1. 连接因特网的所有设备都被称为**主机(host)**或**端系统(end system)。**端系统通过**通信链路和分组交换机**链接到一起。

链路的传输速度以**比特/秒(bit/s，或bps)**度量。当一台端系统向另一台端系统发送数据时，发送端系统将数据分段，并为每段加上首部字节。由此形成的信息包成为**分组**。

分组交换机从它的一条入通信链路接收到达的分组，并从它的一条出通信链路转发该分组。两种最著名的分组交换机是**路由器和链路层交换机。**路由器通常用于网络核心中，链路层交换机常用于接入网中。

从发送端系统到接收端系统，一个分组所经历的一系列通信链路和分组交换机被称为通过该网络的路径。

端系统通过**因特网服务提供商(ISP)**接入因特网，包括如公司ISP、大学ISP等。每个ISP自身就是一个由多台分组交换机和多段通信链路组成的网络。为端系统提供接入的ISP是互联的。

端系统、分组交换机等因特网部件都要运行一系列协议，控制因特网中信息的接受和发送。TCP、IP是因特网中两个最为重要的协议。**IP协议**定义了在路由器和端系统之间发送和接收的**分组格式**。因特网的主要协议统称为TCP/IP。

1. 服务描述

涉及多个相互交换数据的端系统的应用被称为分布式引用程序，应用程序运行在端系统上。

与因特网相连的端系统提供了一个**套接字接口**，规定了运行在一个端系统上的程序请求因特网基础设施向运行在另一个端系统上的特定目的地程序交付数据的方式。因特网套接字接口是一套发送程序必须遵循的规则集合。

1. 网络核心

网络核心是指互联网中的主干网络，也称为互联网骨干网络。它由一些大型的网络服务提供商（如AT&T、Verizon、Level 3等）和国家级的互联网骨干节点（如中国的CERNET、美国的NSFNET等）组成，这些节点通过高速的光纤、电缆、卫星链路等方式相互连接，形成了一个庞大的、分布式的网络核心。

1. 协议

为了完成一项工作，需要两个或多个通信实体运行相同的协议。在因特网中，两个或多个**远程通信实体**的所有活动都受协议的制约。

计算机网络广泛地使用了协议，不同的协议用于完成不同的通信任务。

1. Web

Web（全称为World Wide Web）是一种基于互联网的信息交流和共享系统，是互联网最重要的应用之一。它由一个分布式的信息资源组成，这些资源通过超链接（Hyperlink）相互链接，形成一个庞大的、跨越全球的信息网络。  
 Web最初由英国计算机科学家蒂姆·伯纳斯-李（Tim Berners-Lee）在1989年发明，并于1991年正式公开发布。Web的核心技术是超文本标记语言（HTML）、统一资源定位符（URL）和超文本传输协议（HTTP），这些技术为Web提供了一种基于文本、图像、音频和视频等多媒体内容的信息交流和共享方式。  
 Web的优点在于它能够方便、快捷地获取和共享信息，使得人们能够更加高效地进行学习、工作和娱乐。同时，Web也成为了商业活动和社交媒体等应用的主要平台之一，对于经济和文化的发展产生了深远的影响。

Web服务器：

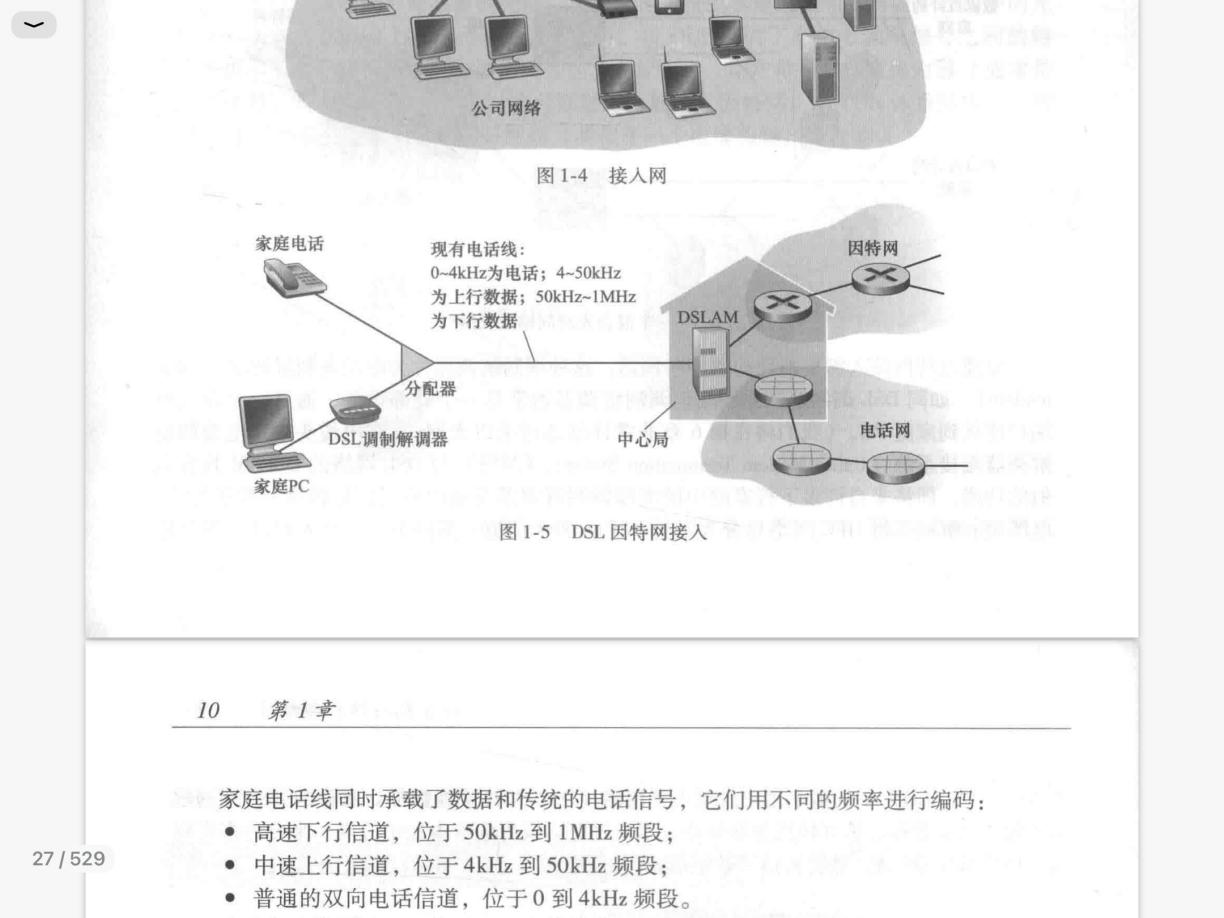
Web服务器是指在互联网上提供Web服务的计算机软件。它负责接收来自客户端浏览器的HTTP请求，并将相应的Web页面或其他资源发送回客户端。Web服务器可以是独立的计算机或者运行在其他计算机上的进程或线程。

1. 接入网

接入网是指将端系统物理连接到其**边缘路由器**的网络，边缘路由器是端系统到任何其他远程端系统的路径上的第一台路由器。

宽带住宅接入流行的有**数字用户线(DSL)和电缆**两种。使用DSL时，用户的本地电话公司即它的ISP。

DSL调制解调器将数字数据转换为高频音，通过电话线传输给数字用户线接入复用器(DSLAM)，DSLAM将模拟信号转换回数字形式。

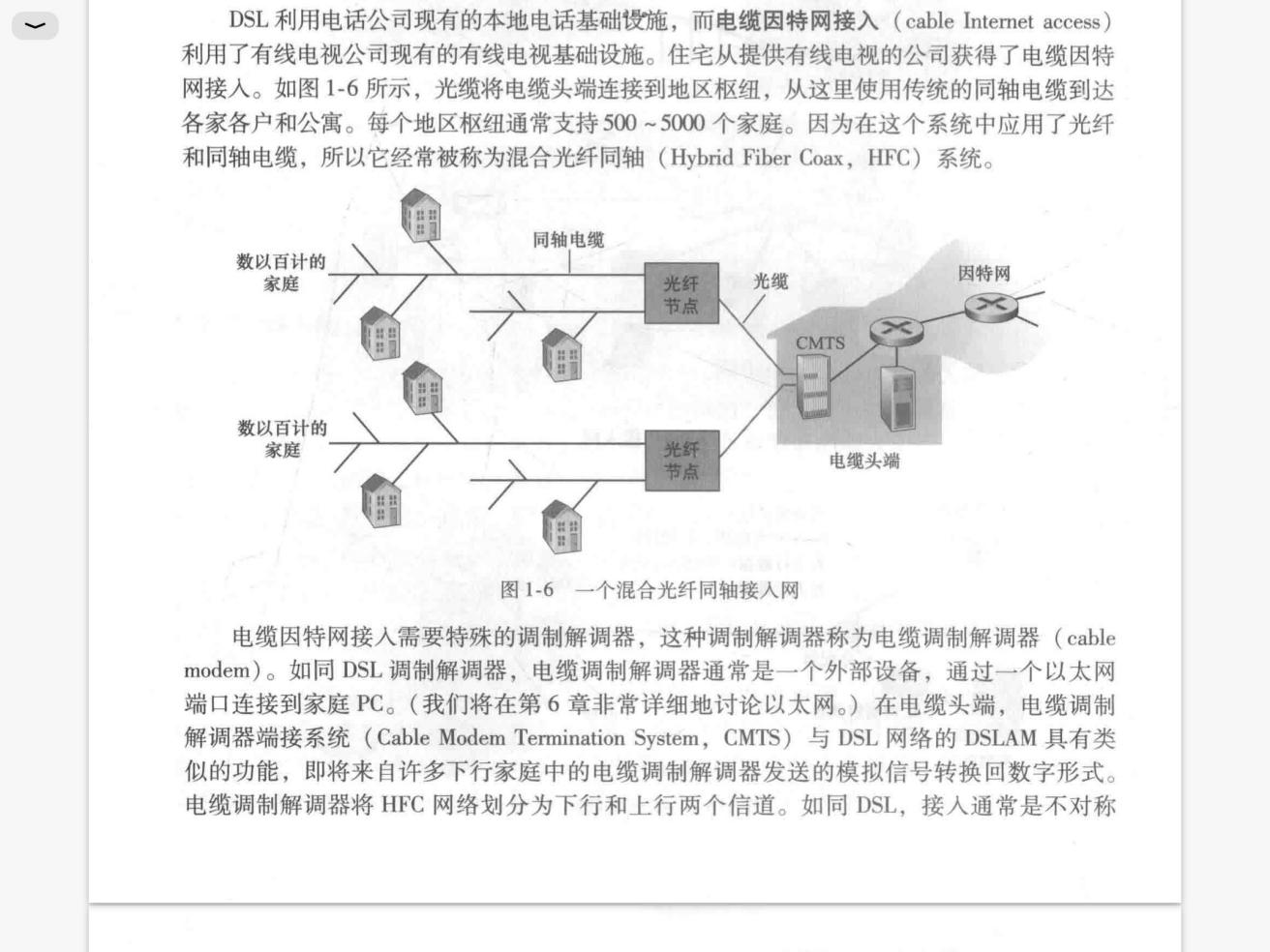


高速下行信道，位于50kHz到1MHz频段；中速上行信道位于4kHz到50kHz频段；普通的双向电话信道位于0到4kHz频段。

在用户侧，一个分配器将到达家庭的数据信号与电话信号分开，并将数据信号转发给DSL调制解调器。电话公司侧，DSLAM将数据和电话信号分开，并将数据送往因特网。DSLAM通常与数百上千个家庭相连。

DSL标准定义了多个传输速率，包括12Mbps下行和1.8Mpbs上行传输速率。上行速率和下行速率是不同的，所以这种接入是不对称的，实际速率小于定义的速率。

**电缆因特网接入**利用有线电视公司现有的有线电视基础设施。



从家庭开始，电缆通过光纤节点再通过光缆到达地区枢纽。电缆调制解调器同DSL解调器，接在用户端。在电缆头端，电缆调制解调器端接系统(CMTS)与DSL网络的DSLAM类似。

如果有多个家庭同时用电缆，则传输速率将降低。

光纤到户(FTTH)传输能达到更高速率，是从本地中心局直接到家庭提供了一条光线路径。

卫星链路也是接入网技术，传输速率慢。

1. 以太网

以太网是最流行的局域网(LAN)接入技术，局域网将端系统连接到边缘路由器。多个以太网用户使用双绞铜线与一台以太网交换机相连，以太网交换机再与更大的因特网相连。

局域网是指连接在一个有限地理范围内的计算机和设备组成的网络，通常由一组交换机或路由器连接起来，而不需要经过互联网。**局域网是接入网的一部分**，需要通过其他部分才能接入互联网。

1. 网络核心

网络核心是因特网端系统的分组交换机和链路构成的网状网络。

(1 存储转发传输。多数分组交换机在链路的输入端使用**存储转发传输机制**，是指在交换机能够开始向输出链路传输该分组的第一个比特之前，必须接收到整个分组。



L是数据量(KB、MB)，R是传输速率，N是路径数量(N-1是路由器数量)。

(2 排队时延与分组丢失。每台分组交换机有多条链路与之相连。对于每条链路，该分组交换机都有一个输出缓存(也称输出队列)，用于存储路由器准备发往那条链路的分组。

到达的分组传输时如果该链路被其他分组占据，则必须排队。分组除了存储转发时延以外还有排队时延。

缓存空间有限，一个到达的分组可能发现该缓存被其他分组充满，则会出现分组丢失(丢包)，到达的分组或已经排队的分组之一将被丢弃。

(3 转发表和路由选择协议。路由器决定它应当向哪条链路转发分组的方式取决于转发表和路由选择协议。

在因特网中，每个端系统都有一个称为IP地址的地址。源主机发送分组时，该分组的首部包含目的地的IP地址。每台路由器都有一个**转发表，用于将目的地址映射为输出链路。**分组到达路由器时，路由器检查该分组的目的地地址并用这个地址搜索转发表，发现适当的出链路，将该分组导向该出链路。

**路由选择协议**用于自动地设置**转发表**

1. 电路交换

通过网络链路和交换机进行数据转移有两种基本方法：**电路交换和分组交换**。分组交换已提到过。

**电路交换**会在端系统间通信会话期间，**预留** 端系统间沿路径通信所需的资源(缓存、链路传输速率)。

链路中的电路是通过**频分复用(FDM)或时分复用(TDM)**来实现的。

（1 FDM.链路的频谱由跨越链路创建的所有连接共享。在连接期间为每条连接专用一个频段。在电话网络中，这个频段的宽度为4kHz，该频段的宽度被称为带宽。

(2 TDM.时间被划分为固定期间的帧，每帧又被划为固定数量的时隙，当网络跨越一条链路创建一条连接时，网络在每个帧中为该连接指定一个时隙。这些时隙专门由该连接单独使用，一个时隙可用于传输该连接的数据。

1. 网络的网络

ISP必须彼此互联，通过创建网络的网络可以做到。ISP有市级、省级、国家级、世界级等，他们彼此相连。第一层ISP最高。

接入ISP被认为是客户，客户接入的高级的ISP被认为是提供商。客户需要向提供商付费。

存在点(PoP)存在于ISP等级结构的除底层外的所有层次。一个PoP是提供商网络中的一台或多台路由器群组，客户ISP能够与提供商ISP连接。

除了第一层的ISP以外，任何ISP可以选择**多宿**，即可以与两个或多个提供商ISP连接。

客户ISP为了减少支付给提供商的费用，位于相同等级结构层次的邻近一对ISP能够**对等。**即将它们的网络连到一起，使它们之间的流量无需经过上游的中间ISP传输。

第三方公司能够提供一个**因特网交换点(IXP)**，多个ISP能够在这里一起对等。

接入ISP、区域ISP、第一层ISP、PoP、多宿、对等、IXP共同组成了**网络结构4**.

现今的因特网是**网络结构5**，它是在网络结构4的顶部增加了**内容提供商网络**。内容提供商网络与较低层ISP对等并与同层ISP连接，减少了支付费用并且控制服务内容。

1. 分组交换网中的时延

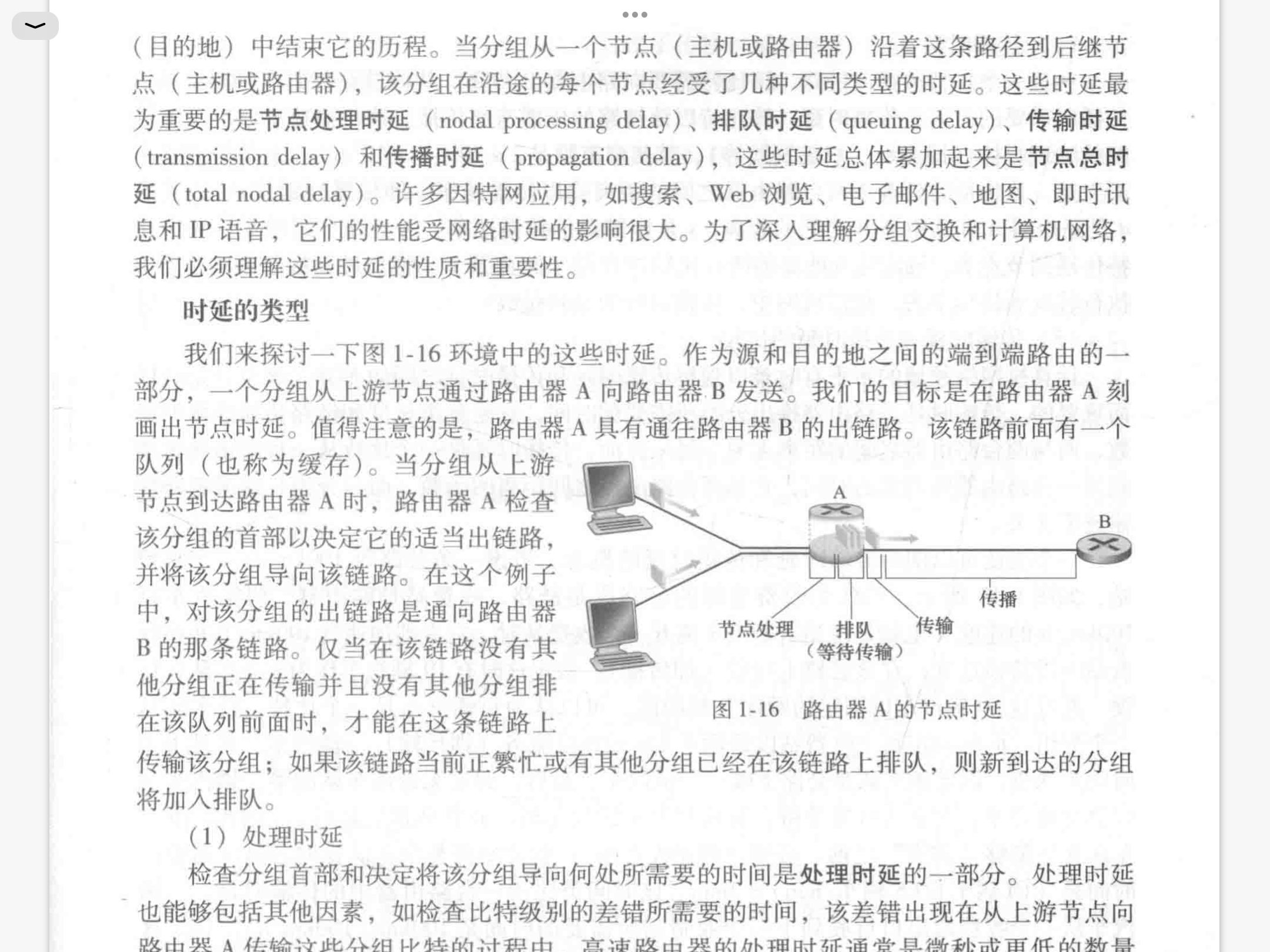
分组从一台主机出发，经过一系列路由器传输，到达另一台主机，在沿途中经历了几种不同类型的时延，重要的有**节点处理时延、排队时延、传输时延、传播时延**，这些时延累加起来是**节点总时延**。设路由器A向路由器B发送信息。

(1 **处理时延**。检查分组首部和决定将该分组导向何处所需要的时间是**处理时延的一部分**，处理时延还包括检查比特级别的差错等时间。在节点处理之后，路由器将该分组引向通往路由器B链路之前的队列。

(2 **排队时延。**若队列为空，排队时延为0.

(3 **传输时延。**传输时延是将所有分组的比特**推向链路**所需要的时间。设该分组的长度是L比特，R bps(即b/s)表示链路的传输速率。传输延时是L/R。

(4 **传播时延。** 一个比特被推向链路后，该比特需要向路由器B传播。从该链路的**起点到路由器B**所需要的时间是传播时延，该比特以该链路的传播速率传播，传播速率取决于物理媒体(光纤、双绞铜线等)。**传播时延=两台路由器之间的距离/传播速率**。



1. 排队时延与丢包

当表征排队时延时，常用平均排队时延、方差等统计量来度量。分组到达队列的速率(每秒到达队列的分组数)为a pkt/s，假定所有分组都是由L比特组成，则La bps是**比特到达队列的速率**R是传输速率，La/R 被称为**流量强度**，大于1时队列将无限增加。

当到达的分组发现队列满时，路由器将丢弃这个分组，这个分组不会到达目的地。

1. 端到端时延

是各路由器与主机时延之和。

1. 吞吐量

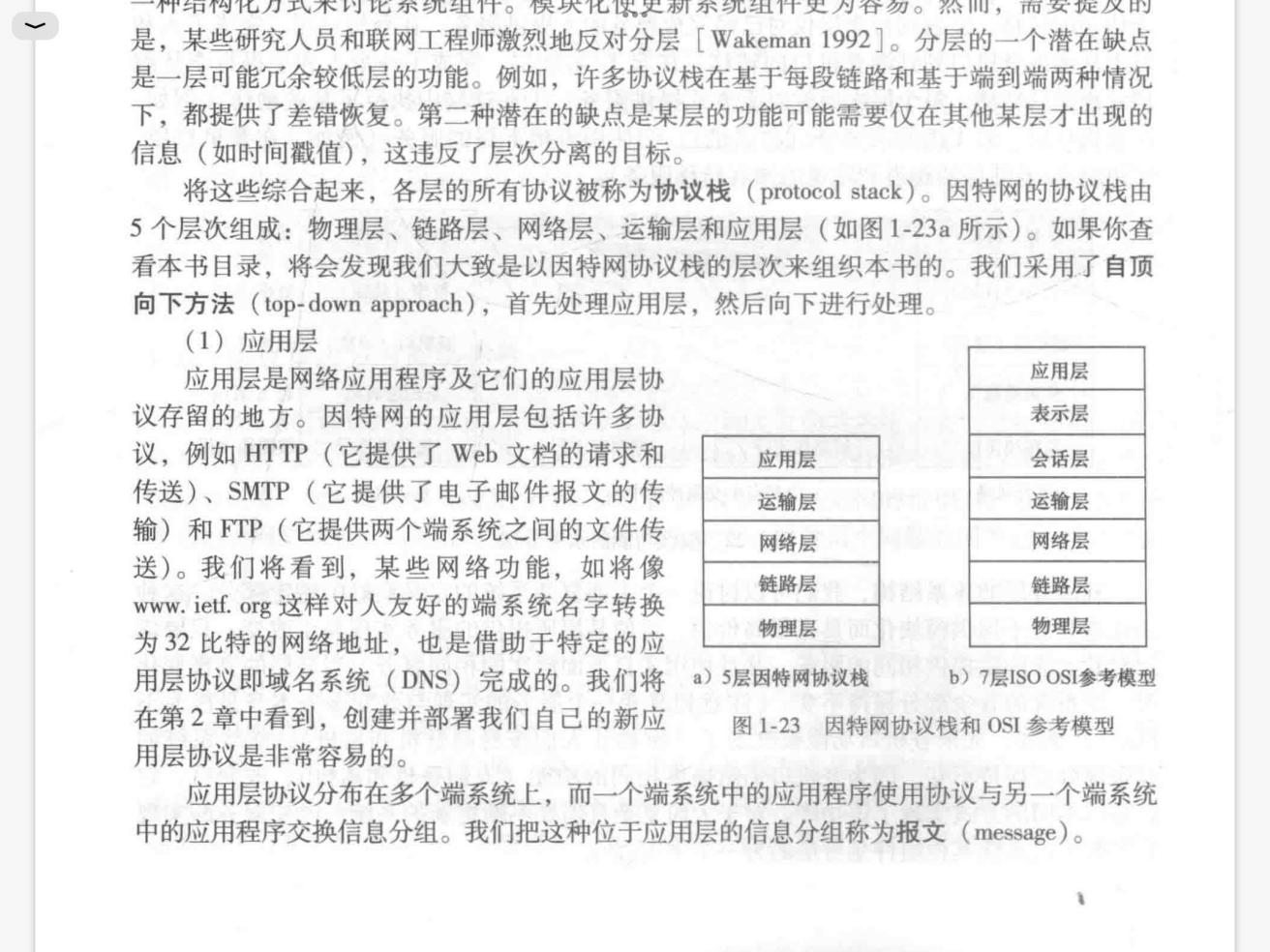
**瞬时吞吐量**是主机B接收到该文件的速率(bps为单位)。若该文件由F比特组成，主机B接收到所有F用时T秒，则文件的**平均吞吐量**是F/T bps。

设服务器与路由器之间的链路速率为R1，路由器与客户之间的链路速率为R2，则吞吐量为min(R1, R2)，它是**瓶颈链路**的传输速率。

1. 协议分层

网络设计者以**分层**的方式**组织协议**以及**实现这些协议**的网络硬件和软件。

各层的所有协议被称为协议栈，因特网的协议栈由5个层次组成：物理层、链路层、网络层、运输层、应用层。



(1 应用层。 应用层是网络应用程序及它们的应用层协议存留的地方，包括HTTP、SMTP、FTP等。应用层协议分布在**多个端系统**上，一个端系统中的应用程序使用协议与另一个端系统中的应用程序交换信息分组，这种位于应用层的信息分组称为**报文**。

(2 运输层。运输层在引用程序端点之间**传送应用层报文**。在因特网中 有TCP和UDP协议，运输层的分组称为**报文段**。

(3 网络层。 网络层的分组被称为**数据报**，网络层将数据报从一台主机移动到另一台主机。网络层包括著名的**网际协议IP**，定义了在数据报中的各个字段以及端系统和路由器如何作用于这些字段。网络层**也包括决定路由的路由选择协议**。网络层**包含网络协议和一些路由选择协议**，通常简单地称为**IP层**。

(4 链路层。网络层通过源和目的地之间的一系列**路由器路由数据报**。为了将分组从一个节点(路由器或主机)移动到路径上的下一个节点，网络层**必须依靠该链路层的服务**。在每个节点，网络层将数据报**下传给链路层**，链路层将数据报传递给下一个节点，在该节点将数据报上传给网络层。

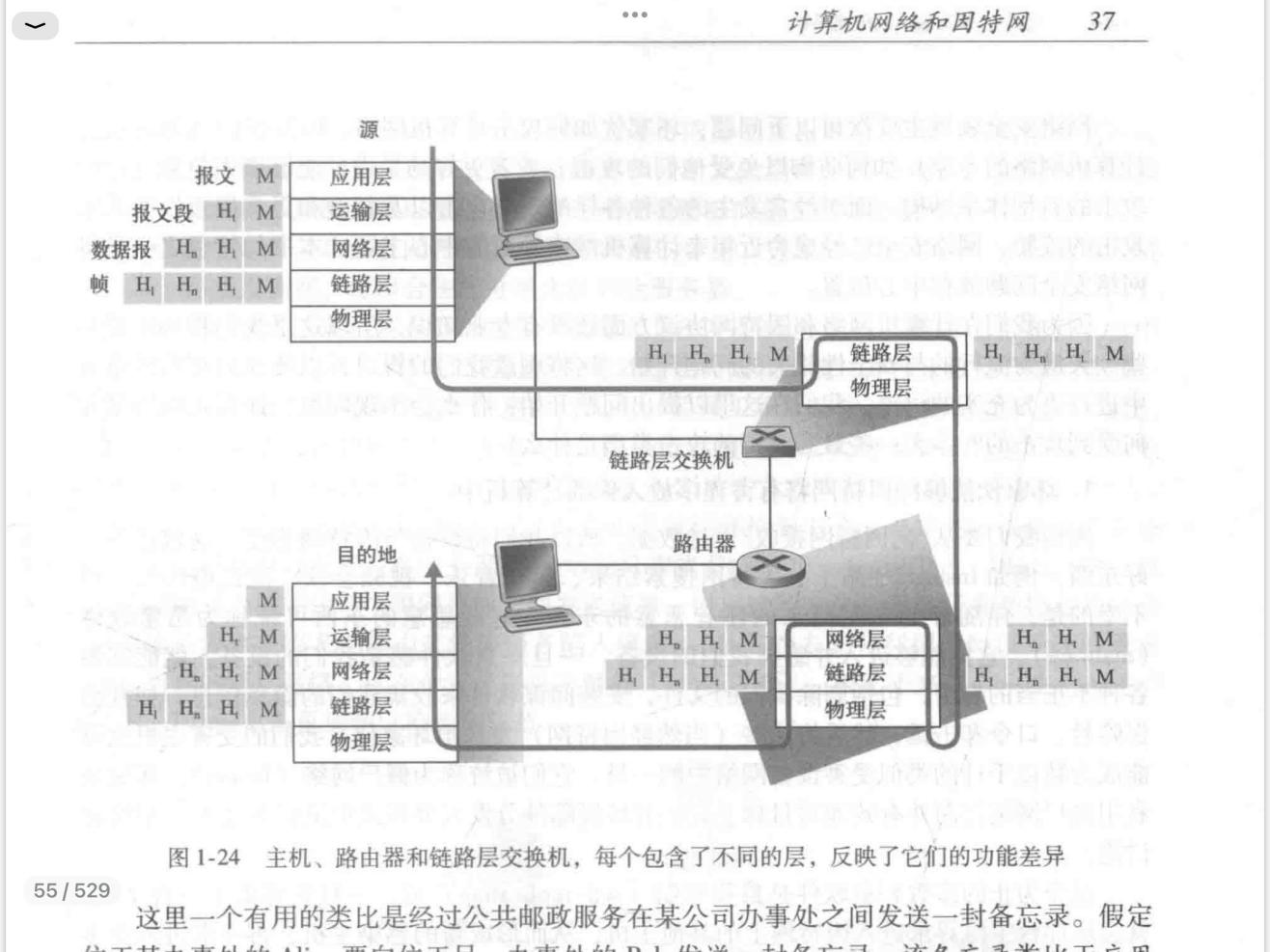
链路层的分组称为**帧**。不同的节点可能会有**不同的链路层协议**来服务，如以太网、WiFi等。

(5 物理层。物理层是将该帧中的**一个个比特**从一个节点移动到下一个节点。这层的**协议与链路和链路的实际传输媒体相关**。

1. 封装

链路层交换机实现了第一层和第二层(链路层、物理层)，路由器实现了第一到第三层

在应用层、运输层、网络层、链路层，源应用层的报文会一层层加上**附加信息**。一个分组具有两种类型的字段：**首部字段和有效载荷字段**，有效载荷字段是来自上一层的分组，首部字段是该层的附加信息。



1. 网络攻击

恶意软件利用**僵尸网络**攻击。病毒是一种**需要某种形式的用户交互**来感染用户设备的恶意软件，蠕虫是一种**无需任何明显用户交互**就能进入设备的恶意软件。

1. 拒绝服务攻击(DoS)

DoS能够使网络、主机或其他基础设施部分**不能由合法用户使用**。Web服务器、电子邮件服务器都能成为DoS攻击的目标。

**分布式DoS(DDoS)**，攻击者控制**多个源**并让每个源向目标猛烈发送流量。相比于来自单一主机的DoS攻击，DDoS**更难以检测和防范**。如果某服务器的接入速率为R bps，R足够大时单一攻击源可能**无法产生足够大的流量**来伤害该服务器，而且可能会被上游路由器阻挡。

1. 嗅探分组

记录**每个流经的分组副本**的被动接收机被称为**分组嗅探器**。

1. IP哄骗

将具有虚假源地址的分组注入因特网的能力被称为IP哄骗

1. 应用层协议原理

**网络核心设备**(路由器、链路层交换机)并不在应用层上起作用，仅在**较低层**(特别是网络层及下面层次)起作用

研发新应用程序时，无需写在网络核心设备上运行的软件。

**应用程序体系结构**规定了如何在各种端上组织该应用程序。

在客户-服务器体系结构中，有一个总是打开的主体称为**服务器**，服务于客服主机的请求，如Web应用程序服务于客户的浏览器请求。在这种体系结构中客户之间不直接通信。

在**P2P体系结构**中，主机与主机之间**直接通信**，主机对被称为**对等方**。

1. 进程通信

进行通信的实际上是进程而不是程序。在**不同端系统**上的进程，通过跨越计算机网络**交换报文**而相互通信。

对每对通信进程，两个进程分别称为**客户和服务器**。发起通信的进程被称为客户，等待联系的进程是服务器。在P2P中，一个进程既能是客户又能是服务器。

进程通过一个称为**套接字的软件接口**向网络发送报文和从网络接收报文。**套接字**是同一台主机内**应用层与运输层之间**的接口。套接字也称为应用程序和网络之间的**应用程序编程接口(API)**。应用程序开发者对于运输层的控制仅限于：选择运输层协议、设定几个运输层参数

1. 进程寻址

为了向目的地主机发送分组，接收进程需要有一个地址。为了标识该接收进程，需要定义两种信息：主机地址、目的主机中接收进程的标识符。

主机由IP地址标识。目的地端口号标识网络应用即 进程的标识符由端口号表示。已经给流行的应用分配了特定的端口号，如Web服务器用端口号80表示，邮件服务器进程用25表示。

1. 可供应用程序使用的运输服务

运输协议能够为调用它的应用程序提供服务，大致有可靠数据传输、吞吐量、定时、安全性四个方面

(1可靠数据传输

如果做一些工作确保应用程序的一段发送的数据完成正确、完全交付给另一端，则该协议提供了**可靠数据传输**。

(2 吞吐量

**可用吞吐量**是**发送进程**能够向接收进程交付比特的速率。具有吞吐量要求的应用程序被称为**带宽敏感的应用**。**弹性应用**能够根据当时可用的带宽或多或少地利用可供使用的吞吐量。

(3 定时

发送方注入进套接字的每个比特到达接收方套接字不迟于...毫秒。

(4 安全性

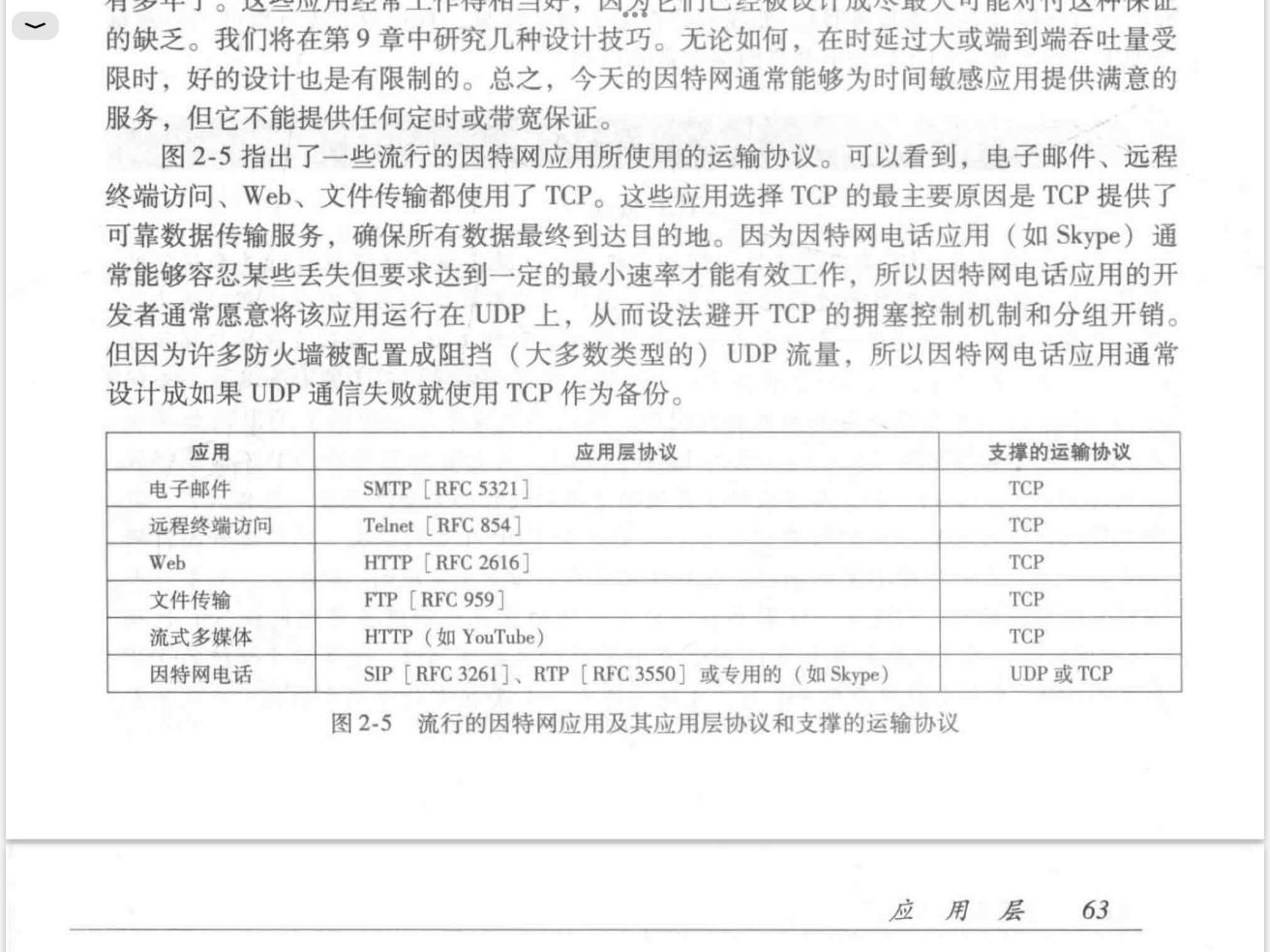
运输协议能够**加密**由发送进程传输的所有数据。在**接收主机**中，运输层协议能够在将数据交付给接收进程之前**解密**这些数据。

1. TCP服务

TCP服务包括面向连接服务和可靠数据传输服务，调用TCP地应用程序能够获得这两种服务。

面向连接的服务：在应用层数据报文开始流动之前，TCP让客户和服务器**互相交换**运输层控制信息(**握手**)。握手后一个**TCP连接**就在两个进程的套接字之间建立了。

UDP服务仅提供最小服务，无握手过程等服务



1. TCP安全

TCP和UDP都没有任何加密机制。TCP的加强方式被称为**安全套接字层(SSL)。**用SSL加强后的TCP能够提供关键的进程到进程的安全性服务。SSL本身不是运输协议。

1. 应用层协议

应用层协议定义了运行在不同端系统上的应用程序进程**如何相互传递报文**，特别是应用层协议定义了：

\*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这些字段是如何描述的

\*字段的语义，即这些字段中的信息的含义

\*确定一个进程何时以及如何发送报文，对报文响应的规则

有些应用层协议是由RFC文档定义的，它们位于公共域中，如HTTP协议作为一个RFC可供使用。还有别的应用层协议是专用的。

Web应用有很多组成部分，包括**文档格式的标准(即HTML)、Web浏览器、Web服务器、应用层协议**。HTTP定义了在浏览器和Web服务器之间传输的报文格式和序列。HTTP只是Web应用的一个部分

1. HTTP

Web的应用层协议是**超文本传输协议(**HTTP)，是Web的核心。HTTP由**两个程序**实现：一个客户程序和一个服务器程序。客户程序和服务器程序运行在不同的端系统中，通过**交换HTTP报文**进行会话。HTTP定义了这些**报文的结构**以及客户和服务器**进行报文交换的方式**。

Web页面由**对象**组成。HTML文件、JPEG图形等都是对象。一个对象是一个**文件**，可通过一个**URL地址寻址**，每个URL地址包括两部分：**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中，www.someSchool.edu是主机名，/someDepartment/picture.gif是路径名。**Web浏览器**实现了HTTP的客户端，所以在Web环境中经常交替使用浏览器和客户这两个术语。

Web服务器用于存储Web对象，每个对象由URL寻址。

HTTP使用TCP作为运输协议。

HTTP服务器不保存关于客户的任何信息，是一个**无状态协议**。

1. 非持续连接和持续连接

每个请求/相应对经一个**单独的TCP**连接发送，该应用程序被称为使用**非持续连接**。所有的请求及响应经过**相同的TCP**连接发送，该应用程序被称为使用**持续连接**。HTTP既能使用非持续连接，也能使用持续连接，默认为持续连接。

(1 使用**非持续连接**的HTTP

每个TCP连接在服务器发送一个对象后关闭，**并不为其他的对象而持续下来**。用户能够配置现代浏览器来控制连接的并行度，即同时打开**多个并行**的TCP连接。

**往返时间(RTT)**:一个短分组从客户到服务器然后**再返回客户**所花费的时间，包括分组传播时延、分组在中间路由器和交换机上的排队时延、分组处理时延。**总的响应时间**是**两个RTT**加服务器**传输**HTML文件的时间。一个RTT用于**创建TCP**，另一个RTT用于**请求和接收**一个对象。

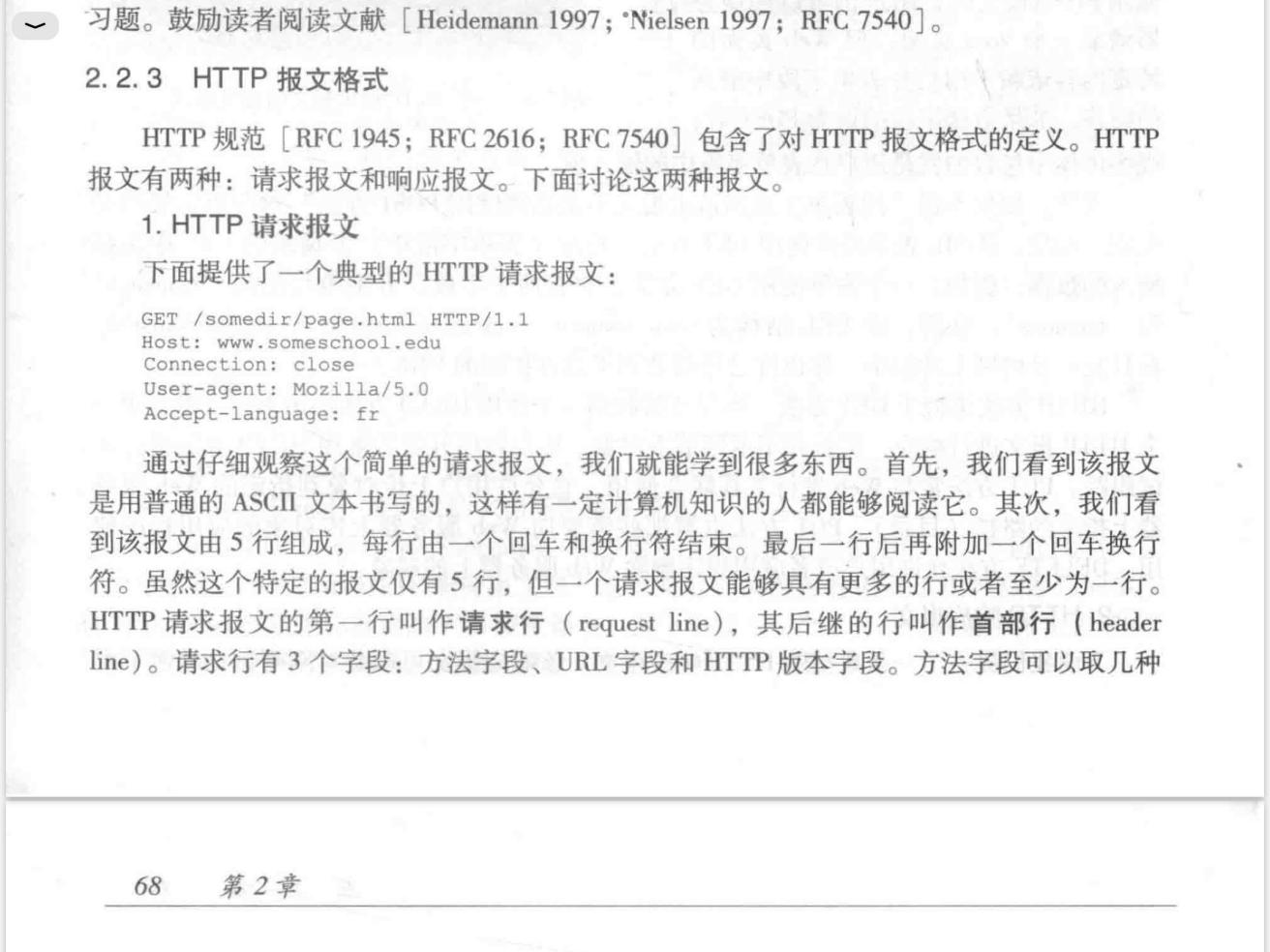
(2采用持续链接的HTTP

服务器在发送响应后**保持该TCP连接打开**，后续的请求和报文能够通过**相同的连接**传送。如果一条连接经过一定时间间隔未被使用，HTTP服务器会关闭该连接。

1. HTTP报文格式

HTTP报文有两种：请求报文和响应报文

(1 HTTP请求报文



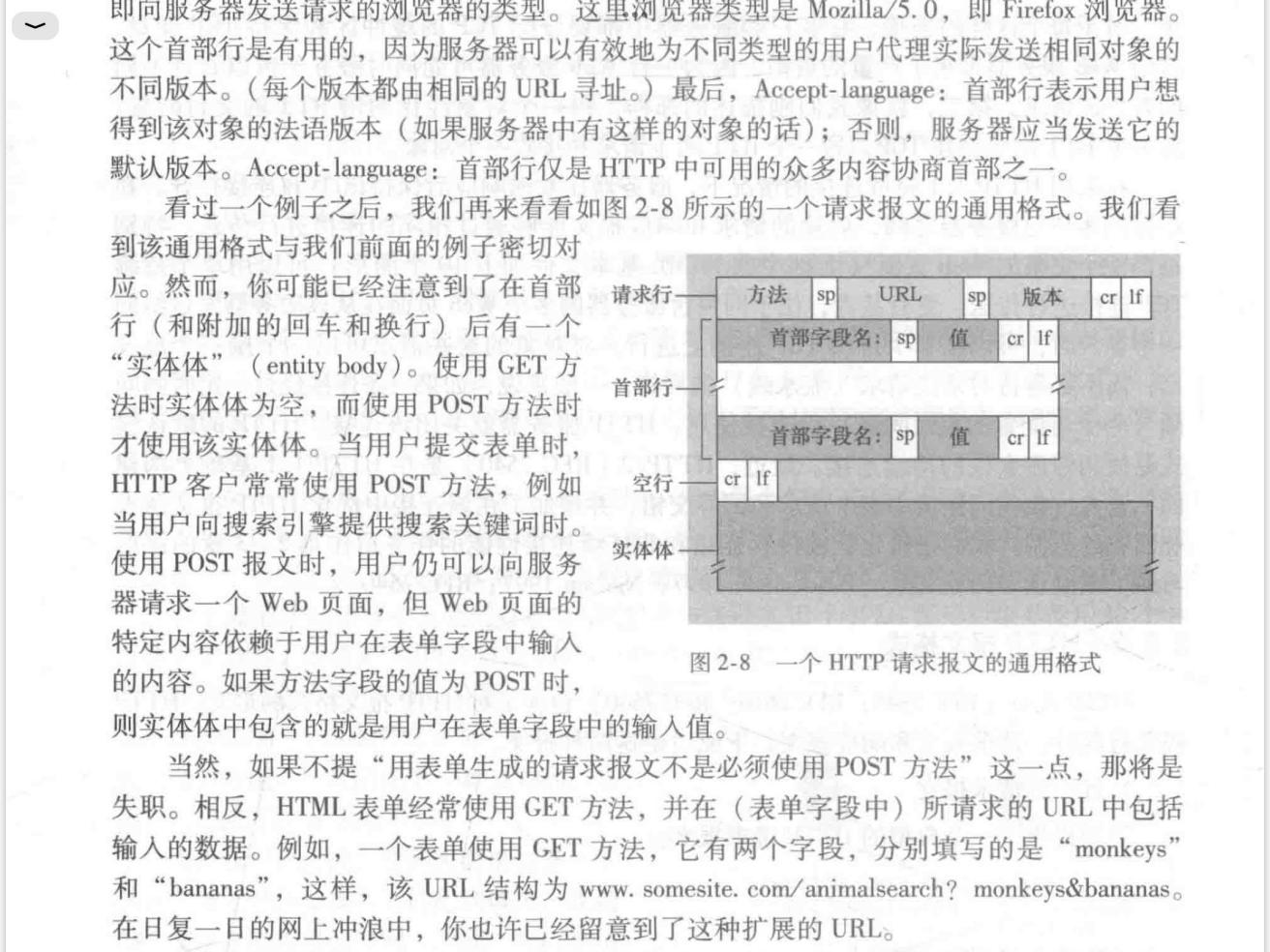
在最后一行有一个回车换行符，为空行。HTTP请求的**第一行**叫做**请求行**，后继的所有行叫作**部首行**。请求行有3个字段：方法字段、URL字段、HTTP版本字段。字段通过空格分隔。**方法字段**包括GET、POST、HEAD、PUT、DELETE几个不同的值，绝大部分HTTP使用GET方法。当浏览器**请求一个对象**时，使用GET方法。**URL字段**带有请求对象的标识。**版本字段**是HTTP/1.1版本。

首部行**Host**: [www.someschool.edu](http://www.someschool.edu) 指明了对象所在的主机，该首部行是Web代理高速缓存所要求的。

首部行**Connection：close** 表示服务器在发送完被请求的对象后就关闭这条连接。

**User-agent**: 指明**用户代理**，即向服务器发送请求的**浏览器的类型**。服务器可以为不同类型的用户代理实际发送相同对象的不同版本。

Accept-language: 用户想得到该对象的**语言版本**，舍去该首部行将发送默认版本。



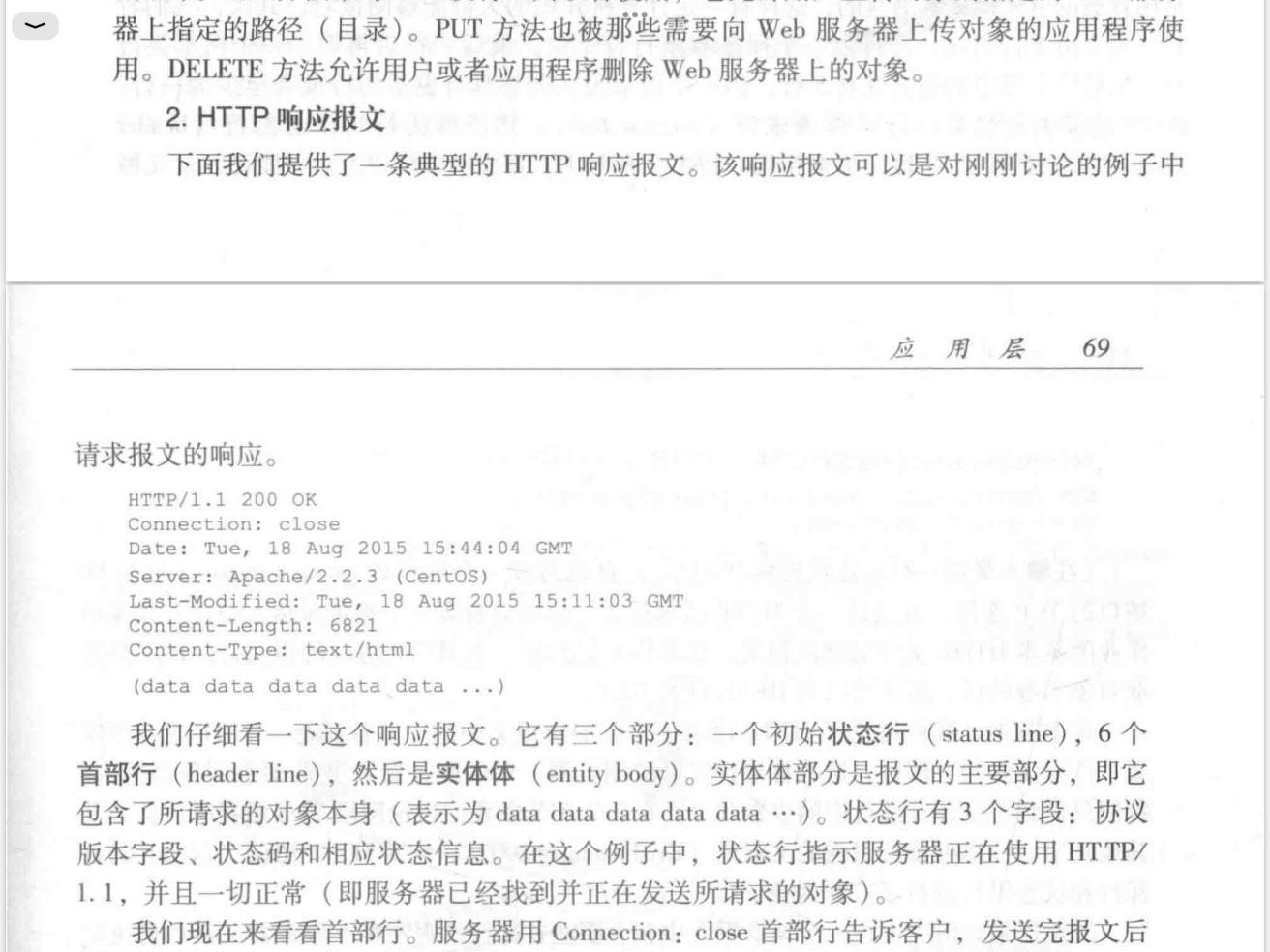
在空行后有**实体体**，使用GET方法时实体体为空，使用**POST**时才使用该实体体。当用户提交表单时常用POST方法。方法字段的值为POST时，实体体中包含的是用户在**表单字段中的输入值**。

用表单生成请求报文并不必须使用POST方法，HTML表单经常使用GET方法。

**HEAD方法**类似于GET方法，当服务器收到一个使用HEAD方法的请求时，将会用一个HTTP报文进行响应，但**不返回请求对象**，HEAD常用于调试跟踪。PUT方法允许用户**上传对象**到指定的Web服务器上指定的路径。DELETE允许用户或应用程序**删除**Web服务器上的对象。

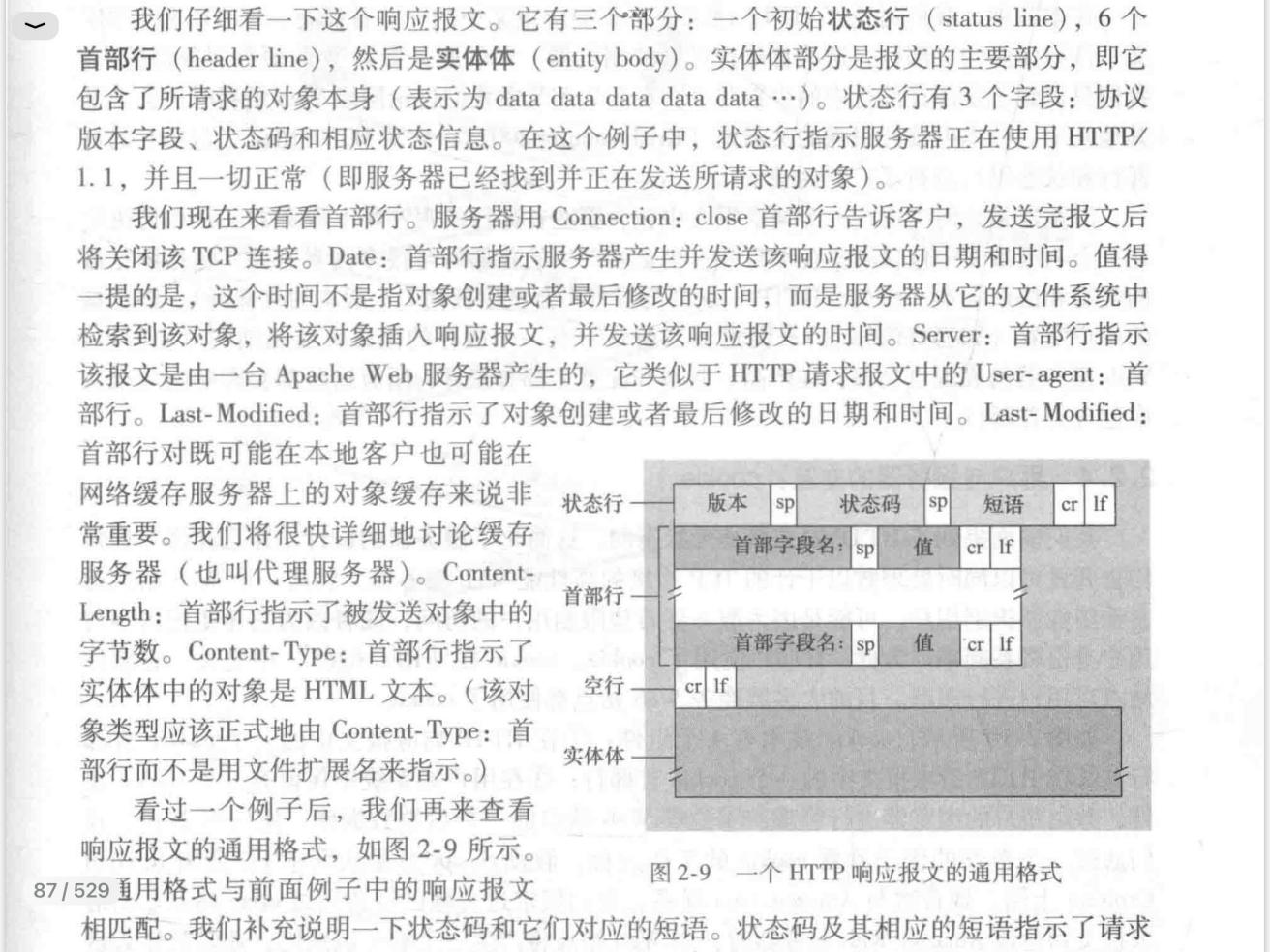
(2 HTTP响应报文

HTTP响应报文是对请求报文的响应。



响应报文有三个部分：**1个初始状态行、6个首部行、之后是实体体**。**实体体是报文的主要部分**，包含了所请求的对象本身(表示为data data data...)。状态行有3个字段： **协议版本字段、状态码、相应状态信息**。这个例子中服务器正在使用HTTP/1.1，并且一切正常。

**首部行Connection:close** 告诉客户，发送完报文后将**关闭该TCP连接**。**Date:** 指示服务器产生并发送该响应报文的**日期和时间**。**Server：**指示该报文是由一台Apache Web**服务器**产生的，类似于HTTP请求报文中的User-agent: 首部行。Last-Modified：指示对象创建或**最后修改**的日期和时间。**Content-Length**:首部行指示了被发送对象中的**字节数**。**Content-Type**:首部行指示了**实体体中的对象**是HTML文本。



一些常见的状态码和相关的短语：

**200 OK:** 请求成功，信息在返回的响应报文中。

**301 Moved Permanently:** 请求的对象被永久转移了，新的URL定义在响应报文的Location:首部行中，客户软件将自动获取更新的URL。

**400 Bad Request**: 通用差错代码，该请求不能被服务器理解。

**404 Not Found**：被请求的文档不在服务器上

**505 HTTP Version Not Supported**: 服务器不支持请求报文使用的HTTP协议版本。

1. cookie

一个Web站点希望能够**识别用户身份**，会使用**cookie**。cookie允许站点对用户进行跟踪。

cookie有4个组件：①在HTTP**响应报文**中的一个cookie首部行；②在HTTP**请求报文中**的一个cookie首部行③在**用户端系统**中保留有一个cookie文件，并由用户的**浏览器进行管理**；④位于Web站点的一个**后端数据库**。

在访问服务器的Web站点时，该站点将产生一个**唯一识别码**如1678，并以此为索引在它的后端数据库中产生一个**表项**。接下来Web服务器用一个包含**Set-cookie:首部**的HTTP响应报文对用户的浏览器响应。首部行包括**Set-cookie:1678**。

当用户的浏览器收到该HTTP响应报文时，会在它**管理的特定cookie文件**中添加一行，该行包括**服务器的主机名**和**识别码**。当用户每请求一个Web页面，浏览器会查询该cookie文件并抽取用户对这个网站的识别码，并放到HTTP**请求报文中的首部行**中，如Cookie:1678。

在这种方式下，网站服务器可以**跟踪**用户在站点的活动(访问顺序、访问时间)

1. Web缓存

**Web缓存器**也叫**代理服务器**。是能够代表初始Web服务器来满足HTTP请求的网络实体。Web缓存器有自己的磁盘存储空间，并在存储空间中保存**最近请求过的对象的副本(该对象在初始服务器)**。Web接收客户的HTTP请求一个对象，如果Web缓存器中没有该对象，就打开**TCP连接**到初始服务器，发送一个对该对象的HTTP请求。当Web接收到该对象时，会在**本地存储空间**存储一份副本，并向客户的HTTP响应报文发送**该副本**。

1. 条件GET方法

HTTP协议有一种机制，允许缓存器证实它的对象**是最新的**。这种机制就是**条件GET方法**。如果：请求报文**使用GET方法**且**包含If-Modified-Since**: 首部行，则这个HTTP请求报文就是一个条件GET请求报文。

缓存器发送包含If-Modified-Since的HTTP请求，其值等于缓存该文件时的日期时间。若未被修改，则Web服务器会发送响应报文，包含304 Not Modified状态行，表示该对象没有被修改，响应报文中不会存储该对象。

使用HTTP传送前不用将多媒体数据编码为ASCII码

1. 电子邮件

因特网电子邮件系统主要有3个组成部分：**用户代理、邮件服务器、简单邮件传输协议(SMTP)**。

用户代理允许用户阅读、回复等对报文进行操作。当撰写完邮件后，邮件代理**向邮件服务器**发送邮件，此时邮件放在邮件服务器的**外出报文队列**中，接收方的**用户代理**在其邮件服务器的邮箱中取得该报文。

每个接收方在其中的某个邮件服务器上有一个邮箱。邮件发送过程是：**发送方的用户代理——发送方的邮件服务器——接收方的邮件服务器——接收方的邮箱——接收方的用户代理**

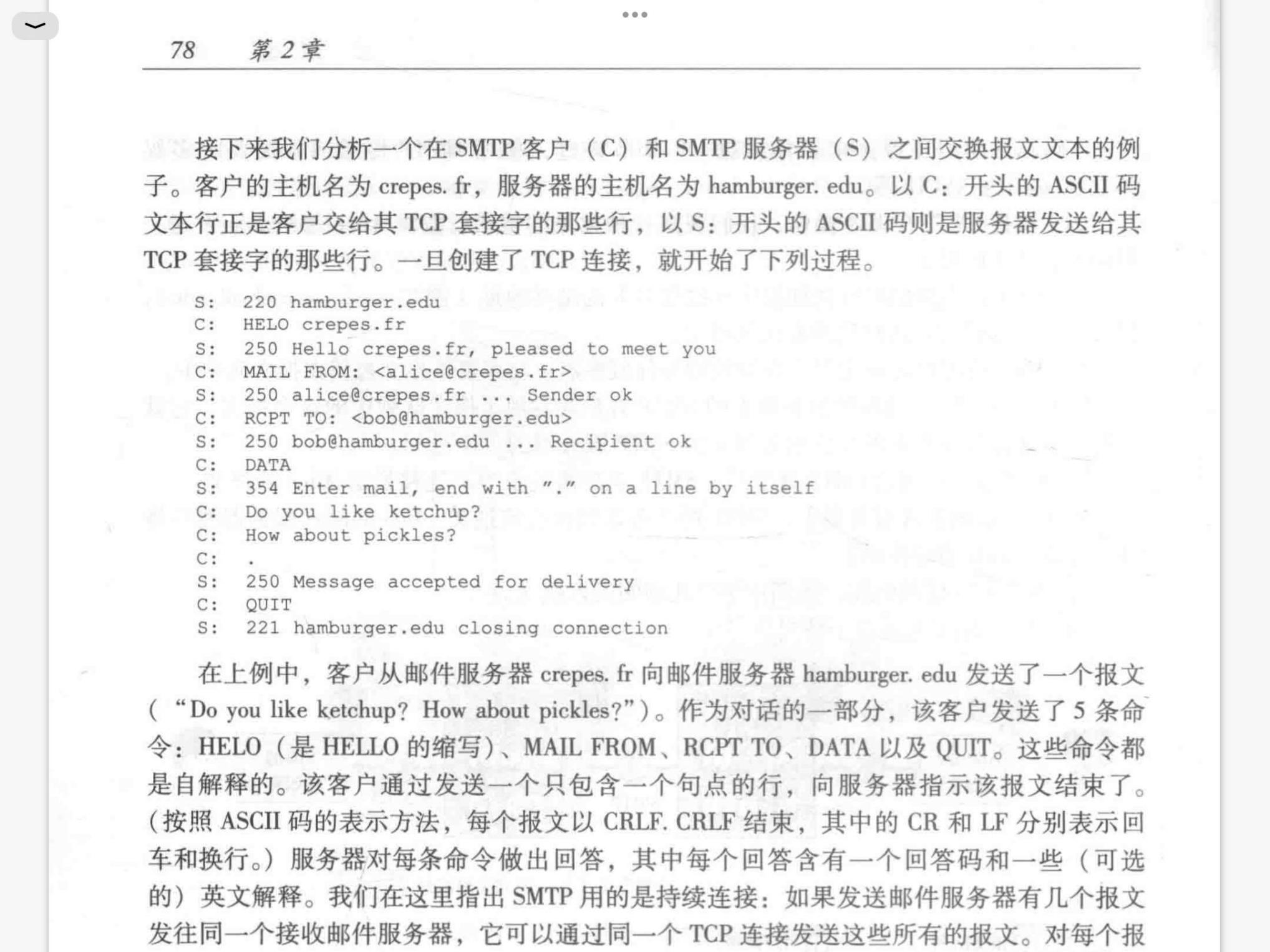
**SMTP**是电子邮件的主要应用层协议，使用TCP作为运输协议。SMTP有**两个部分**：运行在发送方邮件服务器的**客户端**和运行在接收方邮件服务器的**服务器端**。发送邮件的是客户，接收邮件的是服务器。SMTP之间会建立TCP连接。

SMTP邮件报文的**体**只能采用简单的**7比特ASCII码**表示

SMTP一般不使用中间邮件服务器发送邮件

1. 邮件的交换报文文本(客户对客户服务器的命令)

设客户的主机名为crepes.fr，服务器的主机名为hamburger.fr。SMTP客户为C，SMTP服务器为S。



以C开头的是**客户服务器**交给其**TCP套接字**的行，以S开头的是**接收方服务器**发送给其**TCP套接字**的行。

客户发送了**5条命令**：**HELLO、MAIL FROM、RCPT TO、DATA、QUIT**。客户通过发送一个**只包含一个句点.**的行，向接收方 服务器表示该报文结束。

服务器对每条命令作出回答，每个回答含有一些**回答码**和一些**英文解释**。SMTP是**持续连接**，即多个报文会通过一个TCP连接发送。

对每个报文，客户用一个**新的MAIL:FROM:行开始**，用一个独立的句点表示结束。当**所有邮件发送完后**才发送QUIT。

1. SMTP与HTTP对比

HTTP是从**Web服务器向Web客户**(通常是浏览器)传送文件。SMTP**从一个邮件服务器向另一个邮件服务器**传送文件。二者都使用**持续连接**。

HTTP是**拉协议**，用户使用HTTP从已装载信息的服务器上拉取信息，TCP连接是由**想接收文件的机器**发起的；SMTP是**推协议**，TCP连接是由**发送该文件的机器**发起的。

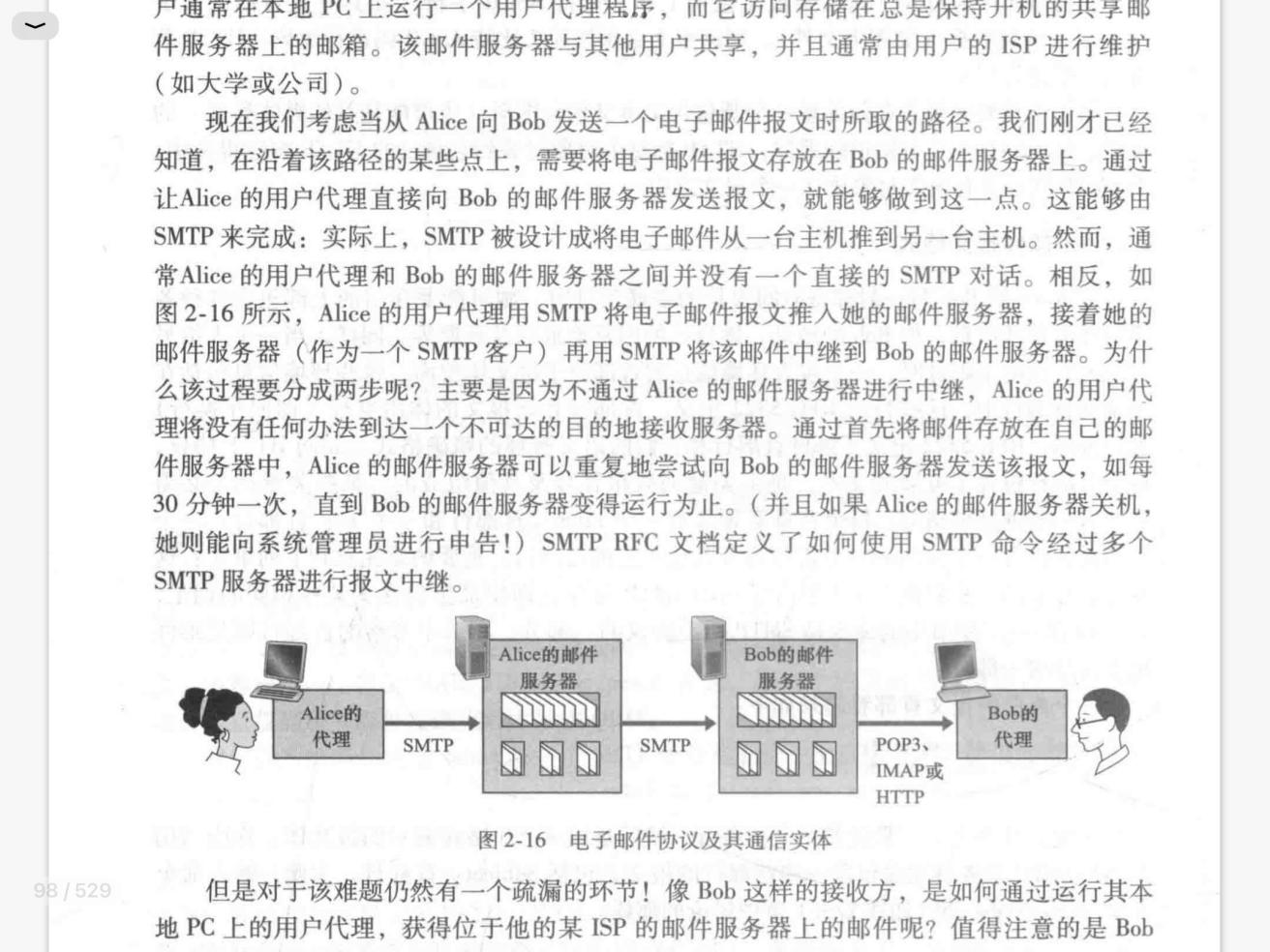
1. 邮件报文格式

首部行和报文的体用**空行**分隔，每个首部**必须含有一个From:、To:**。也许含有一个Subject以及其他的可选首部行。

1. 邮件访问协议

发送方用SMTP将**邮件从用户代理推入邮件服务器**，邮件服务器再用SMTP将邮件中继到接收方服务器。

取报文是一个**拉操作**，不能使用SMTP推协议，需要**特殊的邮件访问协议**，将邮件服务器上的报文传给本地PC，这些协议有第三版的邮局协议(**POP3**)、因特网邮件访问协议(**IMAP**)、**HTTP**。



1. POP3

当用户代理(客户)打开到邮件服务器(服务器)端口110上的TCP连接后，POP3开始工作。POP3按照**3个阶段工作**：**特许、事务处理、更新**。

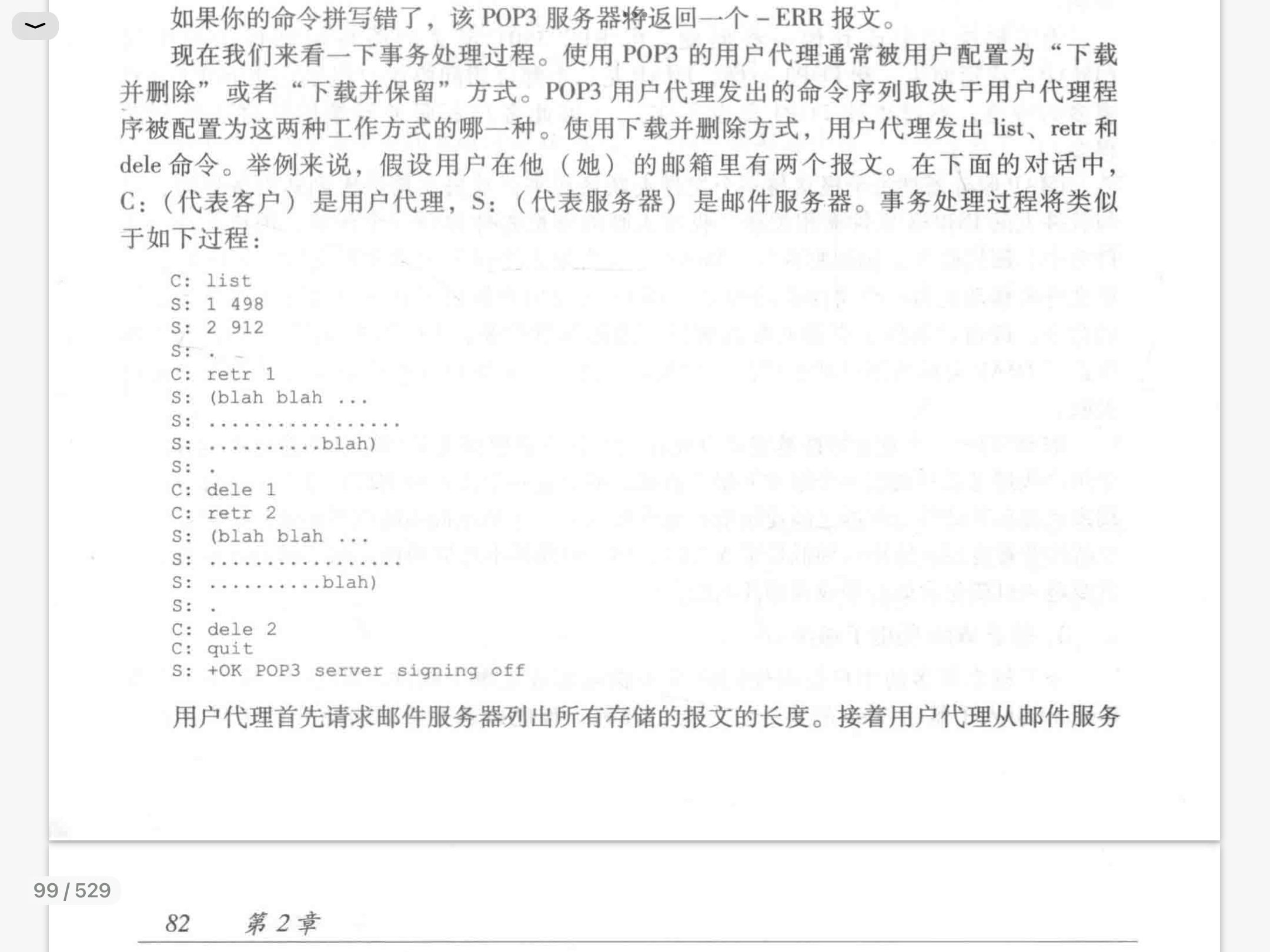
在特许阶段，用户代理发送用户名和口令来**鉴别用户**；

在事务处理阶段，用户代理**取回报文，报文**，还能进行对报文做删除标记等操作；

更新阶段出现在客户发出**quit命令后**，目的是**结束POP3会话**，邮件服务器会删除有删除标记的报文。

特许阶段有两个主要的命令：user<user name>和pass<password>。分别输入用户名和密码

事务处理阶段，用户代理发出一些命令，服务器作出回答。回答有：+OK表示前面的命令是正常的；-ERR表示前面的命令有错。使用POP3的用户代理通常被配置为**“下载并删除”或“下载并保留”**。使用下载并删除方式，用户代理向邮件服务器发出list、retr、dele、quit命令。在**quit命令后**，POP3服务器更新并**从服务器上**删除邮件。



1. IMAP

POP3协议不能提供任何在服务器上**创建远程文件夹**等方法，IMAP可以。IMAP**将每个报文与文件夹联系**起来，与收件人的INBOX文件夹相关联。IMAP还提供了查找、移动邮件等服务。

1. 基于Web的电子邮件

基于Web的电子邮件，用户代理是普通的浏览器，用户代理与服务器之间使用HTTP协议而不是SMTP协议，邮件服务器之间发送和接收邮件使用SMTP协议。

1. DNS

因特网上的主机可用多种方式标识，一种是用主机名，如www.facebook.com、www.google等；主机也可用IP地址标识。

需要一种能进行主机名到IP地址转换的目录服务，这就是**域名系统(DNS)**。DNS是一个由分层的**DNS服务器**实现的分布式数据库，也是一个使得主机能够**查询分布式数据库**的**应用层协议**。DNS服务器通常是运行BIND软件的**UNIX机器**。DNS协议运行在**UDP**之上，使用**53**号端口。

DNS作为应用层协议的不同之处是不直接与用户接触，通常由其他应用层协议所使用，包括HTTP、SMTP、FTP。

在浏览器中请求URL时，该主机也运行着DNS应用的客户端。浏览器会将URL包含的主机名传给DNS客户端，最终浏览器会得到主机的IP地址，进而发起TCP连接

1. DNS服务器

DNS服务器大致有**3**种类型：**根DNS服务器、顶级域(TLD)DNS服务器、权威DNS服务器**。

访问www.amazon.com，客户首先与**根服务器**之一联系，它将返回顶级域名为com的**TLD服务器**的IP地址；客户再与**TLD服务器之一**联系，它将返回amazon.com的**权威服务器**的IP地址；最后客户与amazon.com**权威服务器之一**联系，它为主机名www.amazon.com返回其IP地址。

根DNS服务器有400多个，遍及全世界。

1. 本地DNS服务器

每个**ISP**都有一台**本地DNS服务器**。当主机与某个ISP连接时，该ISP提供一台主机的IP地址。主机发出DNS请求，该请求被发往本地DNS服务器，它起着**代理**的作用，并将该请求**转发到DNS服务器层次结构**中。

1. DNS缓存

当某DNS服务器接收一个DNS回答时，它能将映射**缓存在本地存储器**中。DNS服务器将在一段时间后丢弃该缓存信息。

1. DNS资源记录

共同实现DNS分布式数据库的所有DNS服务器都存储了**资源记录(RR)**，RR提供了**主机名到IP地址的映射**。每个**DNS回答报文**包含了一条或多条资源记录。

资源记录是一个包含了下列字段的4元组：

(Name, Value, Type, TTL)

TTL是该记录的**生存时间**，决定资源记录应当**从缓存中删除**的时间。

如果Type = **A**, 则Name是**主机名**，Value是**该主机名对应的IP地址**。

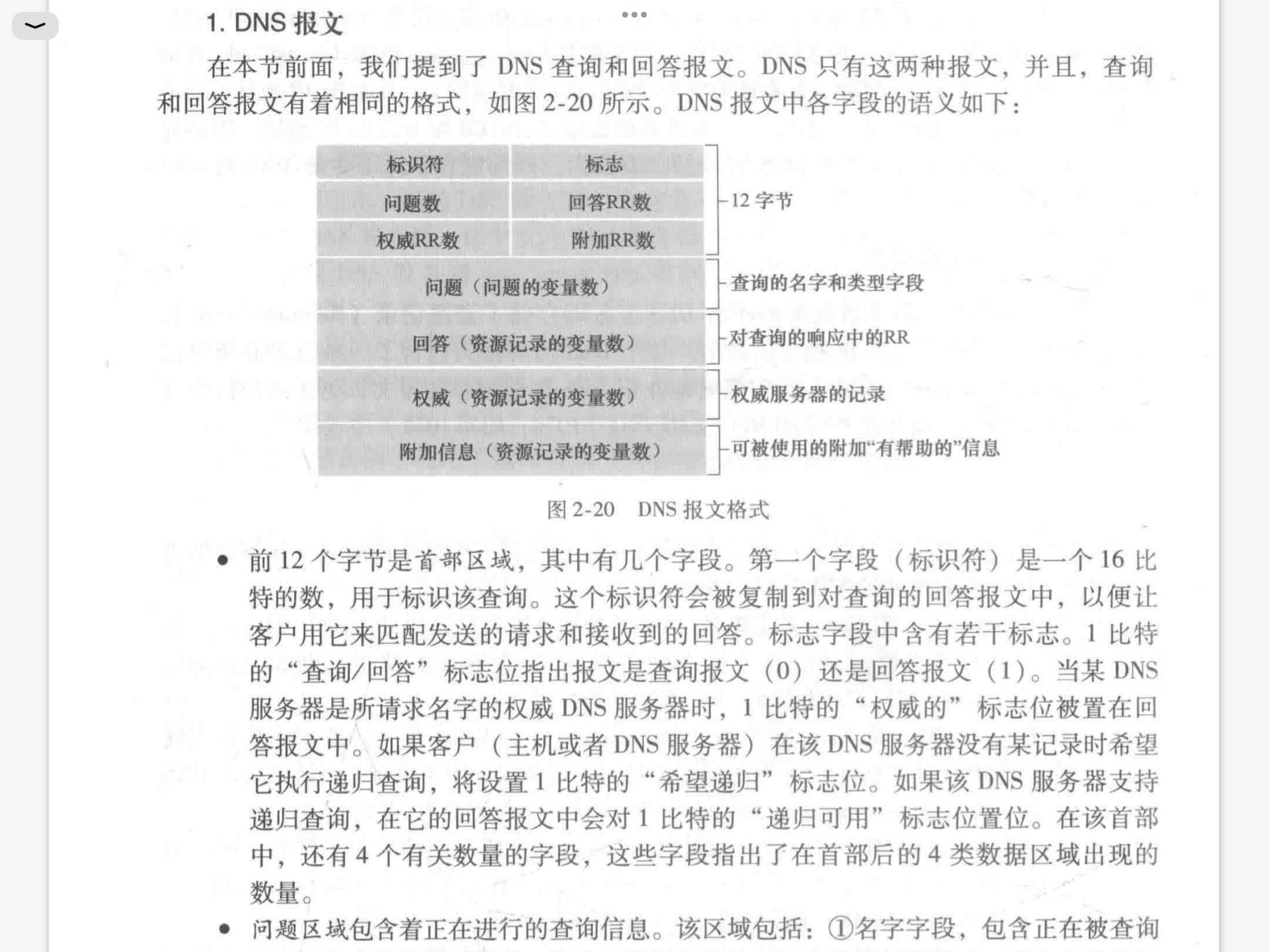
如果Type = **NS**, 则Name是**域**如(foo.com)，**Value**是知道主机IP地址的**权威DNS服务器的主机名，如dns.foo.com**。这会发生在不用于该主机名的权威服务器，该服务器**也会发送一条A记录**，其中的Value字段记录正确的DNS域记录的服务器的IP地址。如**(umass.edu, dns.umass.edu, NS); 和(dns.umass.edu, 128.119.40.111, A)。**

如果Type = **CNAME**，则Value是**规范主机名**，Name是**别名**。

如果Type = MX，则该主机是**邮件服务器**，Name是邮件服务器的**别名**，Value是**规范主机名**。

1. DNS报文

DNS只有查询报文和回答报文两种，且具有相同格式。各字段含义如下：



前12字节是**首部区域**。标识符的标志字段有若干标志，1比特的“查询/回答”标志位指出报文是查询报文(0)还是回答报文(1)。

**问题区域**包含正在进行的查询信息，该区域包括：①名字字段，包含被查询的**主机名**；②类型字段，主机地址与一个名字相关(A类型)，与某个名字的邮件服务器相关(MX类型)

**回答区域**包含了对最初请求的名字的资源记录。

**权威区域**包含了其他权威服务器的记录。

**附加区域**包含了其它有帮助的记录。

1. P2P文件分发

P2P中，连接的主机彼此**直接通信**，称为**对等方**。在服务器向多个客户分发文件时，如果客户彼此是对等方，则可以**互相重发**它已经收到的该文件的任何部分，**协助该服务器**。

最流行的p2p文件分发协议是**BitTorrent**，有许多BitTorrent客户软件。

1. 带宽

**带宽**（bandwidth）是指在特定的时间内，**能够传输数据的最大速率**。在计算机和网络通信领域中，带宽通常用来描述数据传输的速率，以比特率（bits per second）或字节率（bytes per second）来衡量。例如，一个网络连接的带宽为100 Mbps（兆比特每秒），表示该连接每秒最多可以传输100兆比特的数据。

1. BitTorrent

BitTorrent是一种用于文件分发的流行P2P协议。参与一个特定文件分发的**所有对等方的集合**被称为一个**洪流**。在一个洪流中的对等方彼此下载**等长度**的文件块，典型的块长度为256KB。当一个对等方下载块时，也为其他对等方上载了多个块。

每个洪流具有一个基础设施节点，称为**追踪器**。当一个对等方加入洪流时，向追踪器**注册**自己，并**周期性地**通知追踪器自己的存在，这样追踪器可以跟踪洪流中的对等方。

当一个新的对等方(用户)加入洪流时，追踪器随机选择之前洪流的一个**子集**，将这个子集的IP地址发送给用户。用户试图与该列表上的**所有对等方创建并行的TCP连接**，成功连接的对等方为“**邻近对等方**”。

在决定请求哪些块的过程中，使用**最稀缺优先**，即先请求在用户的**邻居中副本数量最少的块**。

决定响应请求，使用**对换算法**。用户根据当前能够以**最高速率**向他提供数据的邻居，给出其优先权。特别的，用户对每个邻居都持续地持续地测量接收到的比特的速率，并确定以最高速率流入的4个邻居。每过10s，**重新计算该速率**并可能**修改**这4个对等方的集合。这4个对等方被称为**疏通**。每过30s，随机选择另一个邻居并向其发送块。

1. 因特网视频

视频能够被压缩，可以用比特率来权衡视频质量。比特率越高图像质量越好。

1. HTTP流和DASH

在HTTP流中，视频只是存储在HTTP服务器中的一个**普通文件**，有特定的URL。客户与服务器创建一个TCP连接并发送对该URL的HTTP GET请求。在客户侧，字节被收集在**客户应用缓存**中，缓存字节数超过门限时会播放。

HTTP的动态适应性流(DASH)：在DASH中，视频编码分为几个不同的版本，每个版本具有不同的比特率，对应于不同的质量水平。使用DASH后，每个视频版本存储在HTTP服务器中，有一个不同的URL。

1. 内容分发网

为了应对向分布于全世界的用户分发巨量视频数据的挑战，几乎所有主要的视频流公司都利用**内容分发网(CDN)**。CDN管理分布在多个地理位置上的服务器，在它的服务器中存储视频副本，向最近用户提供视频。CDN可以是专用CDN、第三方CDN。

1. 套接字编程

典型的网络应用是由**一对程序(客户程序和服务器程序)**组成的，它们位于**两个不同的端系统**中。运行这两个程序时，创建了一个客户进程和一个服务器**进程**，同时它们通过从**套接字**读出和写入数据在彼此之间进行通信。创建网络应用时，主要任务是**编写客户程序和服务器程序**的代码。同时需要选择UDP或TCP。

网络应用程序分为**开放和专用**两种，应用的**协议**分别是**公开的和未公开的**。

1. UDP套接字编程

一台主机可能运行许多网络应用进程，**每个进程**具有一个或多个**套接字**，在目的主机**指定特定的套接字**也是必要的。当生成一个套接字时，就为它分配一个**端口号**。

附在分组上的**目的地址**包含目的主机的**IP地址**、目的地套接字的**端口号**。

发送方的**源地址**也是由源主机的IP地址和源套接字的端口号组成，该源地址也要附在分组上，这个过程通常由**操作系统自动完成**。

客户端：

from socket import \* #该socket模块形成了python中所有网络通信的基础，利用这个库可以创建套接字  
serverName = 'hostname' #提供或包含 服务器的 IP地址或主机名。  
serverPort = 12000  
clientSocket = socket(AF\_INET, SOCK\_DGRAM) #创建套接字，称为clientSocket。第一个参数是地址簇，AF\_INET指示了底层网络使用了IPv4  
#第二个参数指示套接字为UDP套接字  
message = input('输入小写单词:') #获取输入，作为报文  
clientSocket.sendto(message.encode(), (serverName, serverPort))#encode()方法将报文由字符串类型转换为字节类型，  
# 因为需要向套接字中发送字节。这里的sendto()方法需要为报文附上目的地址(serverName, serverPort)  
# 并向进程的套接字clientSockcet发送结果分组  
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)#当分组到达客户套接字时，分组的数据被放置到变量ModifiedMessage中，  
#源地址被放到变量serverAddress中，包含服务器的IP地址和服务器的端口号。参数2048表示取缓存长度2048作为输入  
print(modifiedMessage.decode()) #将报文从字节转化为字符串并打印  
clientSocket.close() #关闭套接字

服务器端：

from socket import \*  
serverPort = 12000  
serverSocket = socket(AF\_INET, SOCK\_DGRAM)  
serverSocket.bind('', serverPort) #将端口号12000与该服务器的套接字绑定在一起。  
print("服务器准备好接收")  
while True:  
 message, clientAddress = serverSocket.recvfrom(2048)  
 modifiedMessage = message.decode().upper()  
 serverSocket.sendto(modifiedMessage.encode(, clientAddress)) #将分组发送到服务器的套接字中

1. TCP套接字编程

创建TCP连接时，需要将其与**客户套接字**地址和**服务器套接字**地址关联起来。

服务器上有**欢迎套接字与连接套接字**。欢迎套接字是所有要与服务器通信的客户的**起始接触点**(在这里进行三次握手)，连接套接字是随后**为每个客户通信而生成**的套接字。

TCP客户端：

from socket import \*  
serverName = '172.245.79.47'  
serverPort = 12000  
clientSocket = socket(AF\_INET, SOCK\_STREAM) #创建了客户的套接字，称为clientSocket。第一个参数表示底层网络使用IPv4，  
# 第二个参数表示该套接字是套接字  
clientSocket.connect((serverName, serverPort)) #发起客户和服务器之间的TCP连接，用于握手。connect的参数是服务器端的地址  
#执行完这句代码后，执行了三次握手，并建立起了一条TCP连接  
sentence = input("输入小写句子：") #输入  
clientSocket.send(sentence.encode()) #通过该客户的套接字并为TCP连接发送字符串sentence。TCP连接无需显式创建分组并为分组附上目的地址  
modifiedSentence = clientSocket.recv(1024) #接收服务器的字节  
print("从服务器接收到", modifiedSentence.decode())  
clientSocket.close() #关闭套接字

服务器端：

from socket import \*  
serverPort = 12000  
serverSocket = socket(AF\_INET, SOCK\_STREAM) #创建欢迎套接字  
serverSocket.bind(('', serverPort))  
serverSocket.listen(1) #让服务器聆听来自客户的TCP连接请求，设置最大请求连接数为1  
print("服务器准备接收")  
while True:  
 connectionSocket, addr = serverSocket.accept() #当客户敲门时，调用accept()方法创建新的套接字，名称为connectionSocket  
 #这个套接字用于敲门的特定用户，建立TCP连接  
 sentence = connectionSocket.recv(1024).decode()  
 capitalizeSentence = sentence.upper()  
 connectionSocket.send(capitalizeSentence.encode()) #通过套接字发送到TCP连接  
 connectionSocket.close() #关闭套接字

1. 运输层

运输层的协议是在**端系统**中实现的。在发送端，运输层从发送进程接收到报文，转换为**运输层分组(报文段)**。之后运输层将报文段传递给网络层。在接收端，运输层处理接收到的报文段，使其为接收应用进程使用。运输协议能够提供的服务常常**受制于底层网络层协议**的服务模型。

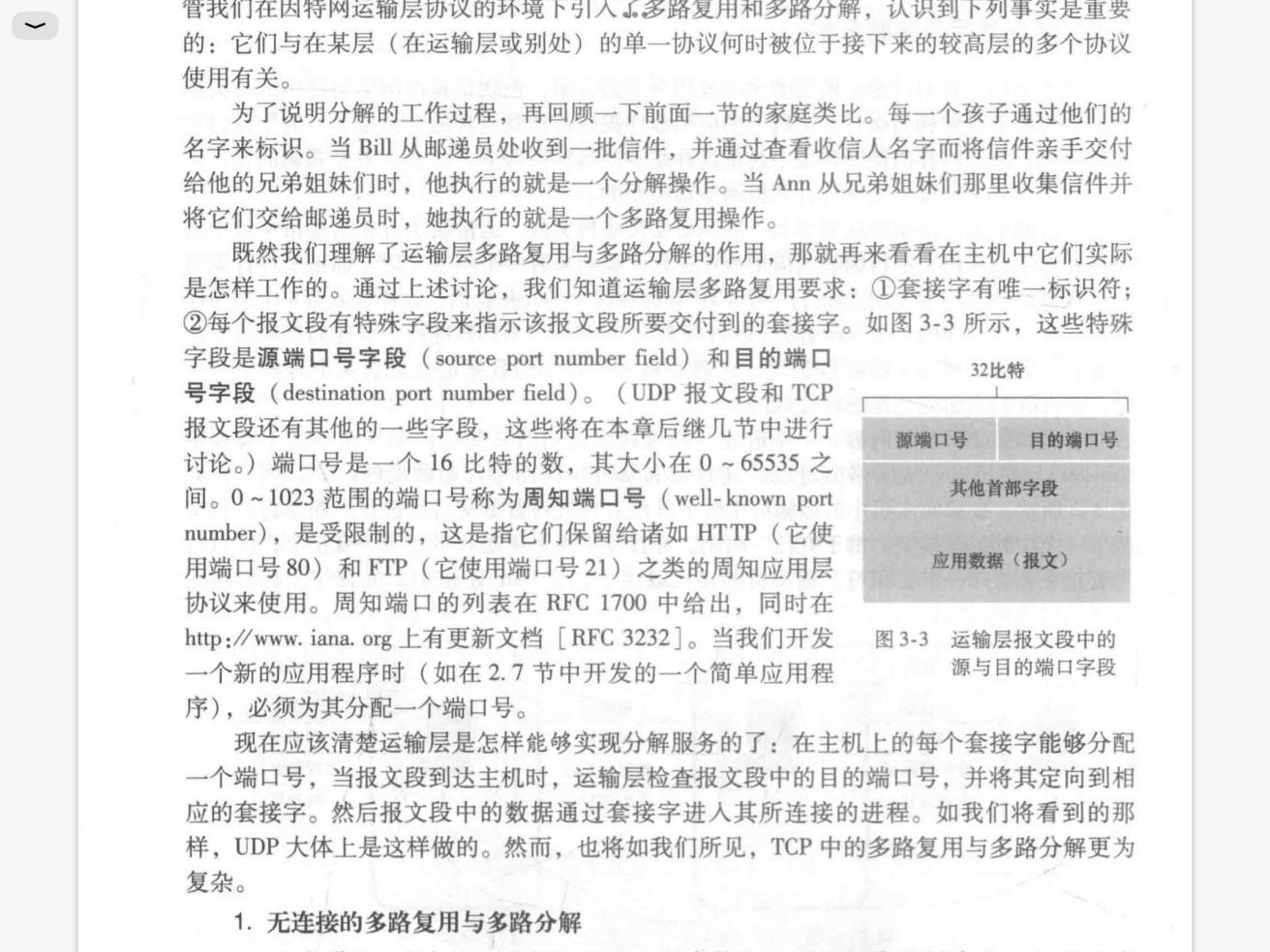
UDP(用户数据传输协议)不可靠、无连接

TCP(传输控制协议)可靠、面向连接

将主机间交付扩展到进程间交付被称为**运输层的多路复用与多路分解**。将运输层报文段中的数据**交付到正确的套接字**的工作为**多路分解**。从不同套接字中**接收数据块**，并为每个数据块封装首部信息从而**生成报文段**，然后将报文段传递到网络层，称为**多路复用**。

在多路复用中，每个套接字有**惟一标识符**，每个报文段有**特殊字段**来指示该报文段所要交付到的套接字。这些特殊字段是**源端口字段、目的端口字段**。

端口号是**16bit**的数，大小在 **0~65535**之间。0~1023是受限制的**周知端口号**，已被HTTP(80端口)、FTP（21）之类的周知应用层协议来使用。



1. 在UDP套接字编程中，

clientSocket = socket(AF\_INET, SOCK\_DGRAM)

这会创建一个UDP套接字，运输层会**自动**为套接字分配一个端口号，该端口号是当前未被主机中任何其他UDP端口使用的号。**另一种指定方式**是在**创建套接字**后： clientSocket.bind(‘’, 19157)

通过套接字**bind()**方法为UDP套接字关联一个特定的端口号(如19157).

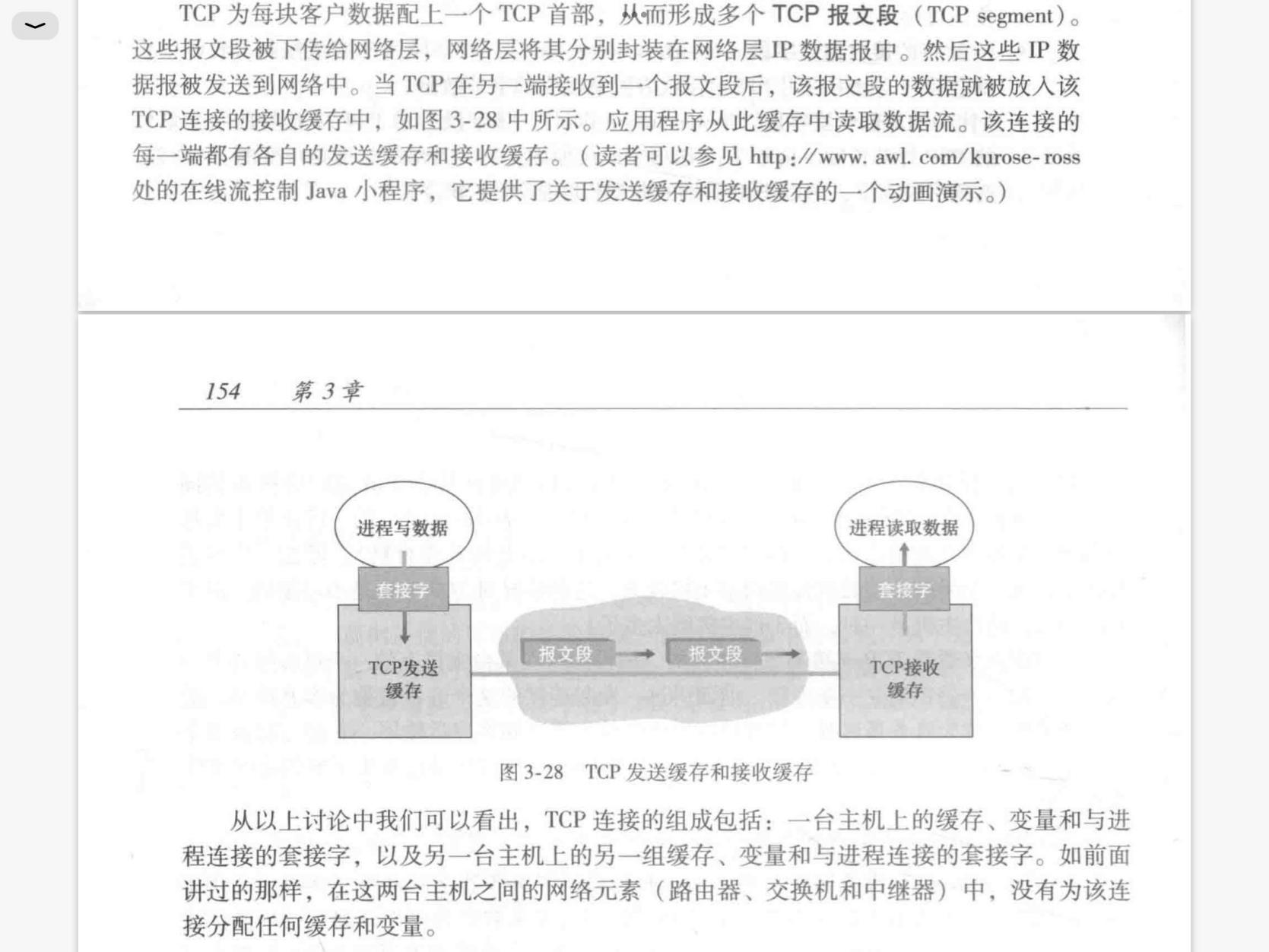
1. UDP

**DNS**是一个通常使用UDP的应用层协议。**QUIC**是**快速UDP因特网连接协议**，用于Chrome浏览器。

1. TCP

TCP提供**全双工服务**：一台主机上的进程A与另一台主机上的进程B存在一条TCP连接，则应用层数据**在进程B流向A时，也可由A流向B**。TCP只能用于**一个发送方和一个接收方**之间。

三次握手：客户首先发送一个特殊的TCP报文段，服务器用另一个特殊的TCP报文段来响应，最后客户再用第三个特殊报文段作为响应。**前两个报文段不包含应用层数据，第三个报文段可以包含**。**三个报文段的存在**构成了**三次握手**。



数据通过套接字后，进入TCP**发送缓存**中。发送缓存是在三次握手期间设置的缓存之一。TCP接下来会**不时地**从发送缓存里取出一块数据传递到网络层。TCP可从缓存中取出并放入**报文段中的数据量受限于最大报文段长度(MSS)**。MSS由本地发送主机发送的**最大链路层帧长度**(即**最大传输单元（MTU）)**来设置。

1. 网络层

在网络中的每一台主机和路由器**都有一个网络层部分**。网络层能被分解为两个相互作用的部分，即**数据平面和控制平面**。网络层的**数据平面**是网络层中**每台路由器的功能**，决定到达路由器输入链路之一的数据报(网络层的分组)**如何转发**到该路由器的输出链路之一。**控制平面**则控制数据报沿着从源主机到目的主机的端到端路径中**路由器之间的路由方式，**涉及路由选择算法、路由选择协议。

网络层的作用是将分组从一台发送主机移动到一台接收主机。两种重要的**网络层功能**有**转发、路由选择**。**转发**是将分组从一个输入链路接口转移到适当的输出链路接口的**路由器本地动作**。路由选择是**确定**分组从源到目的地**所采取的端到端路径**的网络范围处理过程。=，是在分组传输前决定的。

每台路由器中都有**转发表**，路由器依靠到达的**分组首部的字段值**并将这些首部值在转发表中**索引**，得到**输出链路接口**。

1. 网际协议：IPv4、寻址、IPv6

IP版本4被称为IPv4。IP版本6被称为IPv6，已被提议替代IPv4。这些都是IP协议的不同版本。

1. IPv4编址

主机与物理链路之间的边界称作**接口**。一个**IP地址**与**一个接口**相关联。IP地址是**点分十进制记法**，一共有**32比特**表示IP地址，每个字节都用十进制书写，如193.32.216.9。

一台路由路的3个接口和3个主机接口相连，形成的网络称为**子网**。IP编址为这个子网分配一个地址223.1.1.0/24。其中/24记法称为**子网掩码**，指示32比特中最左侧24比特定义了**子网地址**。其他任何连接到223.1.1.0/24网络的主机都要求其地址具有223.1.1.xxx的形式。

1. 获取主机地址

组织获得地址之后，可为本组织内的主机与路由器接口逐个分配IP地址。主机地址可用**动态主机配置协议(DHCP)**来完成，DHCP允许主机**自动获取**一个IP地址。DHCP能让主机每次与网络连接时能得到相同或不同的IP地址。

1. IPv6

IPv6的地址长度从32比特增加到128比特。

1. 链路层和局域网

在两台主机之间，数据报通过**有线链路或无线链路**传输。运行**链路层协议**的设备称为**节点**。在链路中，**传输节点**将数据报封装在**链路层帧**中，并将该帧传送到链路中。不同的运输方式就是链路层协议。

路由器中，链路层是实现在路由器的**线路卡**中。主机中，链路层的主体部分是在**网络适配器**中实现的，有时被称为**网络接口卡**。

1. 链路层寻址

主机和路由器的适配器都有**链路层地址**。链路层地址有各种不同的称呼(叫法)：**LAN地址、物理地址、MAC地址**。对大多数局域网而言，MAC地址由**6字节**表示，通常用16进制表示。适配器的MAC地址**不变**。

局域网中，ARP将IP地址解析为一个MAC地址。ARP只为在同一个子网上的主机和路由器接口解析IP地址。

以太网（Ethernet）是一种有线局域网（LAN）技术。

1. 无线网络

端系统设备被称为**无线主机**。主机通过**无线通信链路**连接到一个基站。

1. WIFI

IEEE 802.11 无线LAN称为WIFI。

1. SSL

TCP的强化版本被称为SSL，SSL版本3的一个稍加修改的版本被称为运输层安全性(TLS)。URL以https开始而不是http则使用了SSL。

1. 14.
2. 15.
3. 16.
4. 17.
5. 18.
6. 19.
7. 20.