应用程序。
- Gnutella 与 Napster 最大的区别：不使用集中式的目录服务器，而是使用洪泛法在大量 Gnutella 用户之间进行查询。
- 为了不使查询的通信量过大，Gnutella 设计了一种有限范围的洪泛查询，减少了倾注到互联网的查询流量，但也影响到查询定位的准确性。
- 第三代 P2P 文件共享程序采用分散定位和分散传输技术。例如 KaZaA, 电骡 eMule, 比特洪流 BT (Bit Torrent) 等。

### 使用 P2P 的比特洪流 BT 主要特点

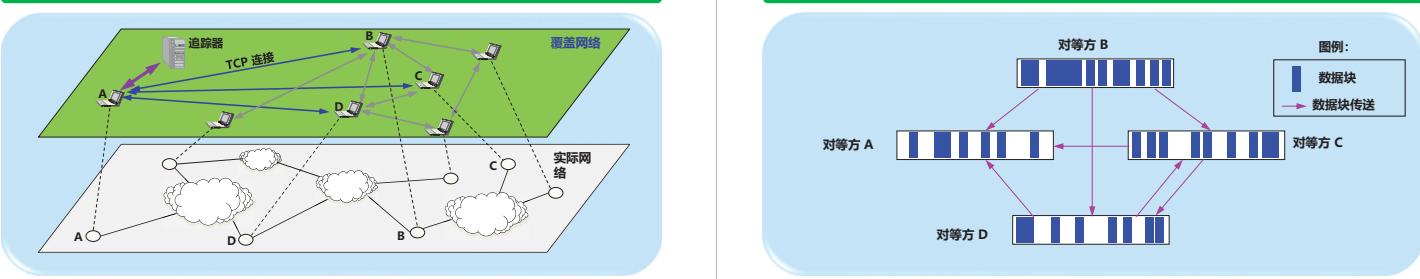
- BitTorrent 所有对等方集合称为一个洪流 (torrent)。
- 下载文件的数据单元为长度固定的文件块 (chunk)。
- 基础设施结点，叫做追踪器 (tracker)。
- A 和对等方建立了 TCP 连接。所有与 A 建立了 TCP 连接的对等方为相邻对等方 (neighboring peers)。



### 相邻关系是逻辑的，对等方的数目是动态变化的



### 对等方之间互相传送文件数据块

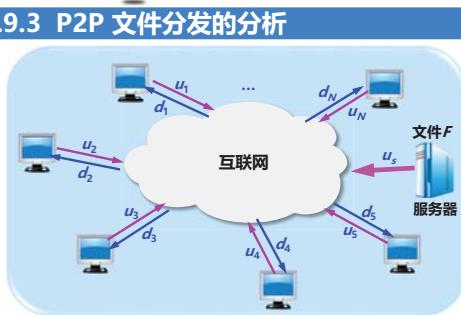


### BT 协议

- 问题：**哪些文件块是首先需要向其相邻对等方请求的？
- 方法：**A 使用**最稀有的优先** (rarest first) 的技术，首先向其相邻对等方请求对应的文件块。
- 稀有：**如果 A 所缺少的文件块在相邻对等方中的副本很少，那就是“很稀有的”。
- 问题：**在很多向 A 请求文件块的相邻对等方中，A 应当向哪些相邻对等方发送所请求的文件块？
- 方法：**凡当前以**最高数据率**向 A 传送文件块的某相邻对等方，A 就**优先**把所请求的文件块传送给该相邻对等方。

### 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。



### 客户-服务器方式下分发的最短时间分析

- 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分析

- 初始**服务器文件分发的最少时间不可能小于  $F/u_s$ ；
- 下载**文件分发的最少时间不可能小于  $F/d_{min}$ ；
- 上载**文件分发的最少时间不可能小于  $NF/u_t$ ，其中是  $u_t$  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：

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

### 时间比较

- 设所有的对等方的上传速率都是  $u$ ，并且  $F/u = 1$  小时。
- 设服务器的上传速率  $u_s = 10u$ 。
- 当  $N = 30$  时，
  - P2P 方式**：最少时间的下限是 0.75 小时 < 1 小时（不管 N 多大）。
  - 客户服务器方式**：最少时间是 3 小时。

### 6.9.4 在 P2P 对等方中搜索对象

- Napster 在一个集中式目录服务器中构建查找数据库，简单，但性能上有**瓶颈**。
- Gnutella 是一种采用全分布方法定位内容的 P2P 文件共享应用程序，它解决了集中式目录服务器所造成的瓶颈问题。但 Gnutella 是在非结构化的覆盖网络中采用查询洪泛的方法进行查找，因此查找的**效率较低**。



#### 6.9.4 在 P2P 对等方中搜索对象

- 现在广泛使用的索引和查找技术叫做**分布式散列表 DHT** (Distributed Hash Table)。
- DHT 也可译为**分布式哈希表**, 由大量对等方共同维护。
- 广泛使用的 **Chord** 算法是美国麻省理工学院于 2001 年提出的。