

第2章 应用层

任课老师：周军海

Email:rj_zjh@hnu.edu.cn

学习目标和主要内容

学习目标

熟练掌握网络应用的原理和实现方面的知识。

主要内容

- ✓ **应用层概念：**应用层协议、客户机与服务器、进程、套接字和运输层接口。
- ✓ **应用程序：**Web、电子邮件、域名系统DNS及P2P对等文件共享。
- ✓ **开发网络应用程序的方法：**套接字API编程。

第2章 应用层

网络应用是计算机网络的重要功能之一。

- ✓ 20世纪80年代：基于文本的电子邮件、文件传输、文本聊天等等。
- ✓ 20世纪90年代：Web应用、IP电话、视频会议等。
- ✓ 20世纪末：即时讯息、P2P对等文件共享。
- ✓ 最近：新一代社交网络应用

本章内容

2.1 应用层协议原理

2.2 Web和HTTP

2.3 FTP

2.4 电子邮件

2.5 DNS

2.6 P2P 文件共享

2.7 TCP的套接字编程

2.8 UDP的套接字编程

2.9 构建一个 Web服务器

2.10 小结

流行的网络应用程序

- E-mail
- Web
- 即时讯息
- 远程登录
- P2P文件共享
- 多用户网络游戏
- 流式存储视频片段
- 因特网电话
- 实时视频会议
- 大规模并行计算
- 社交网络



编制应用程序

网络应用程序的研发要点

- ❑ 写出能够分别在不同端系统运行，并通过网络相互通信的程序。

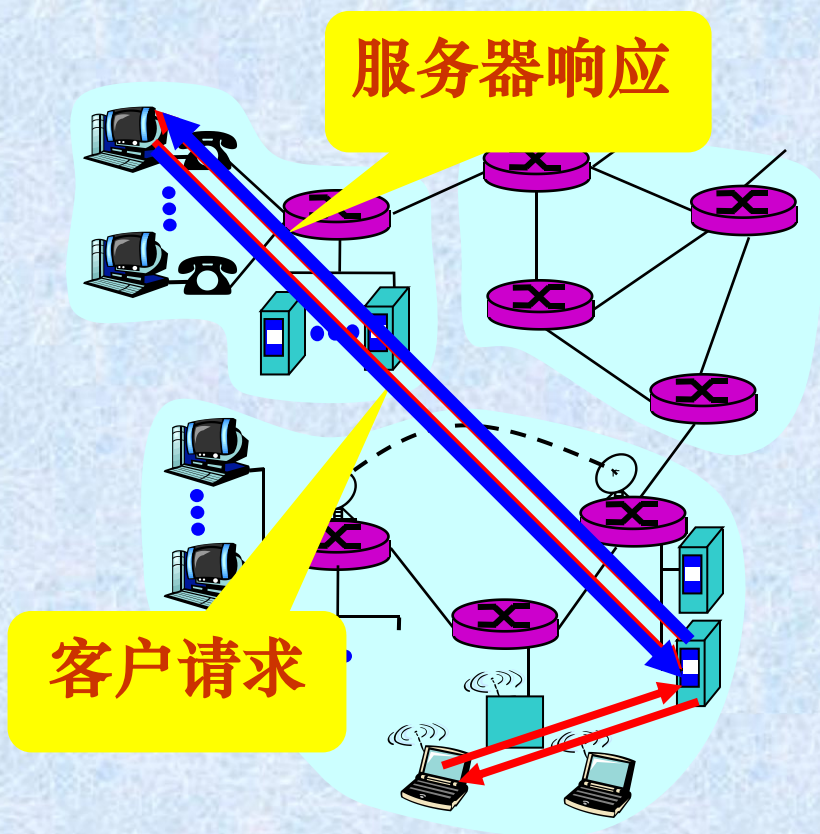
如Web应用程序，由两个可以相互通信的程序组成

- ✓ 浏览器程序：运行在用户主机上；
- ✓ Web服务器程序：运行在Web服务器主机上。
- ❑ 应用程序软件只在端系统运行，不需在网络核心设备上运行。
 - 网络核心设备无应用层，只有较低层。

2.1.1 网络应用程序体系结构

- ❑ 应用程序体系结构：规定如何在各种端系统上组织应用程序，由研发者设计。
- ❑ 三种类型：
 - ✓ 客户机/服务器
 - ✓ 对等 (P2P)
 - ✓ 客户机/服务器与P2P的混合

1、客户机/服务器体系结构



服务器：

- 总是打开
- 为多个客户机请求提供服务
- 永久的IP地址
- 可扩展为服务器场（主机群集）

客户机：

- 总是打开或间歇打开
- 向服务器发出请求
- 具有动态的IP地址
- 彼此之间不直接通信

如Web应用程序：总是打开的Web服务器为运行在客户机主机上的浏览器的请求提供服务（接收客户机请求，并发送响应结果）。

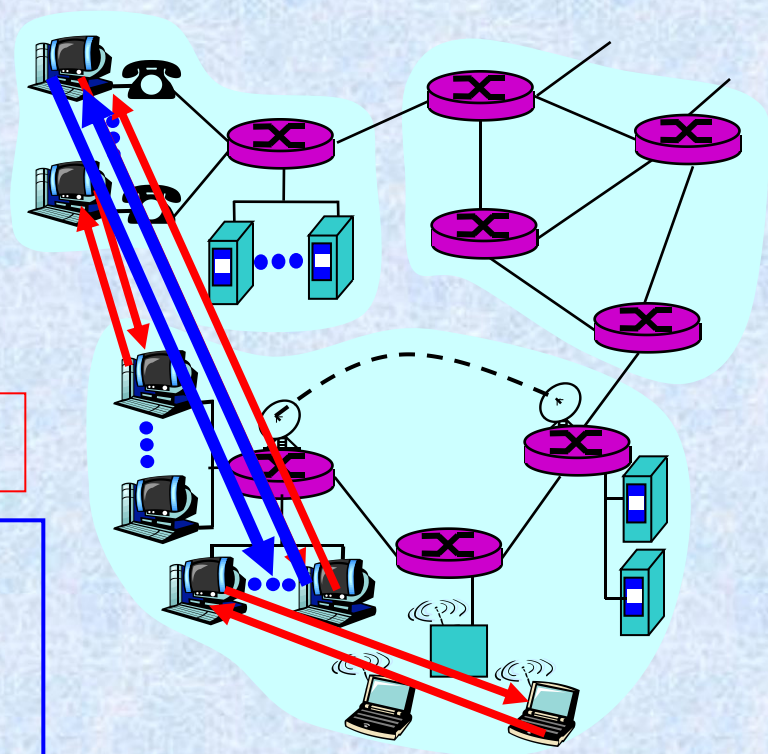
2、纯P2P体系结构（peer-to-peer）

- 无（最少）打开的服务器
- 任意端系统（对等方）可以直接通信
- 对等方间歇地连接，IP地址不固定
- 例：文件分发、因特网电话等。

可扩展度高、难以管理

未来P2P应用面临的三个主要挑战：

- ISP友好：受制于“非对称”宽带应用
- 安全性
- 激励



3、客户机/服务器与P2P的混合

□ **Napster:** MP3文件共享应用程序。

- P2P: **对等方直接交换**MP3文件

- 服务器注册/定位:

 - 对等方在中心服务器上**注册内容**

 - 对等方查询相同的中心服务器以**定位内容**

□ **即时讯息:**

- P2P: 两个用户直接聊天

- 服务器检测/定位:

 - 用户在线时, 向中心服务器**注册其IP地址**

 - 用户联系中心服务器以**找到聊天伙伴的IP地址**

2.1.2 进程通信

- 进程 (process)：在主机上运行的程序。
- 进程通信：
 - ✓ 同一主机中两个进程间的通信：由操作系统控制；
 - ✓ 不同主机中进程间的通信：通过网络交换报文进行。
 - ◆ 发送进程：产生报文并向网络发送；
 - ◆ 接收进程：接收报文，并回送报文。

如图2-1。

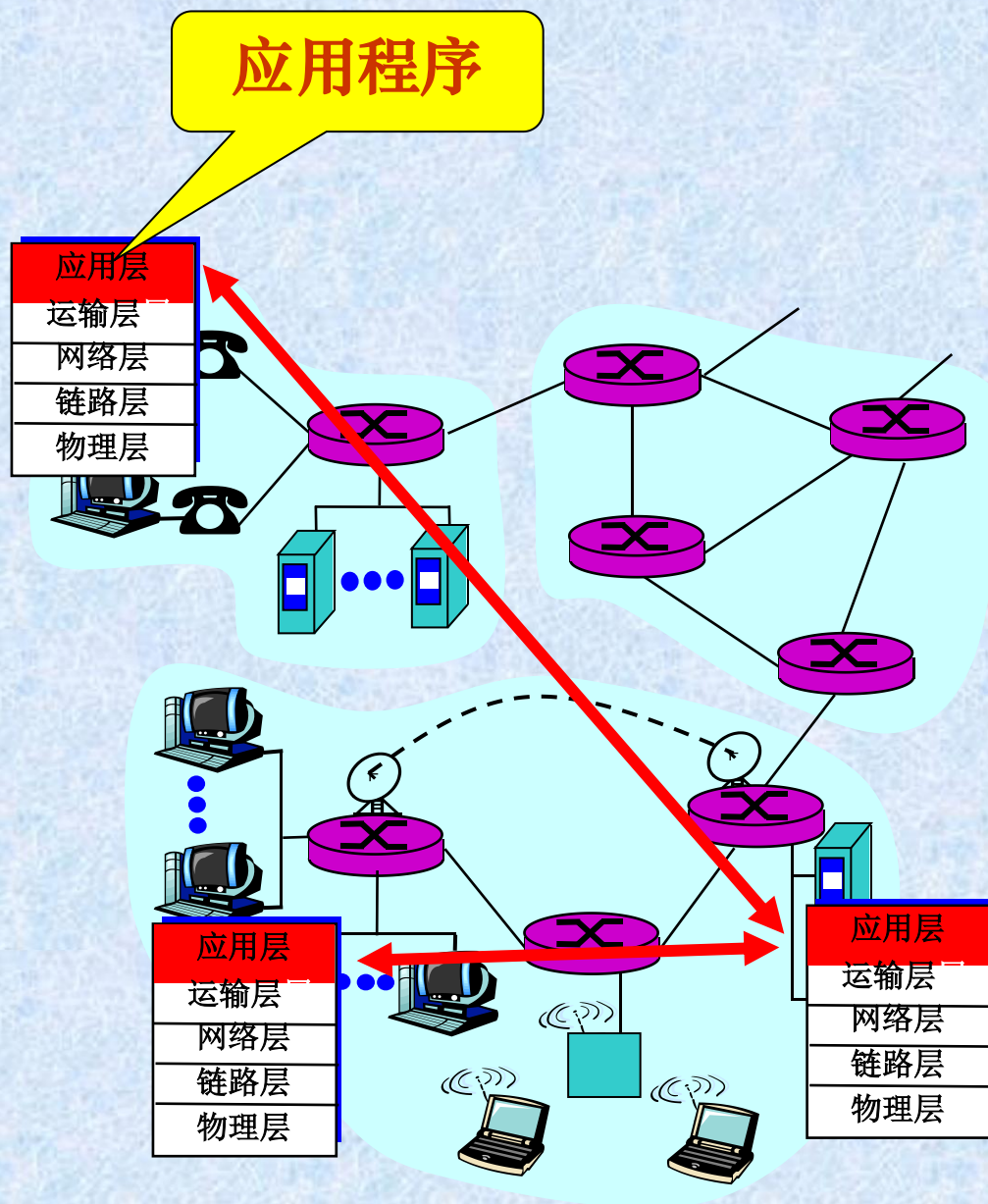


图2-1 在应用层的端系统之间的网络应用的通信

1、客户机和服务器进程

- 网络应用程序由**成对的进程组成**，并通过网络相互发送报文。如图2-1
- 根据功能分别标示为**客户机**和**服务器**
- ✓ **客户机进程**：发起通信的进程。
- ✓ **服务器进程**：等待其他进程联系的进程。

例如：Web应用程序中，一个客户机浏览器进程向某个Web服务器进程发起联系，交换报文。

又如：一个P2P文件共享系统中，文件从一个对等方中的进程传输到另一个对等方中的进程。

说明：

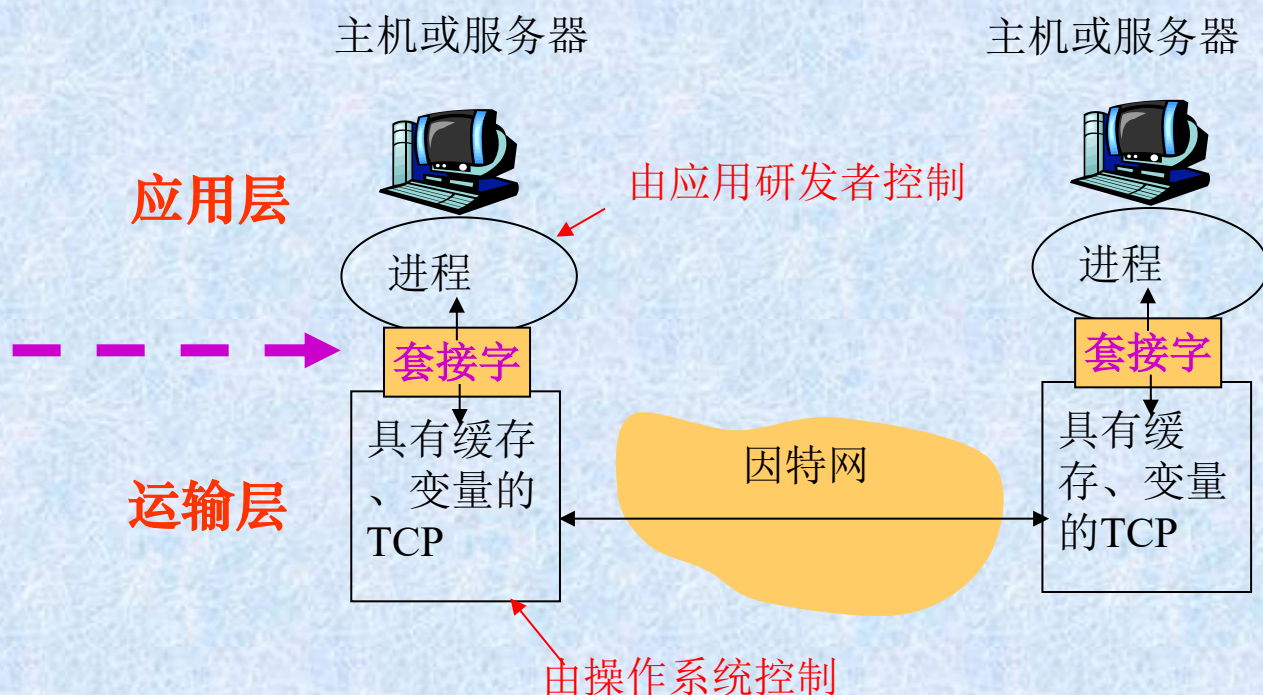
P2P结构的应用程序也可分别看成是客户机进程或服务器进程。

如，对等方A（**客户机**）请求对等方B（**服务器**）发送某个文件。

2、套接字(socket)

✓ **套接字**：同一台主机内**应用层与运输层**之间的接口。

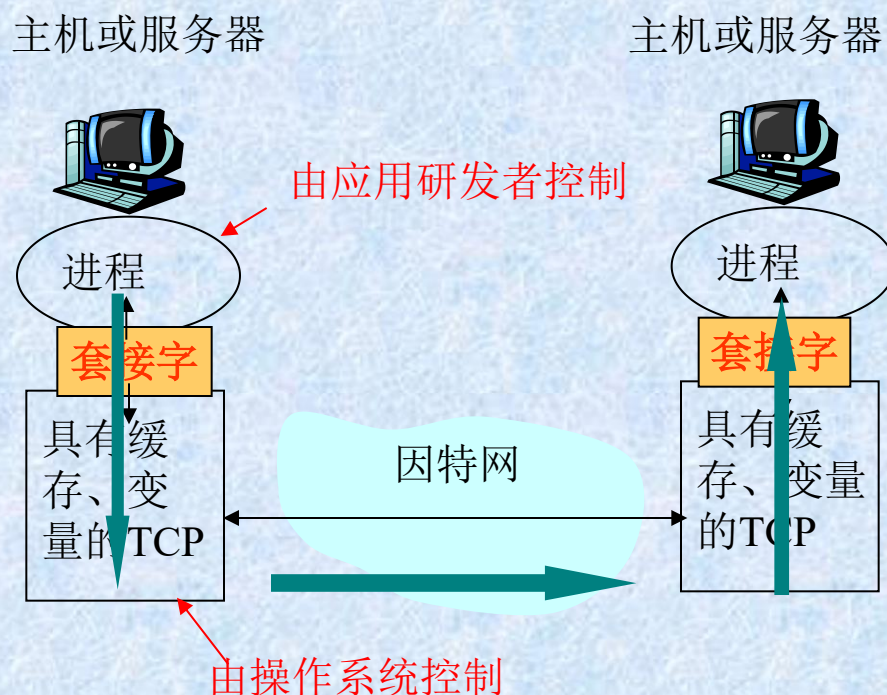
也叫应用程序和网络之间的**应用程序接口API**，是在网络上建立网络应用程序的**可编程接口**。



进程与套接字关系

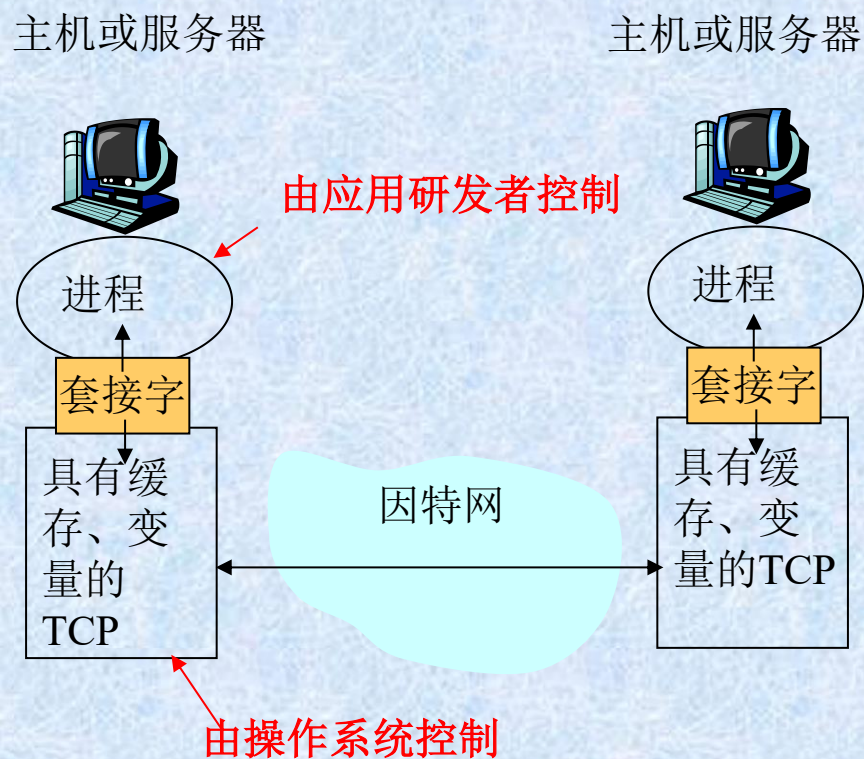
- ✓ 进程类似**房子**，套接字是进程的**门**。
- ✓ 进程通过**套接字**在网络上发送和接收报文。

- ✓ **发送进程**：把报文推出门（套接字）。
- ✓ **传送报文**：通过下面网络把报文传送到目的进程门口。
- ✓ **接收进程**：通过其门（套接字）接收报文



说明

- ✓ 应用程序开发者**可以控制套接字应用层端**的全部；
- ✓ 对套接字的**运输层端几乎不能控制**（只能选择运输层协议、设定几个运输层参数等）。
- ✓ 应用程序开发者选择了一个运输层协议，则应用程序就建立在由该协议提供的运输层服务之上。如TCP协议。



3、进程寻址

如何识别
进程？

- ✓ 主机上的进程可以有多个。
- ✓ 网络中有多个主机，每个主机上有多个进程。

□ 进程识别信息：表示哪台主机上的哪一个进程。

源主机上的进程向目的主机上的进程发送报文时，应带有接收进程的识别信息（标识）。

□ 进程寻址：根据进程识别信息找到相应进程。

确定主机 → 确定进程

进程识别信息（两部分）

- ❑ 主机名称或地址：网络中的哪一个主机。

因特网中，用IP地址标识（32位/128位，全球惟一）。

- ❑ 进程的标识：主机中的哪一个进程。

因特网中，采用端口号标识(port number)。

- ✓ 常用的应用程序被指派固定的端口号（周知端口）。

如，Web服务进程(HTTP协议)：80

邮件服务进程(SMTP协议)：25

- ✓ 创建一个新的网络应用程序时，必须分配一个新的端口号，不能重复。

4、用户代理（user agent）

是用户与网络应用程序之间的接口。

如：

✓ Web应用的用户代理：是一些浏览器软件。

一个通过套接字收发报文，并提供用户接口的进程。

✓ 电子邮件应用程序用户代理：是“邮件阅读器”。

允许用户进行邮件的撰写和阅读。

2.1.3 应用层协议

定义了运行在不同端系统上的应用程序**进程间传递报文的格式和方式**。

□ 具体内容：

- ✓ **各种报文类型及其语法**：报文类型：如请求报文和响应报文；报文中的各个字段及描述；
- ✓ **字段的语义**：字段包含信息的含义；
- ✓ **时序**：进程何时、如何发送报文及对报文进行响应的规则。

说明

□ **公共领域协议**：由标准文档RFC定义，如HTTP。

专用层协议：如P2P使用的协议。

□ **应用层协议是网络应用的一部分。**

如Web应用，客户机从Web服务器获得“文档”。

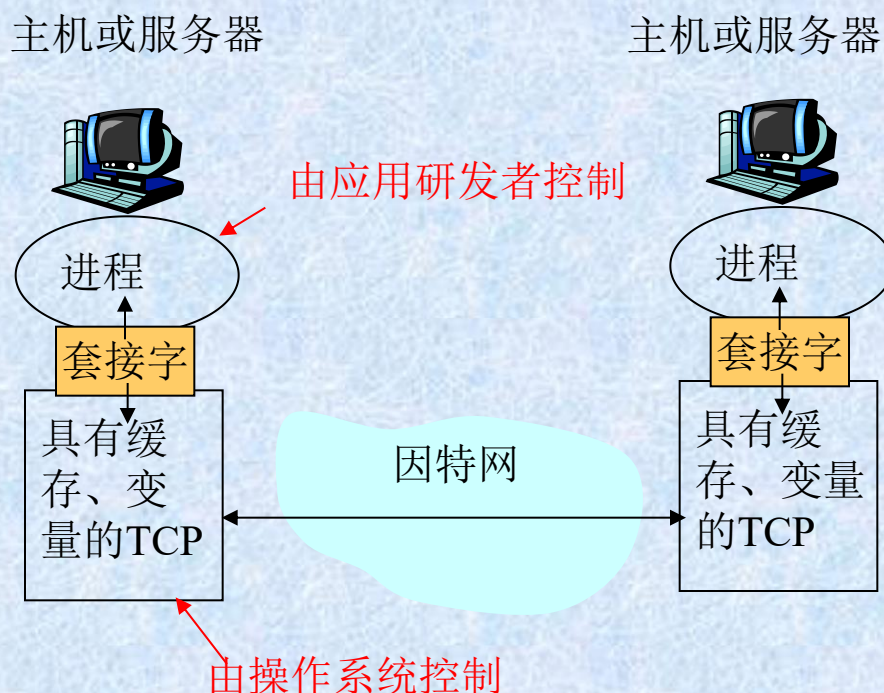
✓ **组成**：HTML、Web浏览器、Web服务器程序，以及一个应用层协议**HTTP(超文本传输协议)**等。

✓ **HTTP**定义了浏览器程序和Web服务器程序间传输的报文格式和序列。

✓ **其他协议**：电子邮件协议SMTP等等

2.1.4 应用程序所需要的服务

- ❑ 应用程序间通信：由**运输协议**跨越网络将发送进程的报文传输到接收进程的门户。
- ✓ **需要使用运输协议所提供的服务。**
- ✓ 运输协议有多种，提供的服务不同。



应用程序需要什么样的运输服务？

- ❑ 可靠的数据传输（无数据丢失）
 - ✓ 数据不能丢失的应用：如文件传输、金融事务等。
 - ✓ 能容忍数据丢失的应用：如多媒体应用。
- ❑ 带宽（数据传输率）
 - ✓ 带宽敏感的应用：需要特定的带宽才能正常工作。
如，因特网电话、其他多媒体应用。
 - ✓ 弹性应用：使用的带宽多或少影响不大。
如电子邮件、文件传输以及Web传输。
- ❑ 定时（数据传输的时间限制）
 - ✓ 交互式实时应用：对时间敏感，要求时延小。
如，因特网电话、视频会议以及多方游戏等。
 - ✓ 非实时应用：时延无限制，时延低更好。
- ❑ 安全性

典型应用的运输服务要求

应用程序	数据丢失	带宽	时间敏感
文件传输	不能丢失	弹性	不
电子邮件	不能丢失	弹性	不
Web 文档	不能丢失	弹性	不
实时音频/视频 (因特网电话/视频会议)	容忍丢失	音频: 5kbps-1Mbps 视频: 10kbps-5Mbps	是, 100 ms
存储音频/视频	容忍丢失	同上	是, 几秒
交互式游戏	容忍丢失	几kbps以上	是, 100 ms
即时讯息	不能丢失	弹性	是和不是

2.1.5 因特网运输协议提供的服务

两个运输层协议：

- 用户数据报协议UDP

- 传输控制协议TCP

- ✓ 每个协议为调用它们的应用程序提供不同的服务模型。

- ✓ 在创建一个新的因特网应用时，要选择其中一个。

1、TCP服务

- 两个方面:
- ✓ 面向连接的服务:
- ✓ 可靠的传输服务:

面向连接的服务

□ 划分三阶段

✓ 建立连接（握手过程）：

客户机程序和服务器程序之间互相交换控制信息，在两个进程的套接字之间建立一个TCP连接。

✓ 传输报文：

连接是全双工的，即连接双方的进程可以在此连接上同时进行报文收发。

✓ 拆除连接：

应用程序报文发送结束。

可靠的传输服务

- ✓ 通信进程可以**无差错、按适当顺序**交付发送的数据。
- ✓ 没有数据丢失和重复。

拥塞控制

当发送方和接收方之间的网络出现拥塞时，**会抑制发送进程速率。**

对整个网络有益。

拥塞控制还会试图限制每个TCP链接，达到**公平共享网络带宽**的目的。

未提供的服务

- ❑ TCP和UDP都没有提供任何加密机制。
- ❑ 安全套接字层(Secure Socket Layer, SSL)
 - 属于应用层协议，位于应用层协议与TCP之间；
 - 提供了加密、数据完整性、端点鉴别等功能，实现进程间安全通信。

未提供的服务

- ✓ **不确保最小传输速率：**发送进程受拥塞控制机制的制约；
- ✓ **不提供时延保证：**数据传输的时间不确定。

TCP协议能保证交付所有的数据，但并不保证这些数据传输的速率以及期待的传输时延。

2、UDP服务

仅提供**最小服务模式**的轻量级运输协议。

- ✓ **无连接**：两个进程通信前没有握手过程；
- ✓ **不可靠数据传输**：不保证报文能够被接收，或收到的报文是乱序到达。
- ✓ **没有拥塞控制机制**：发送进程可以任何速率发送数据
- ✓ **不提供时延保证**

因特网应用、应用协议与运输协议

应用	应用层协议	传输协议
电子邮件	SMTP	TCP
远程终端访问	Telnet	TCP
Web	HTTP	TCP
文件传输	FTP	TCP
远程文件服务器	NFS	UDP或TCP
流媒体	HTTP	TCP
因特网电话	SIP、RTP	UDP或TCP



2.2 Web应用和HTTP协议

- Web是World Wide Web的简称，中文称之为万维网，是用于发布、浏览、查询信息的网络信息服务系统。
- Web关键技术
 - 用户超文本标记语言（**HTML**）实现信息与信息之间的链接；
 - 用统一资源定位技术（**URL**）实现全球信息的精确定位；
 - 用超文本传输协议（**HTTP**）实现分布式信息共享。

2.2 Web应用和HTTP协议

产生于20世纪90年代初期。

- ✓ 改变了人们与工作环境内外的交流方式；
- ✓ 提升因特网地位；
- ✓ 生活和工作发生变化；
- ✓ 方便、快捷得到所需要的信息（**按需操作**）；
- ✓ 任何人可在Web上发布信息；
- ✓ 超链接和搜索引擎帮助人们浏览Web站点。

开拓、创新

- 伯纳斯·李：万维网的发明者，南安普顿大学与麻省理工学院教授。1990年12月25日，他和罗伯特·卡里奥在CERN（欧洲核子研究组织）一起成功通过Internet实现了HTTP代理与服务器的第一次通讯。
- 2017年，他因“发明万维网、第一个浏览器和使万维网得以扩展的基本协议和算法”获得2016年度的图灵奖。



蒂姆·伯纳斯·李

本节内容

2.2.1 HTTP概况

2.2.2 HTTP连接

2.2.3 HTTP报文格式

2.2.4 用户与服务器交互：Cookie

2.2.5 HTTP内容

2.2.6 Web缓存

2.2.7 条件GET方法

2.2.1 HTTP概况

□ HTTP（超文本传输协议）：

应用层协议，Web的核心。

□ 包括两部分：

✓ 客户机程序

✓ 服务器程序

分别运行在不同的端系统中，通过交换HTTP报文进行会话。

□ HTTP协议定义了报文的格式以及客户机和服务器交换报文的方式。

Web常用术语

□ **Web页(文档):** 由若干**对象**组成。

对象: **是文件**。通过一个**URL地址**来寻址。如HTML文件、JPEG图形文件、Java小程序等。

✓ Web页含有**一个基本的HTML文件及几个引用对象**。

例如，一个Web页包含HTML文本和5个JPEG图形文件（即有6个对象）。

✓ 在基本的HTML文件中，**每个对象可由URL来寻址**。

超文本标记语言

统一资源定位符

Web常用术语

✓ **URL: Uniform Resource Locator。统一资源定位符。**
标识万维网WWW上的各种文档，全网范围唯一。

✓ **URL地址组成:**

存放对象的**服务器主机名**和**对象的路径名**。

如

`http:// www.someSchool.edu / somedepartment / picture.gif`


主机名

路径名

Web常用术语

- ❑ **浏览器（客户机）**：是Web应用的**用户代理**，用于显示所请求的Web页，提供导航功能和配置属性。

实现了HTTP协议的客户机端。

- ❑ **Web服务器**：用于存贮Web对象（由URL寻址）。

实现HTTP协议的服务器端。

HTTP协议定义了Web客户机 (浏览器)如何向Web站点请求Web页，以及服务器如何将Web页传送给客户机。

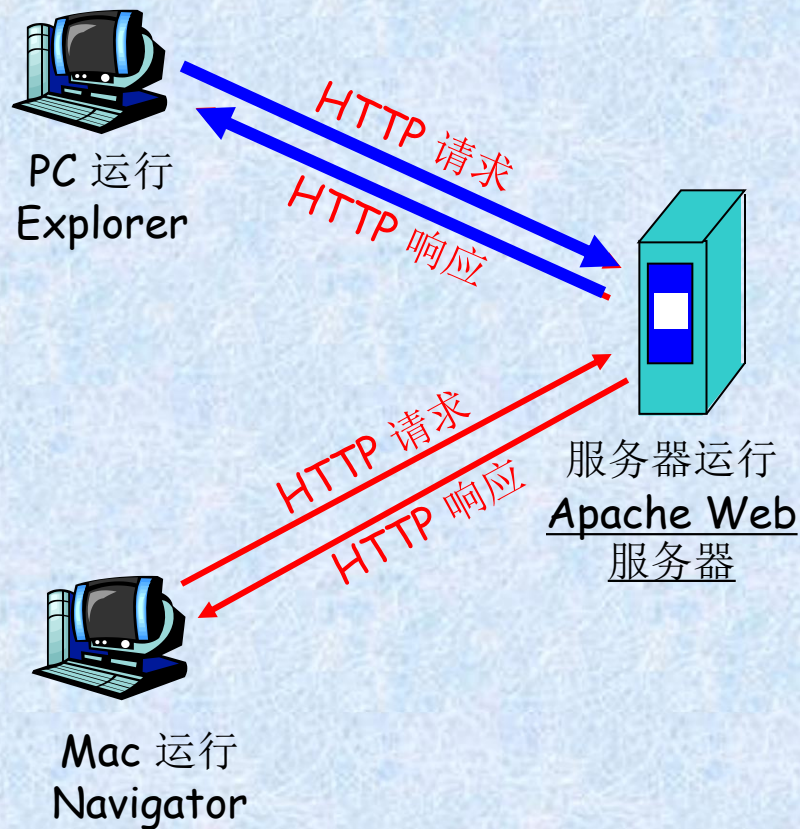
客户机和服务器之间交互过程

□ 客户机：

用户请求一个Web页
(如点击一个超链接)，浏览器向服务器发出对该页所含对象的“HTTP请求报文”。

□ 服务器：

接受请求，回发包含所请求对象的“HTTP响应报文”



说明

□ HTTP协议使用的**底层运输协议是TCP**。

□ **工作过程：**

创建TCP连接→交换报文→关闭TCP连接

- ✓ 客户机先与服务器**建立TCP连接**，然后，浏览器和服务进程通过**套接字访问TCP**：[如图2-3](#)
- ✓ **客户机**：从其套接字接口发送“HTTP请求报文”和接收“响应报文”；
- ✓ **服务器**：从其套接字口接收“HTTP请求报文”和发送“响应报文”。



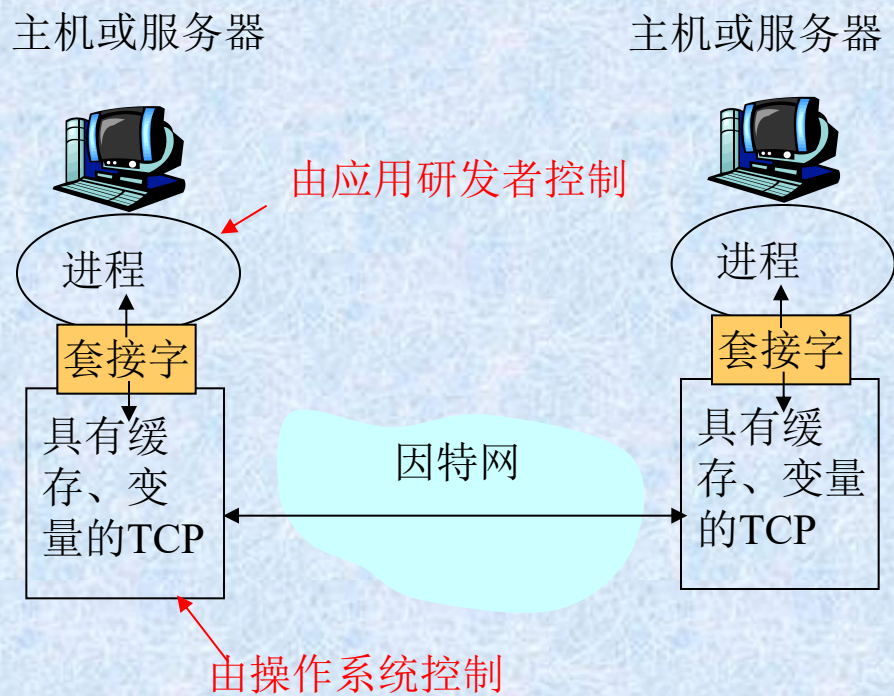


图2-3 应用进程、套接字和下面的运输层协议

说明

分层体系结构
的优点？

- TCP提供可靠的数据传输服务：客户机进程和服务进程发出的每个HTTP报文能完整地到达对方。
- HTTP是无状态协议：服务器不保存关于客户机的任何信息。
 - 如果用户多次请求同一个对象，服务器是否会多次响应？

Web使用客户机/服务器结构，Web服务器总是打开，有一个固定IP地址，为多个浏览器服务。

2.2.2 HTTP连接

非持续HTTP连接

- ✓ 每个TCP连接上只传送一个Web对象
- ✓ 只传送一个请求/响应对

持续HTTP连接

- ✓ 一个TCP连接上可以传送多个Web对象
- ✓ 传送多个请求/响应对

默认方式下使用持续连接

1、非持续连接

例，客户机向服务器请求传送一个Web页：

- ✓ 含有一个基本HTML文件和10个JPEG图形，11个对象位于同一个服务器上。
- ✓ HTML文件的URL为：

<http://www.someSchool.edu/someDepartment/home.index>



工作过程

TCP 连接

报文传输

1a. HTTP客户初始化一个与服务
器主机www.someSchool.edu中
HTTP服务器进程的TCP连接

1b. www.someSchool.edu服务器
主机中的HTTP服务器在80端
口监听来自HTTP客户的TCP
连接请求，收到连接请求，接
受，建立连接，通知客户。

2. HTTP客户发送一个HTTP请
求报文（包含URL）到TCP
连接套接字，报文指明客户
需要的Web对象—
someDepartment/home.index

3. HTTP服务器接收请求报文，产
生一个响应报文（包含被请
求对象），并发送到其TCP连
接套接字

time

4. HTTP服务器关闭TCP 连接

5. HTTP客户机接收包含HTML文件的响应报文，显示并解析HTML文件，发现10个引用的 jpeg对象

6. 对10个jpeg对象重复步骤1~5

time

说明：

- 每个TCP连接在服务器返回对象后关闭（非持续）。
- 每个TCP连接只传输一个请求报文和一个响应报文；

上例中，要建立11个TCP连接。

- 浏览器可同时打开多个连接：
 - ✓ 并行的TCP连接：并行数大于1。默认打开5~10个。
 - ✓ 串行的TCP连接：最大并行数为1。

请求一个HTML文件所需时间

即从客户机请求基本HTML文件开始，到用户收到整个文件为止所花时间。

□ 往返时延RTT(Round Trip Time):

一个小分组从客户机到服务器，再回到客户机所花时间。

包括传播时延、排队时延以及处理时延。

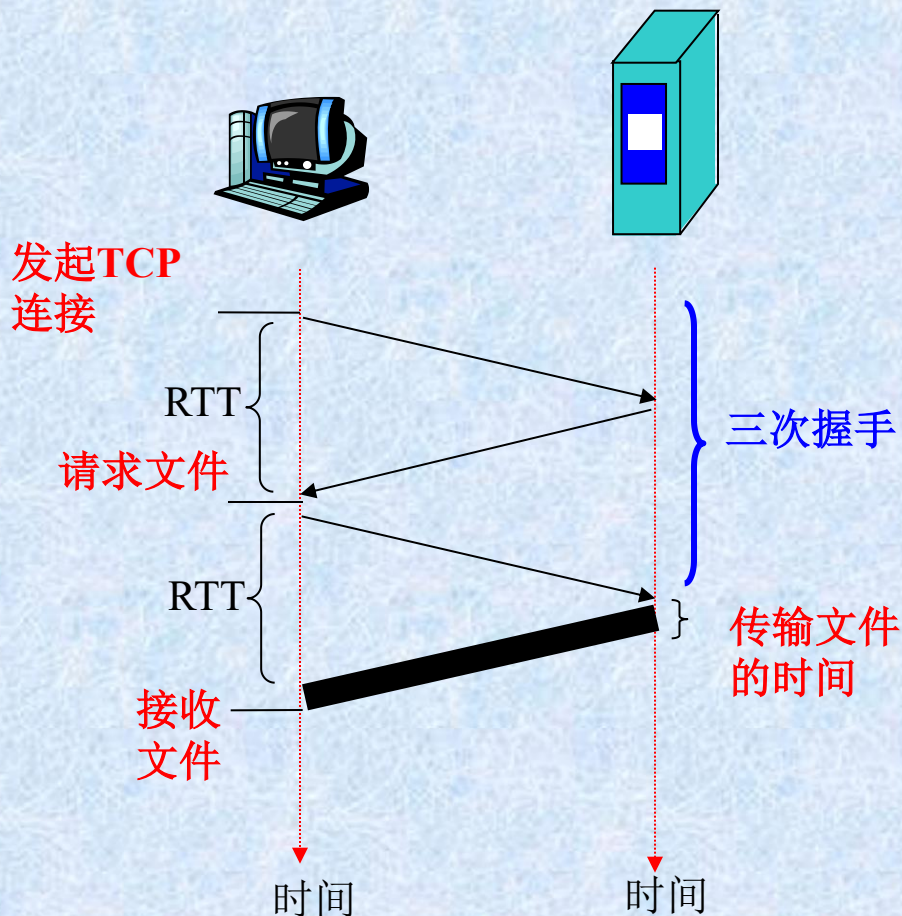
建立TCP连接 → 交换报文

□ TCP连接的“三次握手”过程

- ✓ 客户机发送一个TCP连接请求报文
- ✓ 服务器回送一个TCP确认响应报文
- ✓ 客户机向服务器发送一个包含“HTTP请求”与“TCP确认”的报文

□ 总响应时间：两个RTT时延加上服务器发送文件的时间

总计 = $2RTT$ + 文件传输时间



非持续连接缺点

- ❑ **服务器负担重：**每一个请求对象需建立和维护一个新的连接。
- ❑ **每一个对象的传输时延长：**包含两个RTT时延，一个用于TCP连接建立，一个用于请求和接收对象。



2、持续连接

- 服务器在发送响应后保持该TCP连接：
 - ✓ 相同客户机与服务器之间的后续请求和响应报文通过相同的连接进行传送。

如，一个Web页的所有对象可以通过一个持续TCP连接传送。
 - ✓ 或同一服务器上的多个Web页也可以通过一个持续TCP连接传送给同一个客户机。
- 连接经一定时间间隔(超时间隔)未被使用，服务器就关闭该连接。

持续连接两种方式

- 非流水线方式：客户机只能在前一个响应接收到之后才能发出新的请求。
- ✓ 客户机为每一个引用对象的请求和接收都使用一个RTT时延。
- ✓ 会浪费一些服务器资源：服务器在发送完一个对象，等待下一个请求时，会出现空闲状态。

持续连接两种方式

□ 流水线方式:

- ✓ 客户机可一个接一个**连续产生请求**（只要有引用就产生），即在前一个请求接收到响应之前可以产生新的请求。
- ✓ 服务器一个接一个**连续发送相应对象**。

□ 特点:

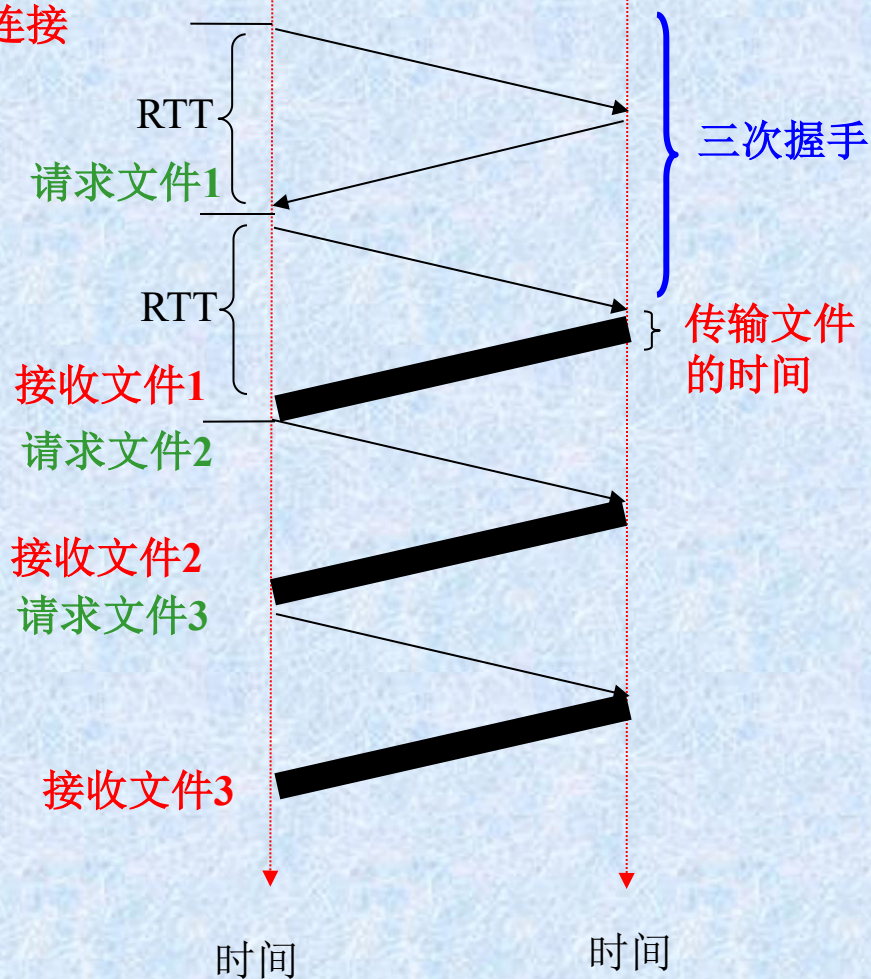
- ✓ 节省**RTT时延**，可能所有引用对象**只需一个RTT时延**。
- ✓ TCP连接空闲时间很短。

□ 默认方式：流水线方式的持续连接。

思考：P8（假定服务器IP地址已知， RTT_0 为主机与服务器之间的往返时延）



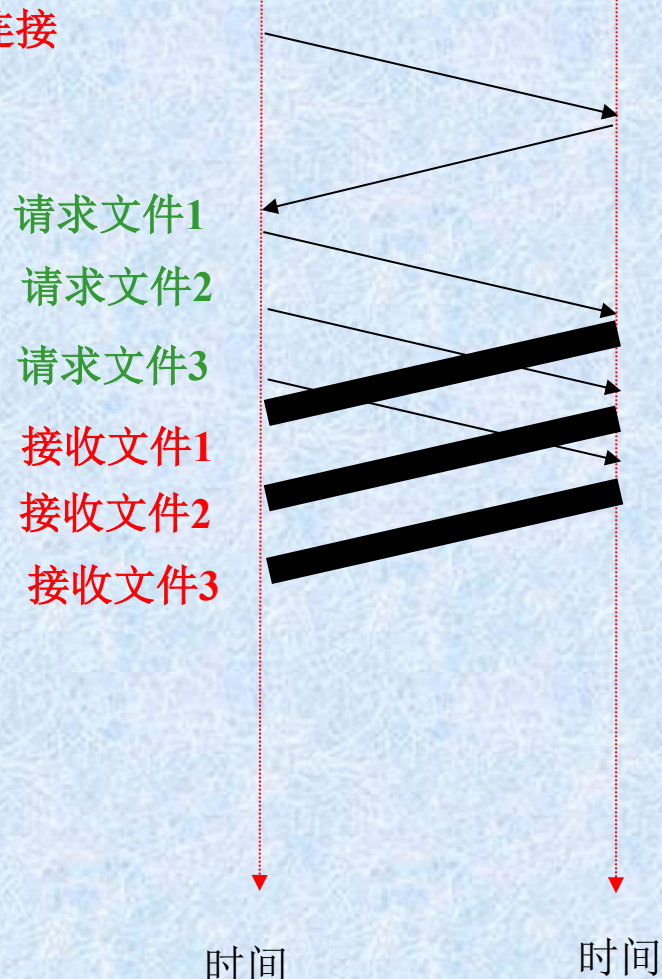
发起TCP
连接



非流水线方式



发起TCP
连接



流水线方式

1、常用的应用程序被指派的固定端口号称为周知端口。其中Web服务进程HTTP协议的周知端口和邮件服务进程SMTP协议所指派的周知端口分别为：

- ☐ **A 8080,25**
- ☐ **B 8080,21**
- ☒ **C 80,25**
- ☐ **D 80,21**

提交