目录:

1. 计算机网络和 Internet

time : 2021-06-03

计算机网络是通信技术和计算机技术结合的产物。

在狭义的计算机网络中，通信主体为传统的计算设备，如：台式计算机，服务器等等，这些计算设备在计算机网络中称为主机。主机间通信的信息为数字化的信息。因此计算机网络是一种特殊的通信网络。

当主机在地理位置上的分布较近时，主机间可以直接相连。但距离较远时，考虑到费用，计算机网络引入了分组交换网络。

分组交换网络是一种计算机网络，只不过主机和主机之间不直接相连。而是通过分组交换机或分组交换机互联形成的网络间接相连。

如今，我们日常使用的 Internet 是最大的分组交换网络。

本书我们以 Internet 作为讨论计算机网络的主要载体。

第 1 节将讨论 Internet 的组成，解释 Internet 是什么的问题。我们将了解到端系统，通信链路，分组交换机以及网络协议。

第 2 节将讨论 Internet 提供的服务。我们每天都在享受 Internet 提供的服务。在这一节，我们会了解到分布式应用程序和套接字。

第 3 节将讨论 Internet 的结构。Internet 是最大的分组交换网络，那么它是怎么组织的呢？第 3 节我们将回答这个问题。我们会了解到 Internet 是由 ISP 网络互联形成的复杂结构。

第 4 节将引入 Internet 分层模型。我们将了解到分层组织 Internet 的好处。我们将知道 Internet 分层模型自顶向下依次为：应用层，运输层，网络层，链路层，物理层。本书就是依据分层模型所组织。

第 5 节将介绍一些抽象但常用的概念，这些概念和计算机网络的性能相关。我们将了解到时延，丢包，吞吐量。

第 6 节将介绍网络的安全性。我们会了解 Internet 并不是一个绝对安全的系统。

第 7 节我们以计算机网络的历史结束本章的理论叙述。

第 8 节我们将介绍 wireshark 分组嗅探器，这为我们以后的实验打好基础。在本书中我们有多个实验，通过亲自做这些实验，我们会加深对计算机网络的理解。

1.1. Internet 的组成

Internet 是端系统通过通信链路和分组交换机相连形成的最大的通信网络，端系统和分组交换机以及其他一些部件上运行着许多网络协议以便端系统正常交换信息。

图 1-1 描述了一个典型的 Internet。下面我们结合图 1-1 说明 Internet 的各个组成部分。

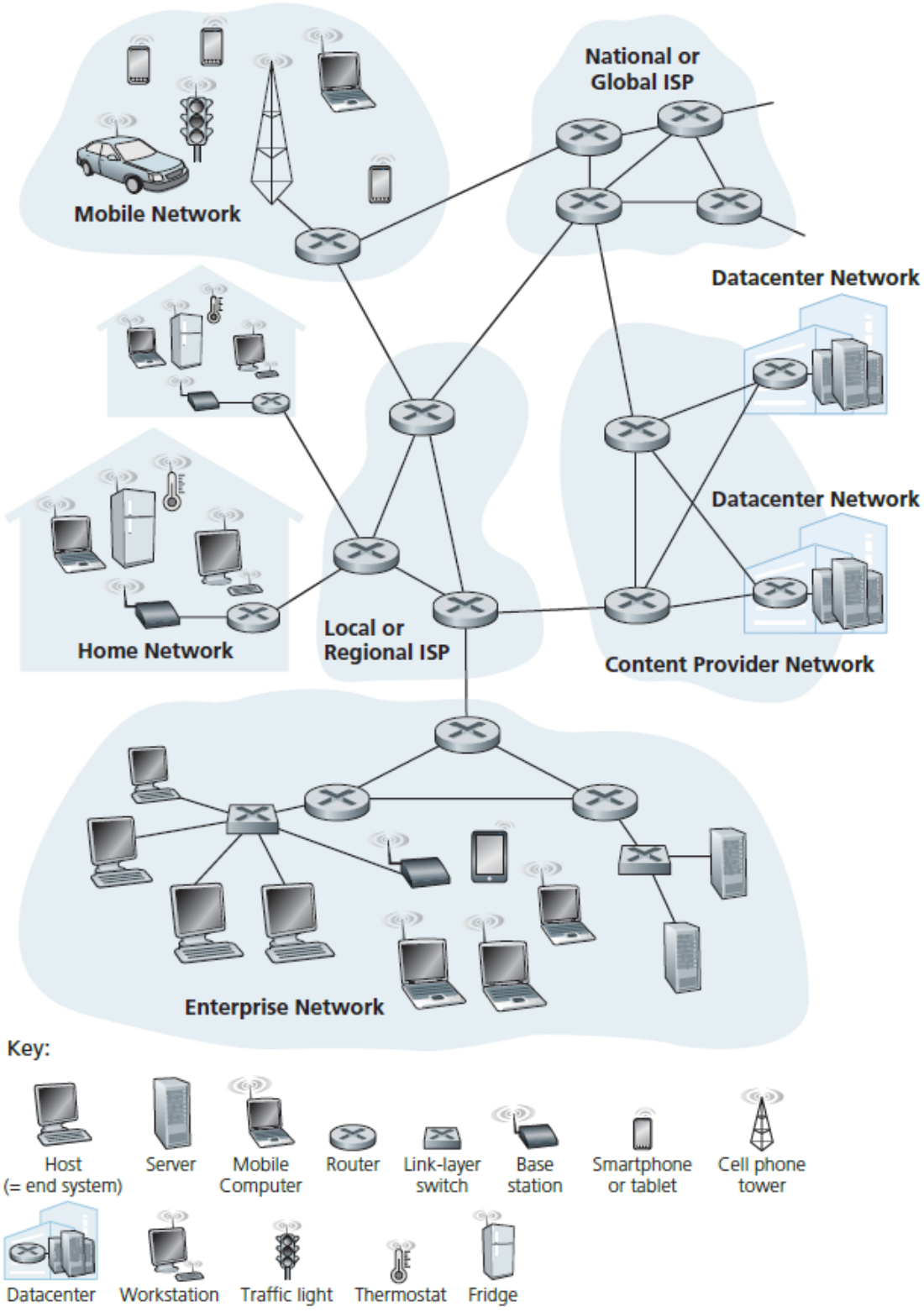


Figure 1.1 ♦ Some pieces of the Internet

1.1.1. 端系统

不久前，Internet 是一个连接着全世界范围内计算设备，如桌面 PC，Linux 工作站，以及服务器的网络。但如今，计算设备已经发生了巨大的变化，如智能手机，平板电脑，电视，游戏机，智能家用电器，汽车，这些非传统计算设备已经接入了 Internet。这些与 Internet 相连的计算设备，称为 **端系统(end system)** 或 **主机(host)** 。

1.1.2. 通信链路

通信链路(communication link) 分为不同的 **物理媒体(phys1cal medium)**，如电缆，铜线，光纤，和无线信道。不同物理媒体的通信链路有不同的 **传输速率(transmiss1on rate)**，传输速率以 bit/s(bps) 度量。

这些物理媒体分为 2 类：**导引型媒体(guided media)** 和 **非导引型媒体(unguided media)**。

导引型媒体中，信号沿着固定的路线传播。如光纤，同轴电缆，双绞铜线。

非导引型媒体中，信号在空气或太空中传播。如无线局域网。

1. **双绞铜线**

最便宜并且最常用的导引型传输媒体是双绞铜线。一百多年来，它一直用于电话网。事实上，从电话机到本地电话交换机的连线超过 99% 使用的是双绞铜线。我们多数人在自己家中和工作环境中已经看到过双绞线。双绞线由两根绝缘的铜线组成，每根大约 1 mm 粗，以规则的螺旋状排列着。这两根线被绞合起来，以减少邻近类似的双绞线的电气干扰。通常许多双绞线捆扎在一起形成一根电缆，并在这些双绞线外面覆盖上保护性防护层。一对电线构成了一个通信链路。**无屏蔽双绞线(Unshielded Twisted Pair, UTP)** 常用在建筑物内的计算机网络中，即用于局域网(LAN)中。目前局域网中的双绞线的数据速率从 10Mbps 到 10Gbps。所能达到的数据传输速率取决于线的粗细以及传输方和接收方之间的距离。

2. **同轴电缆**

与双绞线类似，同轴电缆由两个铜导体组成，但是这两个导体是同心的而不是并行的。借助于这种结构及特殊的绝缘体和保护层，同轴电缆能够达到较高的数据传输速率。同轴电缆在电缆电视系统中相当普遍。电缆电视系统最近与电缆调制解调器结合起来，为住宅用户提供数十 Mbp 速率的 Internet 接入。在电缆电视和电缆 Internet 接入中，发送设备将数字信号调制到某个特定的频段，产生的模拟信号从发送设备传送到一个或多个接收方。同轴电缆能被用作导引型共享媒体。特别是，许多端系统能够直接与该电缆相连，每个端系统都能接收由其他端系统发送的内容。

3. **光纤**

光纤是一种细而柔软的、能够导引光脉冲的媒体，每个脉冲表示一个比特。一根光纤能够支持极高的比特速率，高达数十甚至数百 Gbps。它们不受电磁干扰，长达 100km 的光缆信号衰减极低，并且很难窃听。这些特征使得光纤成为长途导引型传输媒体，特别是跨海链路。在美国和别的地方，许多长途电话

网络现在全面使用光纤。光纤也广泛用于 Internet 的主干。然而，高成本的光设备，如发射器、接收器和交换机，阻碍光纤在短途传输中的应用，如在 LAN 或家庭接入网中就不使用它们。

4. **陆地无线电信道**

无线电信道承载电磁频谱中的信号。它不需要安装物理线路，并具有穿透墙壁、提供与移动用户的连接以及长距离承载信号的能力，因而成为一种有吸引力的媒体。无线电信道的特性极大地依赖于传播环境和信号传输的距离。环境上的考虑取决于路径损耗和遮挡衰落（即当信号跨距离传播和绕过/通过阻碍物体时信号强度降低）、多径衰落（由于干扰对象的信号反射）以及干扰（由于其他传输或电磁信号）。

陆地无线电信道能够大致划分为三类：一类运行在很短距离（如 1 米或 2 米）；另一类运行在局域，通常跨越数十到几百米；第三类运行在广域，跨越数万米。个人设备如无线头戴式耳机、键盘和医疗设备跨短距离运行；无线 LAN 技术使用了局域无线电信道；蜂窝接入技术使用了广域无线电信道。我们将在第 7 章中详细讨论无线电信道。

5. **卫星无线电信道**

一颗通信卫星连接地球上的两个或多个微波发射器/接收器，它们被称为地面站。该卫星在一个频段上接收传输，使用一个转发器（下面讨论）再生信号，并在另一个频率上发射信号。通信中常使用两类卫星：同步卫星和近地轨道（LEO）卫星。

同步卫星永久地停留在地球上方的相同点上。这种静止性是通过将卫星置于地球表面上方 36 000km 的轨道上而取得的。从地面站到卫星再回到地面站的巨大距离引入了可观的 280ms 信号传播时延。不过，能以数百 Mbps 速率运行的卫星链路通常用于那些无法使用 DSL 或电缆 Internet 接入的区域。

近地轨道卫星放置得非常靠近地球，并且不是永久地停留在地球上方的一个点。它们围绕地球旋转，就像月亮围绕地球旋转那样，并且彼此之间可进行通信，也可以与地面站通信。为了提供对一个区域的连续覆盖，需要在轨道上放置许多卫星。当前有许多低轨道通信系统在研制中。LEO 卫星技术未来也许能够用于 Internet 接入。

1.1.3. 分组交换机

当一个端系统向另外一个端系统发送数据时，发送端系统将数据分段，每一个数据段被加上了首部字节。这种数据段被称为 **分组(packet)**。这些分组通过网络发送到目的端系统，在目的端系统被组装为初始数据。

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。在当今的 Internet 中，有 2 种典型的分组交换机：**路由器(router)** 和 **链路层交换机(link-layer switch)**。这两种类型的交换机朝着最终目的地转发分组。链路层交换机通常用于接入网，路由器通常用于网络核心。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称为该网络的 **路径(route)**。

用于传送分组的分组交换网络在许多方面类似于承载运输车辆的运输网络，该网络包括了高速公路、公路和交叉口。例如，考虑下列情况，一个工厂需要将大量货物搬运到数千公里以外的某个目的地仓库。在工厂中，货物要分开并装上卡车车队。然后，每辆卡车独立地通过高速公路、公路和立交桥组成的网络向仓库运送货物。在目的地仓库，卸下这些货物，并且与一起装载的同一批货物的其余部分堆放在一起。因此，在许多方面，分组类似于卡车，通信链路类似于高速公路和公路，分组交换机类似于交叉口，而端系统类似于建筑物。就像卡车选取运输网络的一条路径前行一样，分组则选取计算机网络的一条路径前行。

- **存储转发传输**

存储转发传输是指一个分组交换机在从输入链路接收到一个分组时，首先做的是接收该分组，然后再转发给出链路。这样造成的时延称为 **存储转发时延**。

- **排队时延和分组丢失**

每台分组交换机具有一个 **输出缓存** 。与一个分组交换机相连的出链路有多条。如果到达的分组需要传输到某条链路，却发现该链路正忙于传输其他分组。那么该分组必须在输出缓存中等待。因此除了存储转发时延外分组还要承受 **排队时延**。如果输出缓存完全充满，那么一个到达的分组就会 **丢失**。

- **转发表和路由转发协议**

前面说过一个分组需要分组交换机转发到一条出通信链路上，那么分组交换机是怎么决定转发到哪一条出链路呢?

在 Internet 中，每一个端系统都有一个称为 IP 地址的地址。当源主机向目的端发送一个分组时，源在该分组的首部包含了目的端的 IP 地址。该地址具有一个等级结构。分组交换机会检查目的端 IP 地址的一部分，并向相邻的一台分组交换机转发该分组。具体来说，每台分组交换机会有一个转发表，用于将 IP 地址映射为出链路。

那么转发表是怎么设置的呢？其实这是由 **路由转发协议** 生成的。

- **电路交换**

通过通信链路和交换机转发数据的方式有 2 种：**电路交换** 和 **分组交换**。前面已经讨论过分组交换，现在讨论一下电路交换。

在电路交换网络中，在端系统会话期间，预留了端系统间沿路径通信所需要的资源，如：缓存和链路传输速度。在分组交换网络中，这些资源是不预留的。会话的报文按需使用这些资源，其后果是不得不等待接入通信链路。

传统的电话网络是电路交换网络的一个例子。在发送方能够发送信息之前，该网络必须在发送方和接收方之间建立一条连接。这是一个名副其实的连接，因为此时沿着发送方和接收方之间 路径上的交换机都将为该连接维护连接状态。用电话的术语来说，该连接被称为一条电路。当网络创建这种电路时，它也

在连接期间在该网络链路上预留了恒定的传输速率（表示为每条链路传输容量的一部分）。既然已经为该发送方-接收方连接预留了带宽,则发送方能够以确保的恒定速率向接收方传送数据。

图 1-13 实现了一个电路交换网络。在这个网络中，用 4 条链路互联了 4 台电路交换机。这些链路中每条都有 4 条电路，因此每条链路支持 4 个并行的连接。每台主机都与一台交换机直接相连。当两台主机要通信时，该网络在两台主机之间创建一条专用的**端到端连接**。

因此，主机 A 为了向主机 B 发送报文，网络必须在两方链路的每条上先预留一条链路。在这个例子中，这条专用的端到端连接使用用第一条链路中的第二条电路和第二条链路中的第四条电路。

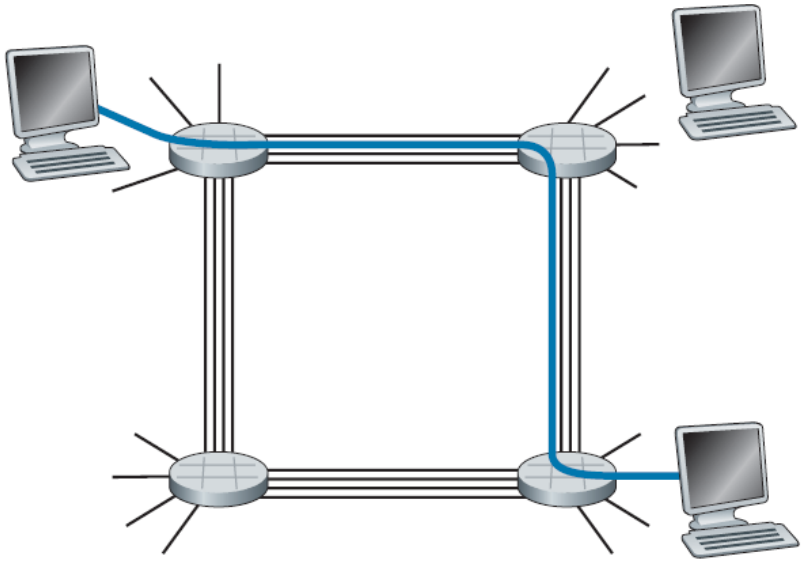


Figure 1.13 ♦ A simple circuit-switched network consisting of four switches and four links

与此相反，考虑一台主机要经过分组交换网络（如 Internet）向另一台主机发送分组所发生的情况。与使用电路交换相同，该分组经过一系列通信链路传输。但与电路交换不同的是，该分组被发送进网络，而不预留任何链路资源之类的东西。如果因为此时其他分组也需要经该链路进行传输而使链路之一出现拥塞，则该分组将不得不在传输链路发送侧的缓存中等待而产生时延。Internet 尽最大努力以实时方式交付分组，但它不做任何保证。

- **电路交换网络中的复用**

电路交换网络中的复用有 2 类：**频分复用** 或 **时分复用**。

- **分组交换和电路交换的对比**

分组交换的性能能够优于电路交换的性能。电路交换不考虑需求，而预先分配了传输链路的使用，这使得已分配而并不需要的链路时间未被利用。另一方面，分组交换按需分配链路使用。链路传输能力将在所有需要在链路上传输分组的用户之间逐分组地被共享。

虽然分组交换和电路交换在今天的电信网络中都是普遍采用的方式，但趋势无疑是朝着分组交换方向发展。

1.1.4. 网络协议

端系统和分组交换机以及其他 Internet 部件都运行着一系列 **协议(protocol)**。这些协议控制 Internet 中信息的发送和接收。**TCP(Transmiss1on Control Protocol, 传输控制协议)** 和 **IP(Internet Protocol, 网际协议)** 是 Internet 协议中最重要的 2 个协议。

下面我们围绕这 2 个问题，深入介绍网络协议。

什么是协议？协议可以做什么?

1. **人类活动的类别**

也许理解计算机网络协议这一概念的一个最容易办法是，先与某些人类活动进行类比，因为我们人类无时无刻不在执行协议。考虑当你想要向某人询问时间时将要怎样做。 图 1-2 中显示了一种典型的交互过程。人类协议(至少是好的行为方式)要求一方首先进行问候(图 1-2 中的第一个“你好”)，以开始与另一个人的通信。对“你好”的典型响应是返回一个“你好”报文。此人用一个热情的“你好”进行响应，隐含着一种指示，表明能够继续向那人询问时间了。对最初的“你好”的不同响应(例如“不要烦我!”，或“我不会说英语”，或某些不合时宜的回答)也许表明了一个勉强的或不能进行的通信。在此情况下，按照人类协议，发话者就不能询问时间了。有时，问的问题根本得不到任何回答，在此情况下，发话者通常会放弃向这个人询问时间。注意在我们人类协议中, 有我们发送的特定报文，也有我们根据接收到的应答报文或其他事件(例如在某个给定的时间内没有回答)采取的动作。显然，发送和接收的报文，以及这些报文发送和接收或其他事件出现时所采取的动作，这些在一个人类协议中起到了核心作用。如果人们使用不同的协议(例如，如果一个人讲礼貌，而另一人不讲礼貌，或一个人明白时间的概念，而另一人却不理解)，这些协议就不能交互，因而不能完成有用的工作。在网络中这个道理同样成立。即为了完成一项工作，要求两个(或多个)通信实体运行相同的协议。

我们再考虑第二个人类类比的例子。假定你正在大学课堂里上课(例如上的是计算机网络课程)。教师正在唠唠叨叨地讲述协议，而你迷惑不解。这名教师停下来问: “同学们有什么问题吗？”(教师发送出一个报文，该报文被所有没有睡觉的学生接收到了。)你举起了手(向教师发送了一个隐含的报文)。这位教师面带微笑地示意你说: “请讲……”(教师发出的这个报文鼓励你提出问题，教师喜欢被问问题。)接着你就问了问题(向该教师传输了你的报文)。教师听取了你的问题(接收了你的问题报文)并加以回答(向你传输了回答报文)。我们再一次看到了报文的发送和接收，以及这些报文发送和接收时所采取的一系列约定俗成的动作，这些是这个“提问与回答”协议的关键所在。

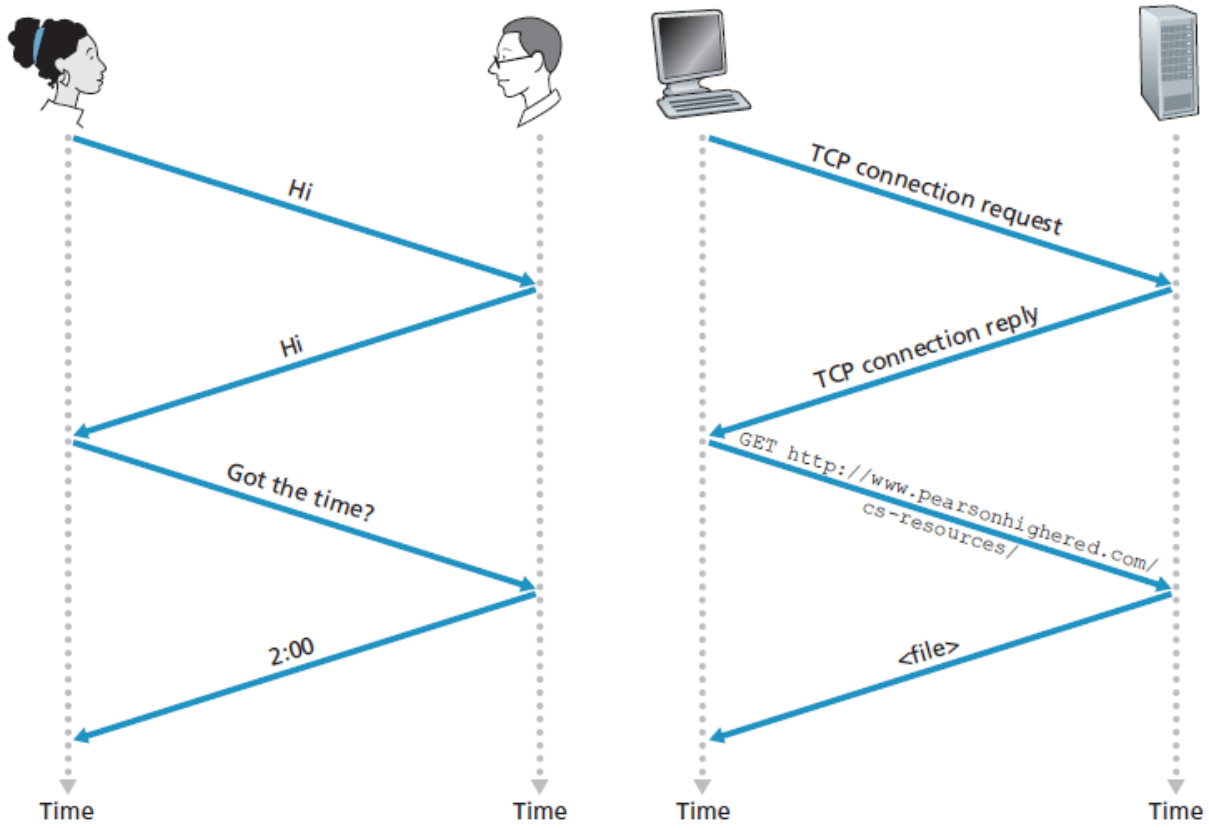


Figure 1.2 ♦ A human protocol and a computer network protocol

2. **网络协议**

网络协议类似于人类协议，除了交换报文和采取动作的实体是某些设备(可以是计算机、智能手机、平板电脑、路由器或其他具有网络能力的设备)的硬件或软件组件。在 Internet 中，涉及两个或多个远程通信实体的所有活动都受协议的制约。例如，在两台物理上 连接的计算机中，硬件实现的协议控制了在两块网络接口卡间的“线上”的比特流；在端系统中，拥塞控制协议控制了在发送方和接收方之间传输的分组发送的速率；路由器中的协议决定了分组从源到目的地的路径。在 Internet 中协议运行无处不在，因此本书的大量篇幅都与计算机网络协议有关。

以大家可能熟悉的一个计算机网络协议为例，考虑当你向一个 Web 服务器发出请求(即你在 Web 浏览器中键入一个 Web 网页的 URL)时所发生的情况。图 1-2 右半部分显示了这种情形。首先，你的计算机将向该 Web 服务器发送一条连接请求报文，并等待回答。 该 Web 服务器将最终能接收到连接请求报文，并返回一条连接响应报文。得知请求该 Web 文档正常以后，计算机则在一条 GET 报文中发送要从这台 Web 服务器上取回的网页名字。最后，Web 服务器向计算机返回该 Web 网页(文件)。

从上述的人类活动和网络例子中可见，报文的交换以及发送和接收这些报文时所采取的动作是定义一个协议的关键元素:

协议定义了在两个或多个通信实体之间交换报文的格式和顺序，发送报文和接收报文以及其他事件所采取的动作。

Internet(更一般地说是计算机网络)广泛地使用了协议。不同的协议用于完成不同的通信任务。当你阅读完这本书后将会知道，某些协议简单而直截了当，而某些协议则复杂且晦涩难懂。掌握计算机网络领域知识的过程就是理解网络协议的构成、原理和工作方式的过程。

1.1.5. Internet 标准

Internet 标准(Internet standard) 由 Internet 工程任务组(Internet Engineering Task Forse 或 IETF)。IETF 的标准文档称为请求评论(Request For Comment 或 RFC)。RFC 最初只是普通的请求评论(因此得名)，目的是解决 Internet 先驱者们面临的网络和协议问题。RFC 往往是技术性很强并相当详细的。它们定义了 TCP，IP，HTTP 和 SMTP 等协议。其他组织也在制定用于网络部件的标准，最引人注目的是针对网络链路的标准。例如 IEEE 802 LAN/MAN 标准委员会 [IEEE 802 2016] 制定了以太网和 WiFi 的标准。

1.2. Internet 提供的服务

1.2.1. 分布式应用程序

端系统上运行着应用程序，因此端系统也称为主机。主机上的大部分应用程序涉及在多个主机之间交换信息。因此这类应用程序被称为 **分布式应用程序**。

端系统上运行的分布式应用程序之间有着组织方式，我们称这种组织方式为分布式应用程序的体系结构。现有 2 种流行的体系结构：**客户-服务器体系结构(client-server architecture)** 和 **对等体系结构(P2P architecture)**。

- **客户-服务器体系结构**

在这种体系结构中，有一个总是打开的主机，称为**服务器**，它服务于许多个其他主机的请求，这些主机称为**客户**。

举个例子。常见的 Web 应用程序就是客户-服务器体系结构。其中，服务器为 Web 服务器，客户为运行在客户机上的浏览器。当一个客户发起请求时，Web 服务器向它响应。

这种体系结构具有以下特点：

1. 客户之间不直接通信
2. 服务器具有公开的，固定的地址，该地址称为 IP 地址

- **对等体系结构**

在这种体系结构中，主机几乎对专用服务器没有依赖，端系统之间直接通信。

常见的应用有：BitTorrent，迅雷，Skype，QQ。

1.2.2. 套接字

与 Internet 相连的端系统提供了一个 **套接字接口(socket interface)**，该接口规定了运行在一个端系统上的程序请求 Internet 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。Internet 套接字接口是一套发送程序必须遵循的规则集合，因此 Internet 能够将数据交付给目的地。我们将在第 2 章详细讨论 Internet 套接字接口。此时，我们做一个简单的类比，在本书中我们将经常使用这个类比。假定 Alice 使用邮政服务向 Bob 发一封信。 当然，Alice 不能只是写了这封信(相关数据)然后把该信丢出窗外。相反，邮政服务要求 Alice 将信放入一个信封中；在信封的中间写上 Bob 的全名、地址和邮政编码；封上信封；在信封的右上角贴上邮票；最后将该信封丢进一个邮局的邮政服务信箱中。因此，该邮政服务有自己的“邮政服务接口”或一套规则，这是 Alice 必须遵循的，这样邮政服务才能将她的信件交付给 Bob。同理，Internet 也有一个发送数据的程序必须遵循的套接字接口，使 Internet 向接收数据的程序交付数据。

当然，邮政服务向顾客提供了多种服务，如特快专递、挂号、普通服务等。同样，Internet 向应用程序提供了多种服务。当你研发一种 Internet 应用程序时，也必须为你的应用程序选择其中的一种 Internet 服务。我们将在第 2 章中描述 Internet 服务。

我们已经给出了 Internet 的两种描述方法：一种是根据它的硬件和软件成来描述，另一种是根据基础设施向分布式应用程序提供的服务来描述。但是，你也许还是对什么是 Internet 感到困惑，请不要担心。这本书除了向你介绍 Internet 的具体构成外，还要介绍支配 Internet 的工作原理以及它工作的来龙去脉。我们将在后续章节中解释这些重要的术语和问题。

1.3. Internet 的结构

1.3.1. 接入 ISP 网络

接入 ISP 网络是指将端系统接入 Internet 核心的 ISP 网络。

端系统通过 **Internet 服务提供商(Internet Service Provider, ISP)** 接入 Internet。ISP 包括如本地电缆或电话公司那样的住宅区 ISP、公司 ISP、大学 ISP，在机场、旅馆、咖啡店和其他公共场所提供 WiFi 接入的 ISP，以及为智能手机和其他设备提供移动接入的蜂窝数据 ISP(基站)。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。各 ISP 为端系统提供了各种不同类型的网络接入，包括如线缆调制解调器或 DSL 那样的住宅宽带接入，高速局域网接入和移动无线接入。ISP 也为内容提供者提供 Internet 接入服务，将 Web 站点和视频服务器直接连入 Internet。

Internet 要将端系统彼此互联，因此为端系统提供接入的 ISP 也必须互联。较低层的 ISP 通过国家的或国际的较高层 ISP 互联起来。较高层 ISP 是由通过高速光纤链路互联的高速路由器组成的。我们将在 1-2-2 节深入地讨论 ISP 的互联结构。

图 1-4 用粗的、带阴影的线高亮显示了几种类型接入 ISP 网络。

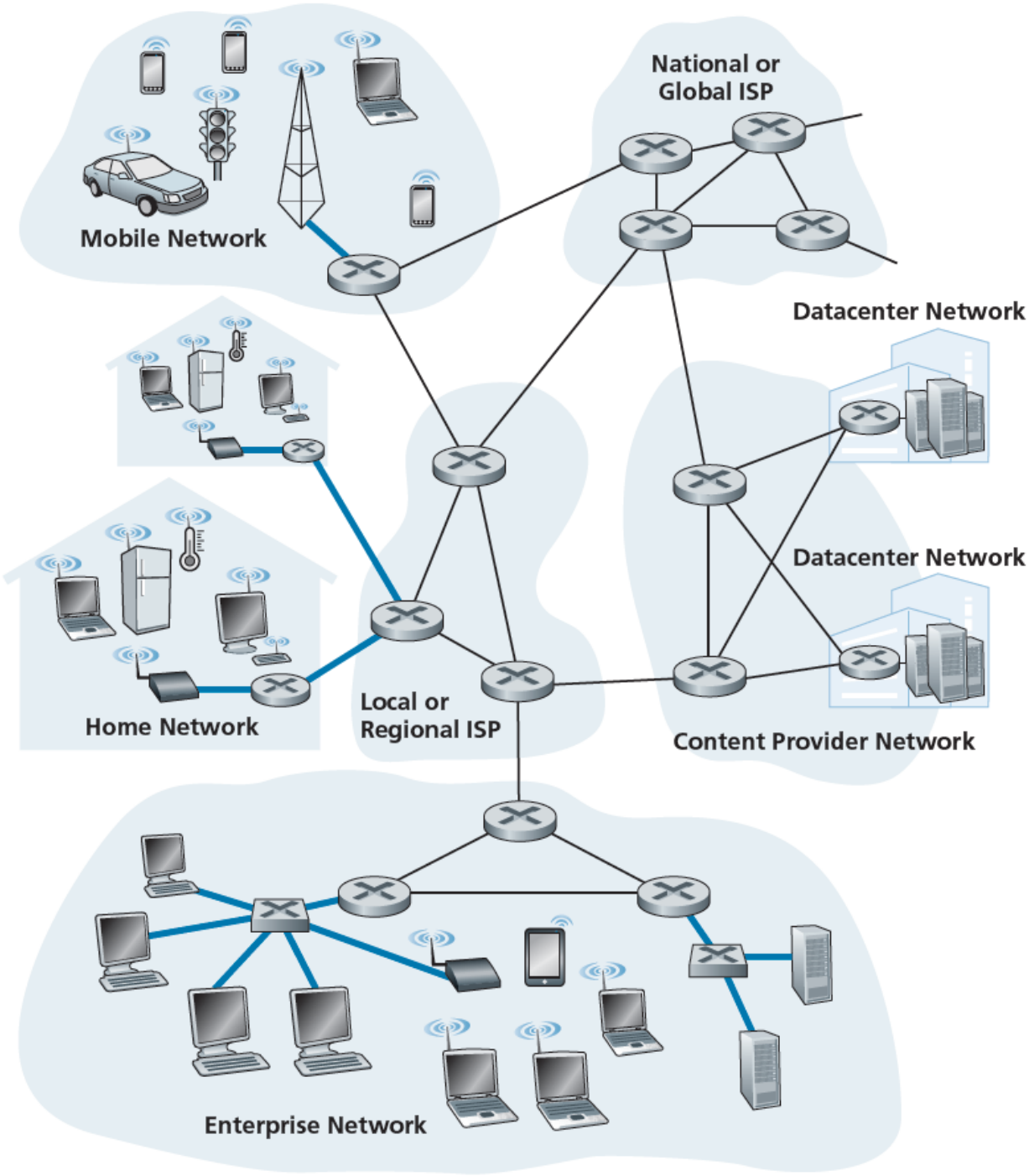


Figure 1.4 ♦ Access networks

1. **家庭接入：DSL 接入，电缆接入，光纤接入，拨号和卫星接入**

- **DSL 接入**

住户通常从本地的电话公司处获得 DSL Internet 接入。这种情况下，ISP 就是用户的本地电话公司。

如图 1-5 所示，每个用户的 DSL 调制解调器使用现有的电话线（即双绞铜线，将在 1.2.2 节中讨论它）与位于电话公司的本地中心局（CO）中的数字用户线接入复用器（DSLAM）交换数据。家庭的 DSL 调制解 调器得到数字数据后将其转换高频音，以通过电话线传输给本地中心局；来自许多家庭的模拟信号在 DSLAM 处被转换回数字形式。

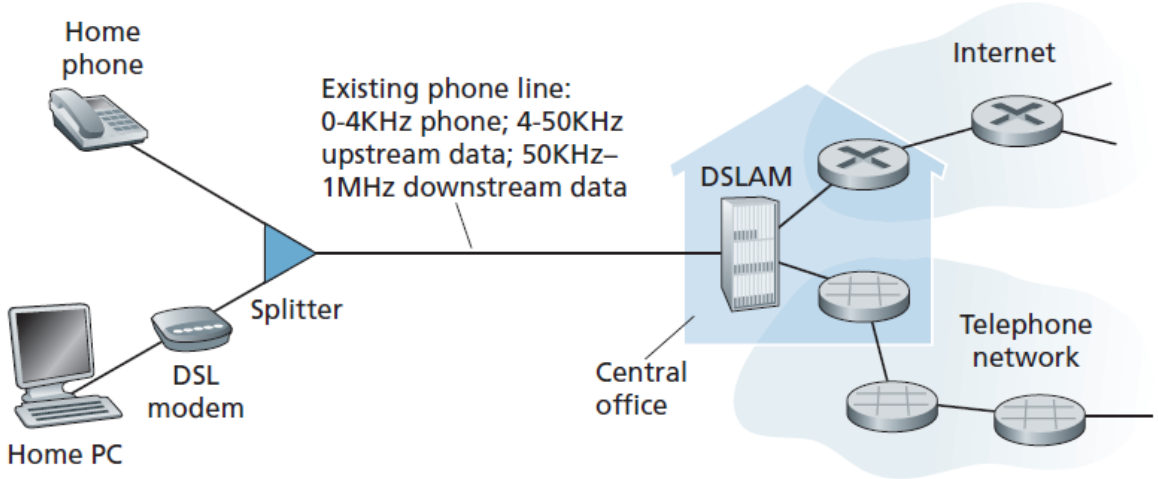


Figure 1.5 ♦ DSL Internet access

DSL 标准定义了多个传输速率，包括 12 Mbps 下行和 1.8Mbps 上行以及 55mbps 下行和 15Mbps 上行。

因为这些上行速率和下行速率是不同的，所以这种接入被称为是不对称的。

- **电缆接入**

住宅从提供有线电视的公司获得了电缆 Internet 接入。这种情况下，ISP 就是用户的有线电视公司。

如图 1-6 所示，光缆将电缆头端连接到地区枢纽，从这里使用传统的同轴电缆到达各家各户和公寓。每个地区枢纽通常支持 500 -5000 个家庭。因为在这个系统中应用了光纤和同轴电缆，所它经常被称为混合光纤同轴(HFC)系统。

电缆 Internet 接入需要特殊的调制解调器，这种调制解调器称为电缆调制解调器，如同 DSL 调制解调器，电缆调制解调器通常是一个外部设备，通过一个以太网端口连接到家庭 PC。在电缆头端，电缆调制解调器端接系统与 DSL 网络的 DSLAM 具有类似的功能，即将来自许多下行家庭中的电缆调制解调器发送的模拟信号转换回数字形式。

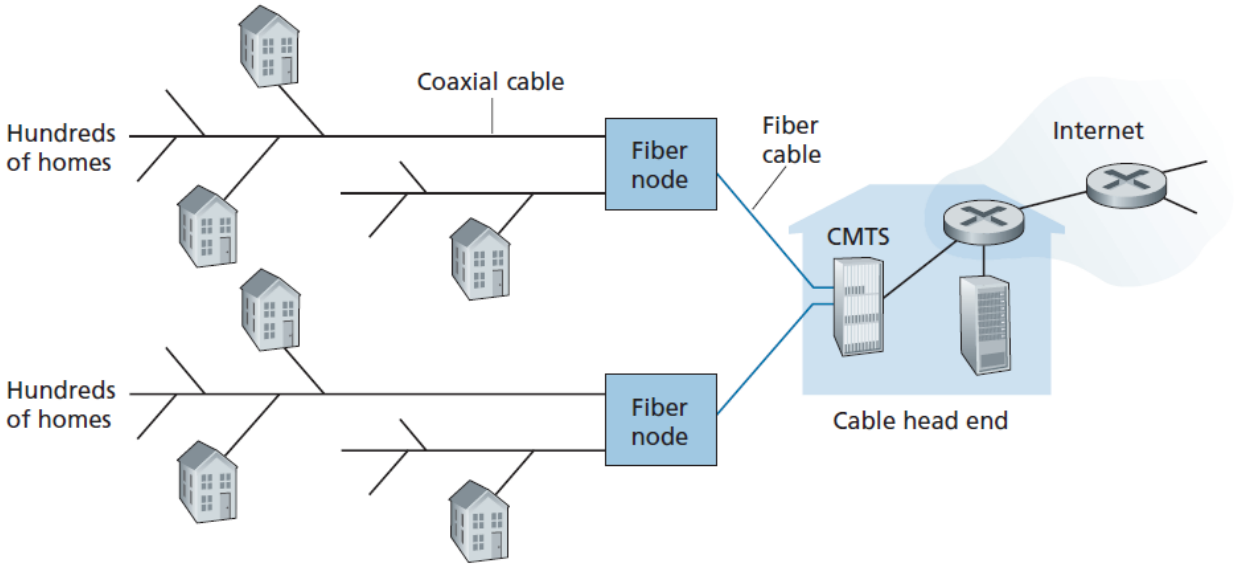


Figure 1.6 ♦ A hybrid fiber-coaxial access network

电缆调制解调器将 HFC 网络划分为下行和上行两个信道。如同 DSL,接入通常是不对称的，下行信道分配的传输速率通常比上行信道的高。DOCS1S 2.0 标准定义了高达 42.8Mbps 的下行速率和高达 30.7Mbps 的上行速率。

- **光纤接入**

光纤接入就是从本地中心局直接到家庭提供一个光纤路径。从本地中心局到家庭有几种有竞争的光纤布局方案。一种是直接光纤，从本地中心局到每个用户直接设置一根光纤。还有一种较为一般，从中心局出来一根光纤，到临近家庭的位置，才分给每个用户一个光纤。这种方案有两种类型：主动光纤网络(AOT)和被动光纤网络(PON)。

这里简要介绍 PON。如图 1-7 所示，每个家庭具有一个光纤网络端接器(ONT),它由专门的光纤连接到邻近的分配器(splitter)，该分配器把一些家庭(通常少于 100 个)集结到一根共享的光纤，该光纤再连接到本地电话和公司的中心局中的光纤线路端接器(OLT)，该 OLT 提供了光信号和电信号之间的转换，经过本地电话公司路由器与 Internet 相连。在家庭中，用户将一台家庭路由器(通常是无线路由器)与 ONT 相连，并经过这台家庭路由器接入 Internet。在 PON 体系结构中，所有从 OLT 发送到分配器的分组在分配器(类似于一个电缆头端)处复制。

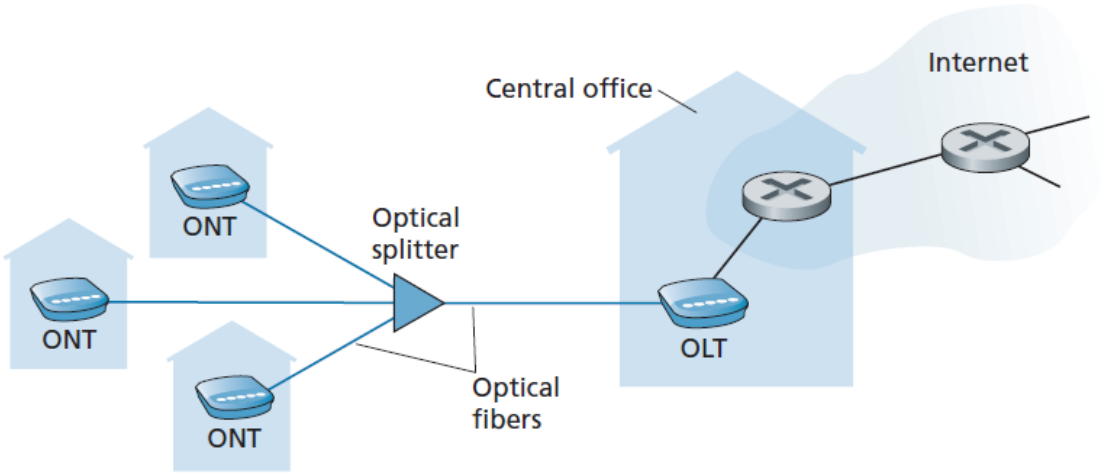


Figure 1.7 ♦ FTTH Internet access

- **拨号和卫星接入**

还可采用另外两种接入网技术为家庭提供 Internet 接入。在无法提供 DSL、电缆和 FTTH 的地方（例如在某些乡村环境），能够使用卫星链路将住宅以超过 1Mbps 的速率与 Internet 相连。StarBand 和 HughesNet 是两家这样的卫星接入提供商。使用传统电话线的拨号接入与 DSL 基于相同的模式：家庭的调制解调器经过电话线连接到 ISP 的调制解调器。与 DSL 和其他宽带接入网相比，拨号接入 56kbps 的慢速率是令人痛苦的。

2. **机构接入：以太网和 WiFi**

- **以太网**

在公司和大学校园以及越来越多的家庭环境中，使用 **局域网(LAN)** 将端系统连接到边缘路由器。尽管有许多不同类型的局域网技术，但是 **以太网** 到目前为止是公司、大学和家庭网络中最为流行的接入技术。

如图 1-8 中所示，以太网用户使用双绞铜线与一台以太网交换机相连，第 6 章中将详细讨论该技术。以太网交换机或这样相连的交换机网络，则再与更大的 Internet 相连。

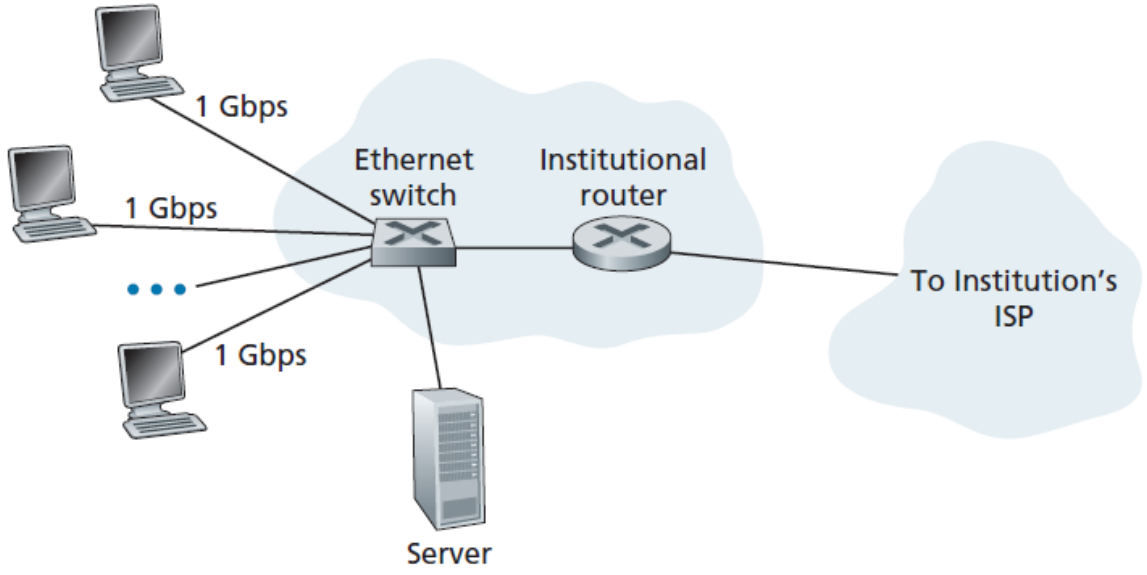


Figure 1.8 ♦ Ethernet Internet access

使用以太网接入，用户通常以 100Mbps 或 1Gbps 速率接入以太网交换机，而服务器可能具有 1Gbps 甚至 10Gbps 的接入速率。

- **WiFi(无线 LAN)**

如今，越来越多的人通过移动 PC，智能手机，平板电脑和其他设备接入 Internet。在无线 LAN 环境中，无线用户从/到一个接入点发送/接收分组，该接入点与企业网连接（很可能使用了有线以太网），企业网再与有线 Internet 相连。

一个无线 LAN 用户通常必须位于接入点的几十米范围内。基于 IEEE 802.11 技术的无线 LAN 接入,更通俗地称为 WiFi,目前几乎无所不在，如大学、商业办公室、咖啡厅、 机场、家庭，甚至在飞机上。

IEEE 802.11 今天提供了高达 100Mbps 的共享传输速率。

虽然以太网和 WiFi 接入网最初是设置在企业（公司或大学）环境中的，但它们近来已经成为家庭网络中相当常见的部件。今天许多家庭将宽带住宅接入（即电缆调制解调器 或 DSL）与廉价的无线局域网技术结合起来，以产生强大的家用网络。图 1-9 显示了典型的家庭网络。这个家庭网络组成如下：一台漫游的便携机和一台有线 PC； — 个与无线 PC 和家中其他无线设备通信的基站（无线接入点）；一个提供与 Internet 宽带接入的 电缆调制解调器；一台互联了基站及带有电缆调制解调器的固定 PC 的路由器。该网络允许 家庭成员经宽带接入 Internet，其中任何一个家庭成员都可以在厨房、院子或卧室漫游上网。

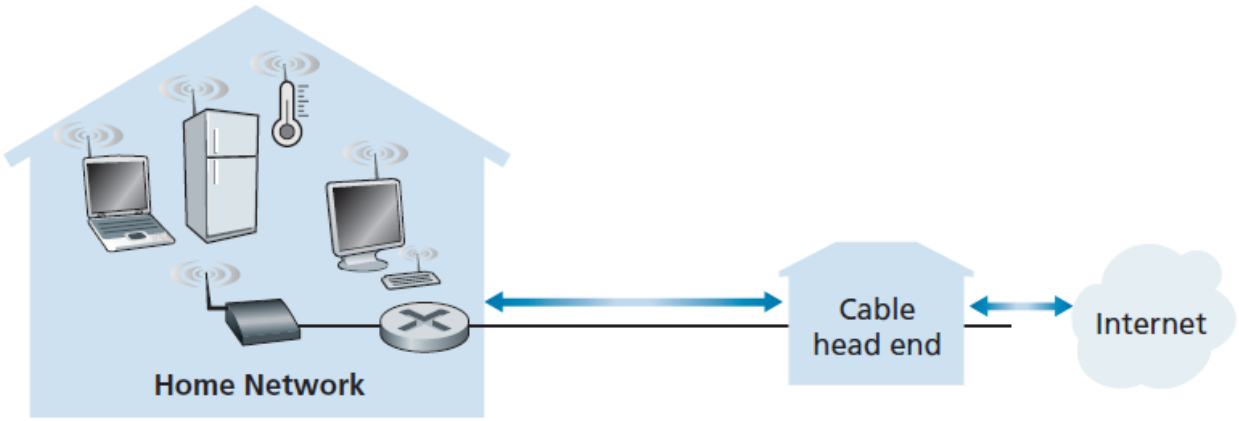


Figure 1.9 ♦ A typical home network

3. **广域无线接入：4G 和 5G**

我们可以使用 iPhone 和安卓等移动设备发信息、在社交网络中分享照片、观看视频和放音乐。这些设备应用了与蜂窝移动电话相同的无线基础设施，通过蜂窝网提供商运营的基站来发送和接收分组。与 WiFi 不同的是，一个用户仅需要位于基站的数万米(而不是几十米)范围内。

电信公司在第 4 代移动通信技术(4G)上做了巨大的投资。4G 移动通信技术可以提供 60Mbps 的下载速度。但是更高速度的广域无线接入技术，第五移动通信技术(5G)已经投入了部署。我们将在第 7 章详细地讨论 WiFi，4G，5G 等技术。

1.3.2. Internet 核心

上一节谈到，接入 ISP 网络将端系统接入 Internet 核心。

Internet 核心是指互联了接入 ISP 网络的核心网络。Internet 核心本身由更高层次的 ISP 网络，内容提供商网络，以及其他一些部件互联形成。

图 1-10 以加粗和带阴影的线高亮显示了 Internet 核心。

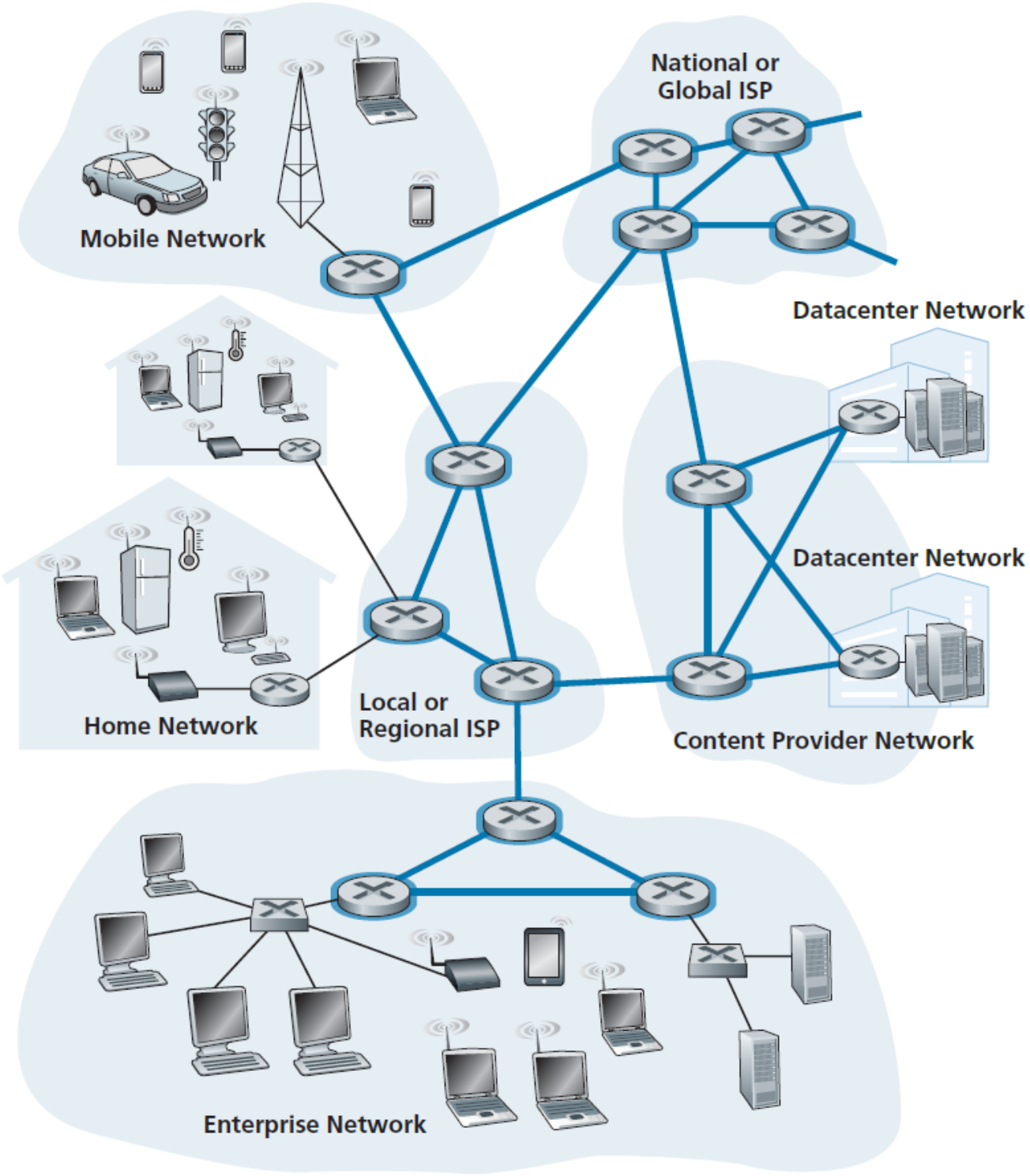


Figure 1.10 ♦ The network core

可是各层次 ISP 网络到底是怎么样互联的呢?

为了理解今天的 ISP 互联结构，我们将逐步地构建一系列 ISP 互联结构，每一个结构都更接近如今的 Internet。

回顾前面的互联接入 ISP 的目标：使所有的端系统能够交换分组。

最简单的方法就是将每一个 ISP 直接与其他的 ISP 相连。当然这是不可行的，这样的设计费用太高，因为这种设计要求每一个 ISP 要与世界上数十万个其他接入 ISP 有一条单独的通信链路。

我们的第一个 ISP 互联结构，用单一的全球传输 ISP 互联所有的 ISP，我们假想的全球传输 ISP 是一个由路由器和通信链路构成的网络，该网络跨越全球，而且至少有一个路由器靠近数十万个接入 ISP 的每一个。为了有利可图，自然要向每个连接的接入 ISP 收费，其价格反映（并不一定正比于）一个接入 ISP 经过全球 ISP 交换的流量大小。因为接入 ISP 向全球传输 ISP 付费，故接入 ISP 被认为是 **客户(customer)**，而全球传输 ISP 被认为是 **提供商(provider)**。

如果某个公司建立并运营一个可赢利的全球传输 ISP，其他公司建立自己的全球传输 ISP 并与最初的全球传输 ISP 竞争则是一件自然的事。这导致了 ISP 互联结构 2。

ISP 互联结构 2 由数十万接入 ISP 和多个全球传输 ISP 组成。接入 ISP 无疑更喜欢 ISP 互联结构 2,因为 它们现在能够根据价格和服务因素在多个竞争的全球传输提供商之间进行选择。然而，值得注意的是，这些全球传输 ISP 之间必须是互联的；不然的话，与某个全球传输 ISP 连接的接入 ISP 将不能与连接到其他全球传输 ISP 的接入 ISP 进行通信。

刚才描述的网络结构 2 是种两层的等级结构，其中全球传输提供商位于顶层，而接入 ISP 位于底层。这假设了全球传输 ISP 不仅能够接近每个接入 ISP,而且费用上也是可行的。现实中，尽管某些 ISP 确实具有令人印象深刻的全球覆盖，并且确实直接与许多接入 ISP 连接，但世界上没有哪个 ISP 是无处不在的。相反，在任何给定的区域，可能有一个 **区域 ISP(regional ISP)**，区域中的接入 ISP 与之连接。每个区域 ISP 则与 **第一层 ISP(tier-1 ISP)** 连接。第一层 ISP 类似于我们假想的全球传输 ISP，尽管它不是在世界上每个城市中都存在，但它确实存在。有大约十几个第一层 ISP，包括 Level 3 Communications，AT&T，Sprint 和 NTT。

再来讨论这个网络的网络，不仅有多个竞争的第一层 ISP,而且在一个区域可能有多 个竞争的区域 ISP。在这样的等级结构中，每个接入 ISP 向其连接的区域 ISP 支付费用，并且每个区域 ISP 向它连接的第一层 ISP 支付费用。(一个接入 ISP 也能直接与第一层 ISP 连接，这样它就向第一层 ISP 付费。)因此，在这个等级结构的每一层，都有客户-提供 商关系。值得注意的是，第一层 ISP 不向任何人付费，因为它们位于该等级结构的顶部。 更为复杂的情况是，在某些区域，可能有较大的区域 ISP (可能跨越整个国家)，该区域 中较小的区域 ISP 与之相连，较大的区域 ISP 则与第一层 ISP 连接。例如，在中国，每个城市有接入 ISP,它们与省级 ISP 连接，省级 ISP 又与国家级 ISP 连接，国家级 ISP 最终与第一层 ISP 连接［Tian 2012］。这个多层等级结构仍然仅仅是今天因特网的粗略近似，我们称它为 ISP 互联结构 3。

为了建造一个与今天的因特网更为相似的网络，我们必须在等级化网络结构 3 上增加 **存在点(Point of Presence, PoP)** 、多宿、对等和因特网交换点。PoP 存在于等级结构的所 有层次，但底层(接入 ISP)等级除外。一个 POP 只是提供商网络中的一台或多台路由器 (在相同位置)群组，其中客户 ISP 能够与提供商 ISP 连接。对于要与提供商 PoP 连接的 客户网络，它能从第三方电信提供商租用高速链路将它

的路由器之一直接连接到位于该 PoP 的一台路由器。任何 ISP (除了第一层 ISP)可以选择 **多宿(multi-home)**，即可以与 两个或更多提供商 ISP 连接。例如，一个接入 ISP 可能与两个区域 ISP 多宿，既可以与两个区域 ISP 多宿，也可以与一个第一层 ISP 多宿。当一个 ISP 多宿时，即使它的提供商之一出现故障，它仍然能够继续发送和接收分组。

正如我们刚才学习的，客户 ISP 向它们的提供商 ISP 付费以获得全球因特网互联能力。客户 ISP 支付给提供商 ISP 的费用数额反映了它通过提供商交换的通信流量。为了减少这些费用，位于相同等级结构层次的邻近一对 ISP 能够 **对等(peer)**，也就是说，能够直接将它们的网络连到一起，使它们之间的所有流量经直接连接而不是通过上游的中间 ISP 传输。当两个 ISP 对等时，通常不进行结算，即任一个 ISP 不向其对等付费。如前面提到的那样，第一层 ISP 也与另一个第一层 ISP 对等，它们之间无结算。对于对等和客户-提供商关系的讨论，［Van der Berg 2008］是一本不错的读物。沿着这些相同路线，第三方公司能够创建一个 **因特网交换点(Internet Exchange Point, IXP)**, IXP 是一个汇合点，多个 ISP 能够在这里一起对等。IXP 通常位于一个有自己的交换机的独立建筑物中［Ager 2012］, 在今天的因特网中有 600 多个 IXP ［IXP List 2016 ］。我们称这个生态系统为 ISP 互联结构 4：由接入 ISP、区域 ISP、第一层 ISP、PoP、多宿、对等和 IXP 组成。

我们现在最终到达了 ISP 互联结构 5，它描述了现今的因特网。在图 1-15 中显示了 ISP 互联结构 5，它通过在 ISP 互联结构 4 顶部增加 **内容提供商网络(content provider network)** 构建而成。谷歌是当前这样的内容提供商网络的一个突出例子。在本书写作之时，谷歌估计有 19 个主要的数据中心分布于北美、欧洲、亚洲、南美和澳大利亚。其中每一个数据中心都有数万到数十万的服务器。此外谷歌也有较小的数据中心，每一个有几百个服务器，这些小型数据中心常常位于 IXP 内。谷歌数据中心都 经过专用的 TCP/IP 网络互联，该网络跨越全球，不过独立于公共因特网。重要的是，谷歌专用网络仅承载出入谷歌服务器的流量。如图 1-15 所示，谷歌专用网络通过与较低层 ISP 对等（无结算），尝试“绕过”因特网的较高层，采用的方式可以是直接与它们连接，或者在 IXP 处与它们连接［Labovitz2010］。然而，因为许多接入 ISP 仍然仅能通过第一层网络的传输到达，所以谷歌网络也外第一层 ISP 连接，并就与这些 ISP 交换的流量向它们付费。通过创建自己的网络，内容提供商不仅减少了向顶层 ISP 支付的费用，而且对其服务最终如何交付给端用户有了更多的控制。谷歌的网络基础设施在 2-6 节中进行了详细描述。

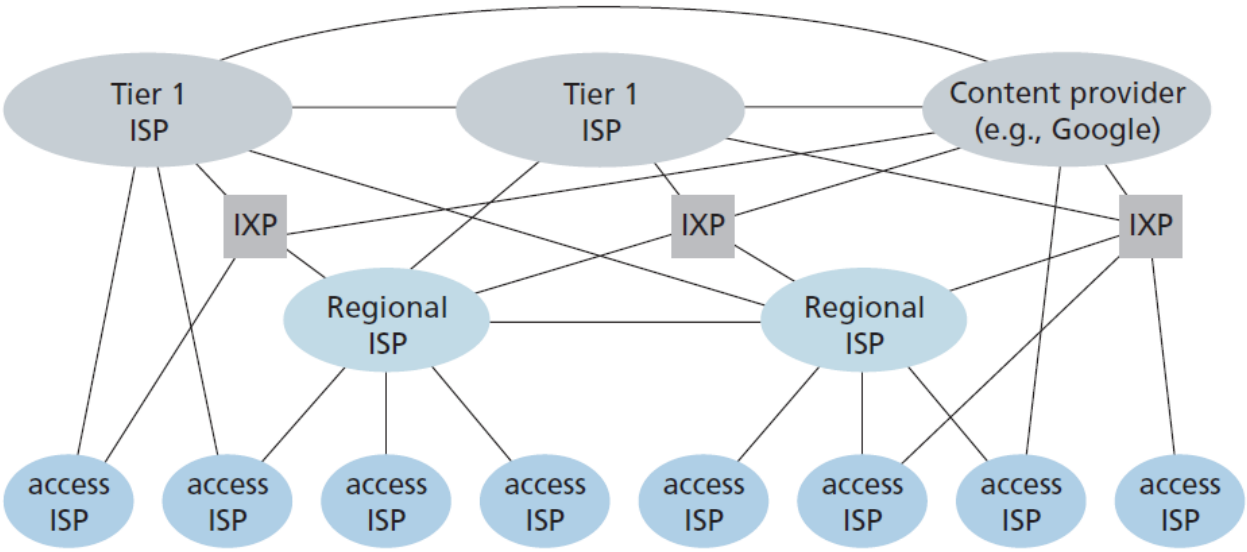


Figure 1.15 ♦ Interconnection of ISPs

1.4. Internet 分层模型

1.4.1. 分层的体系结构

前几节的讨论中，我们了解到 Internet 是个非常复杂的结构。各种分布式应用程序运行在端系统上，各种的网络协议控制着不同的分组发送规则，各种物理媒体的通信链路，等等。

为了使维护，管理，使用 Internet 更加方便，必须有一种良好的组织 Internet 的方式。这个方式就是 Internet 分层模型。

在 Internet 分层模型中，下一层向上一层提供 **服务(service)**，而上一层不必关注下一层的实现细节。当下一层的实现方式更换后，也不影响上一层的运作。不同的层负责不同的职能。

Internet 分层模型具有概念化和结构化的优点［RFC 3439］。如我们看到的那样，分层提供了一种结构化方式来讨论系统组件。模块化使更新系统组件更为容易。然而，需要提及的是，某些研究人员和联网工程师激烈地反对分层［WMenian 1992］。分层的一个潜在缺点是一层可能冗余较低层的功能。例如，许多协议栈在基于每段链路和基于端到端两种情况下，都提供了差错恢复。第二种潜在的缺点是某层的功能可能需要仅在其他某层才出现的信息（如时间戳值），这违反了层次分离的目标。

一个协议层能够用软件或硬件或两者的结合来实现。应用层和运输层几乎使用软件来实现。物理层和数据链路层通常在网络接口卡（例如以太网 或 WiFi 接口卡）中实现。网络层用软硬件结合的方式来实现。

所有各层的协议称为 **协议栈(protocol stack)**。

Internet 的协议栈由 5 个层组成：**物理层**、**链路层**、**网络层**、**运输层**和**应用层**。如图 1-23 所示。

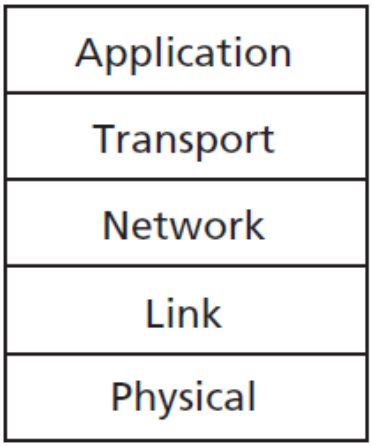


Figure 1.23 ♦ The Internet protocol stack

本书的结构采用了 **自顶向下方法(top-down approach)**，先处理应用层，然后向下处理。

- **应用层**

应用层由运行在端系统上的分布式应用程序及它们的应用层协议组成。Internet 的应用层包括许多协议，例如 HTTP （它提供了 Web 文档的请求和传送）、SMTP （它提供了电子邮件报文的传输）和 FTP （它提供两个端系统之间的文件传送）。我们将看到，某些网络功能，如将像 www.ietf.org 这样对人友好的端系统名字转换为 32 比特的网络地址，也是借助于特定的应用层协议即域名系统（DNS）完成的。我们将在第 2 章中看到，创建并部署我们自己的新应用层协议是非常容易的。

应用层协议分布在多个端系统上，而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。我们把这种位于应用层的信息分组称为 **报文（message）**。

- **运输层**

Internet 的运输层负责在应用程序端点之间传送应用层报文。在 Internet 中，有两种运输协议，即 TCP 和 UDP,利用其中的任一个都能运输应用层报文。TCP 向它的应用程序提供了面向连接的服务。这种服务包括了应用层报文向目的地的确保传递和流量控制（即发送方/接收方速率匹配）。TCP 也将长报文划分为短报文，并提供拥塞控制机制，因此当网络拥塞时，源抑制其传输速率。UDP 协议向它的应用

程序提供无连接服务。这是一种不提供不必要服务的服务，没有可靠性，没有流量控制，也没有拥塞控制。在本书中，我们把运输层的分组称为 **报文段(segment)**。

- **网络层**

Internet 的网络层负责将称为 **数据报（datagram）** 的网络层分组从一台主机移动到另一台主机。在一台源主机中的 Internet 运输层协议（TCP 或 UDP）向网络层递交运输层报文段和目的地址。Internet 的网络层包括著名的网际协议 IP，该协议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。IP 仅有一个，所有具有网络层的 Internet 组件必须运行 IP。 Internet 的网络层也包括决定路由的路由选择协议，它根据该路由将数据报从源传输到目的地。Internet 具有许多路由选择协议。

- **链路层**

Internet 的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（主机或路由器）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。特别是在每个节点，网络层将数据报下传给链路层，链路层沿着路径将数据报传递给下一个节点。在该下一个节点，链路层将数据报上传给网络层。由链路层提供的服务取决于应用于该链路的特定链路层协议。例如，某些协议基于链路提供可靠传递，从传输节点跨越一条链路到接收节点。值得注意的是，这种可靠的传递服务不同于 TCP 的可靠传递服务，TCP 提供从一个端系统到另一个端系统的可靠交付。链路层的例子包括以太网、WiFi 和电缆接入网的 DOCS1S 协议。因为数据报从源到目的地传送通常需要经过几条链路，一个数据报可能被沿途不同链路上的不同链路层协议处理。例如，一个数据报可能被一段链路上的以太网和下一段链路上的 PPP 所处理。网络层将受到 来自每个不同的链路层协议的不同服务。在本书中，我们把链路层分组称为 **帧(frame)**。

- **物理层**

虽然链路层的任务是将整个帧从一个网络元素移动到邻近的网络元素，而物理层的任务是将该帧中的一个个比特从一个节点移动到下一个节点。在这层中的协议仍然是链路相关 的，并且进一步与该链路（例如，双绞铜线、单模光纤）的实际传输媒体相关。例如，以太 网具有许多物理层协议：一个是关于双绞铜线的，另一个是关于同轴电缆的，还有一个是关于光纤的，等等。在每种场合中，跨越这些链路移动一个比特是以不同的方式进行的。

1.4.2. 封装

本节我们来仔细观察一下来自发送端系统的分布式应用程序产生的数据是怎么通过 Internet 分层模型一步步到达目的端系统的。

如图 1-24 所示。

数据从发送端的协议栈向下，通过链路层交换机和路由器的协议栈，然后向上通过接收端的协议栈。

在实现的网络协议上，尽管链路层交换机和路由器都是分组交换机，但是链路层交换机只实现了第一层和第二层，路由器实现了第一层到第三层。这表示路由器可以实现 IP 协议而链路层交换机不能。

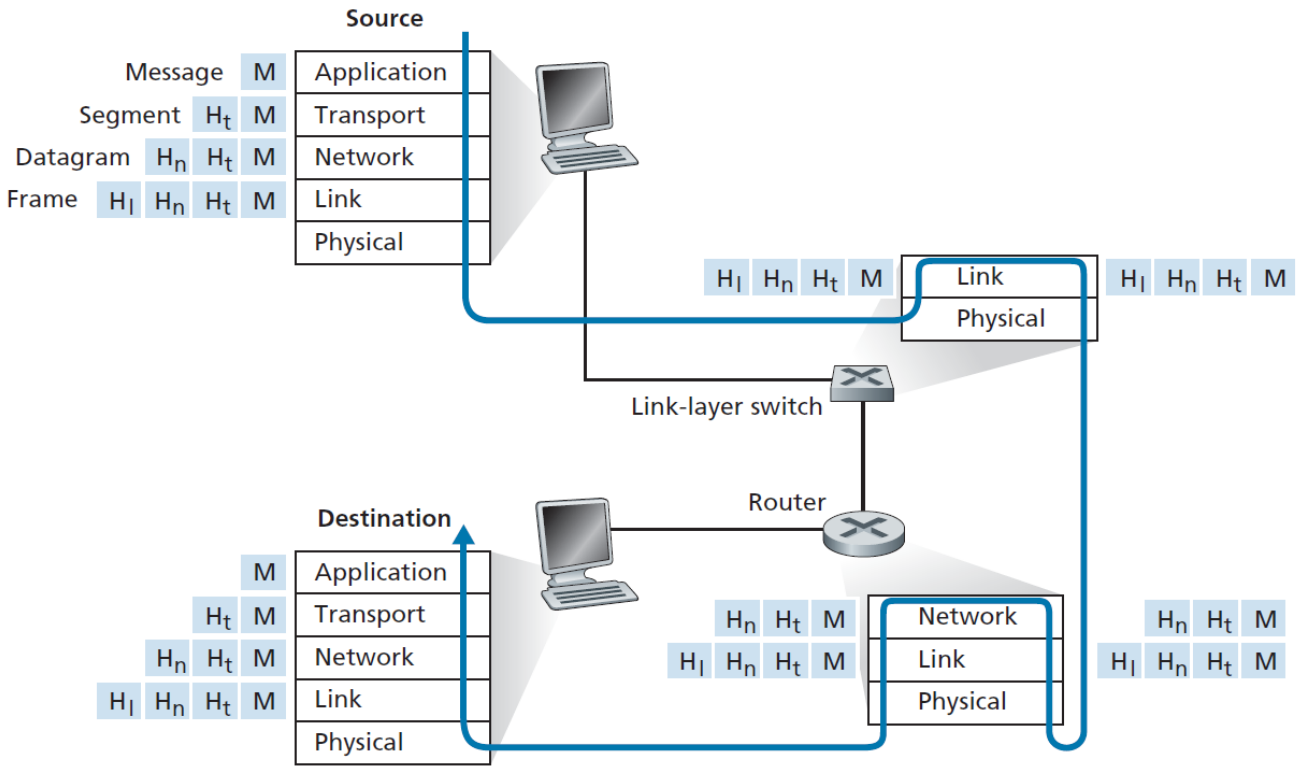


Figure 1.24 ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

在发送主机端，一个应用层报文(M)被传送到运输层，运输层收到报文并加上首部字节(Ht)，Ht 在接收端运输层会被用到。Ht 和 M 共同构成运输层报文段。Ht 可能会涉及以下信息：允许接收端运输层向上向适当的应用程序交付报文的信息；差错检测位信息，该信息让接收方能够判断报文中的比特是否在途中已被改变。报文段被传输到网络层，网络层会附加给报文段一些信息(Hn)，如：发送端和接收端地址等网络层信息。Hn 和报文段构成了网络层数据报。数据报被传输到链路层，链路层给数据报附加上所需信息 HI，构成链路层帧。这就是封装。

1.5. 计算机网络的性能

1.5.1. 节点总时延

前面讲过，分组从一台主机（源）出发，通过一系列路由器传输，在另一台主机(目的地)中结束它的历程。当分组从一个节点(主机或路由器)沿着这条路径到后继节点(主机或路由器)，该分组在沿途的每个节点经受了几种不同类型的时延。这些时延最为重要的是 **节点处理时延(nodal process1ng delay)**，**排队时延(queuing delay)**、**传输时延(transmiss1on delay)** 和 **传播时延(propagation delay)**，这些时延总体累加起来是 **节点总时延(tolal nodal delay)**。许多因特网应用，如搜索、Web 浏览、电子邮件、地

图、即时通讯和 IP 语音，它们的性能受网络时延的影响很大。为了深入理解分组交换和计算机网络，我们必须理解这些时延的性质和重要性。

- **时延的类型**

请看图 1-16 所示的例子。

当分组从上游节点到达路由器 A 时，路由器 A 检查该分组的首部以决定它的出链路。这个例子中只有通向 B。当 A 的出链路不是占用状态时，才会立即传输该分组。如果出链路是占用状态，那么该分组会进入 A 的输出缓存排队。

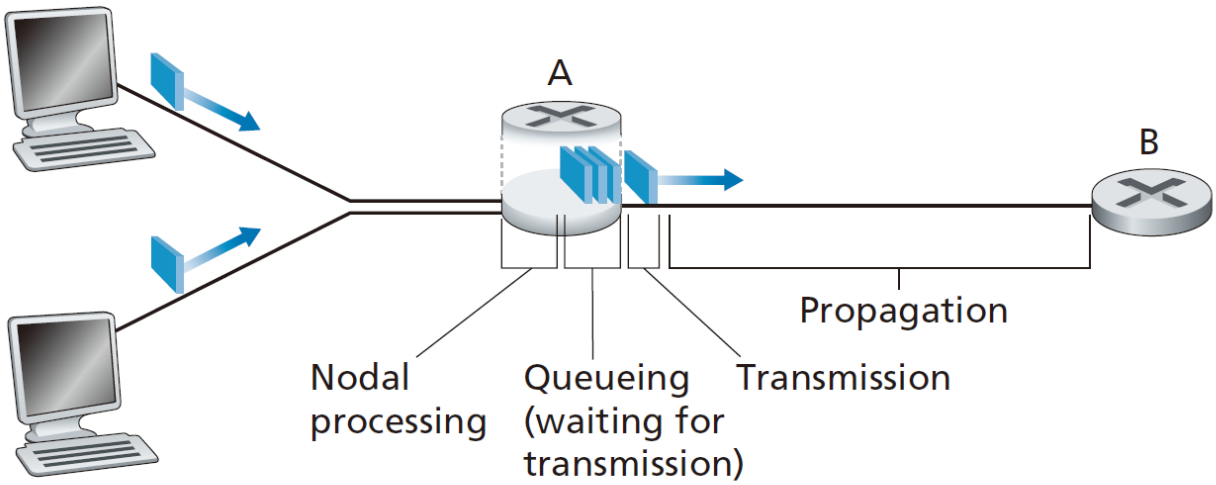


Figure 1.16 ♦ The nodal delay at router A

1. **处理时延**

路由器检查一个分组的首部字节和决定将该分组导向何处是 **处理时延** 的一部分。处理时延还包括检查比特级别的差错所需要的时间。在这种节点处理之后，路由器将该分组引向通往路由器 B 链路之前的队列。

处理时延通常是微妙或更低的数量级。

2. **排队时延**

在队列中，当分组在链路上等待传输时，它经受排队时延。一个特定分组的排队时延长度将取决于先期到达的正在排队等待向链路传输的分组数量。如果该队列是空的，并且当前没有其他分组正在传输，则该分组的排队时延为 0。另一方面，如果流量很大，并且许多其 他分组也在等待传输，该排队时延将很长。我们将很快看到，到达分组期待发现的分组数量是到达该队列的流量的强度和性质的函数。

实际上的排队时延通常是毫秒到微妙量级。

3. **传输时延**

假定分组以先到先服务方式传输——这在分组交换网中是常见的方式，仅当所有已经到达的分组被传输后，才能传输刚到达的分组。用 L 比特表示该分组的长度，用 R bps (即 b/s)表示从路由器 A 到路由器 B 的链路传输速率。例如，对于一条 10Mbps 的以太网链路，速率 R = 10Mbps；对于 100Mbps 的以太网链路，速率 R = 100Mbps。 传输时延是 L/R。这是将所有分组的比特推向链路(即传输，或者说发射)所需要的时间。

实际的传输时延通常在毫秒到微秒量级。

4. **传播时延**

一旦一个比特被推向链路，该比特需要向路由器 B 传播。从该链路的起点到路由器 B 传播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。该传播速率取决于该链路的物理媒体（即光纤、双绞铜线等），其速率范围是 $2 \times 10^8 - 3 \times 10^8$ m/s，这等于或略小于光速。该传播时延等于两台路由器之间的距离除以传播速率。即传播时延是 d/s。其中 d 是路由器 A 和路由器 B 之间的距离，s 是该链路的传播速率。一旦该分组的最后一个比特传播到节点 B，该比特及前面的所有比特被存储于路由器 B。整个过程将随着路由器 B 执行转发而持续下去。

在广域网中，传播时延为毫秒量级。

如果令 d_{proc}，d_{queue}，d_{trans}，d_{prop}分别表示处理时延、排队时延、传输时延和传播时延，则节点的总时延由下式给定：

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

这些时延成分所起的作用可能会有很大的不同。例如，对于连接两台位于同一个大学校园的路由器的链路而言，d_{prop} 可能是微不足道的（例如，几微秒）；然而，对于由同步卫星链路互联的两台路由器来说 d_{prop} 是几百毫秒，能够成为 d_{nodal} 中的主要成分。类似地，d_{trans} 的影响可能是微不足道的，也可能是很大的。通常对于 10Mbps 和更高的传输速率（例如，对于 LAN）的信道而言，它的影响是微不足道的；然而，对于通过低速拨号调制解调器链路发送的长因特网分组而言，可能是数百毫秒。处理时延 d_{proc}通常是微不足道的；然而，它对一台路由器的最大吞吐量有重要影响，最大吞吐量是一台路由器能够转发分组的最大速率。

1.5.2. 丢包

节点总时延的最为复杂和有趣的成分是排队时延 d_{queue}。与其他 3 项时延（即 d_{proc}，d_{trans}和 d_{prop}）不同的是，排队时延对不同的分组可能是不同的。例如，如果 10 个分组同时到达空队列，传输的第一个分组没有排队时延，而传输的最后一个分组将经受相对大的排队时延（这时它要等待其他 9 个分组被传输）。因此，当表征排队时延时，人们通常使用统计量来度量，如平均排队时延、排队时延的方差和排队时延超过某些特定值的概率。

什么时候排队时延大，什么时候又不大呢？该问题的答案很大程度取决于流量到达该队列的速率、链路的传输速率和到达流量的性质，即流量是周期性到达还是以突发形式到达。为了更深入地领会某些要点，令 a 表示分组到达队列的平均速率（a 的单位是分组/秒，即 pkt/s）。 前面讲过 R 是传输速率，即从队列中推出比特的速率（以 bps 即 b/s 为单位）。为了简单起见，也假定所有分组都是由 L 比特组成的。则比特到达队列的平均速率是 La bps。最后，假定该队列非常大，因此它基本能容纳无限数量的比特。比率 La/R 被称为 **流量强度(traffic intens1ty)**，它在估计排队时延的范围方面经常起着重要的作用。如果 La/R > 1，则比特到达队列的平均速率超过从该队列传输出去的速率。在这种不幸的情况下，该队列趋向于无限增加，并且排队时延将趋向无穷大！因此，流量工程中的一条金科玉律是：设计系统时流量强度不能大于 1。

现在考虑 La/R <= 1 时的情况。这时，到达流量的性质影响排队时延。例如，如果分组周期性到达，即每 L/R 秒到达一个分组，则每个分组将到达一个空队列中，不会有排队时延。另一方面，如果分组以突发形式到达而不是周期性到达，则可能会有很大的平均排队时延。例如，假定每 (L/R)N 秒同时到达 N 个分组。则传输的第一个分组没有排队时延; 传输的第二个分组就有 L/R 秒的排队时延；更为一般地，第几个传输的分组具有(n-1)L/R 的排队时延。我们将该例子中的计算平均排队时延的问题留给读者作为练习。

以上描述周期性到达的两个例子有些学术味。通常，到达队列的过程是随机的，即到达并不遵循任何模式，分组之间的时间间隔是随机的。在这种更为真实的情况下，量 L/R 通常不足以全面地表征时延的统计量。不过，直观地理解排队时延的范围很有用。特别是，如果流量强度接近于 0，则几乎没有分组到达并且到达间隔很大，那么到达的分组将不可能在队列中发现别的分组。因此，平均排队时延将接近 0。另一方面，当流量强度接近 1 时，当到达速率超过传输能力(由于分组到达速率的波动)时将存在时间间隔，在这些时段中将形成队列。当到达速率小于传输能力时，队列的长度将缩短。无论如何，随着流量强度接近，平均排队长度变得越来越长。平均排队时延与流量强度的定性关系如图 1-18 所示。

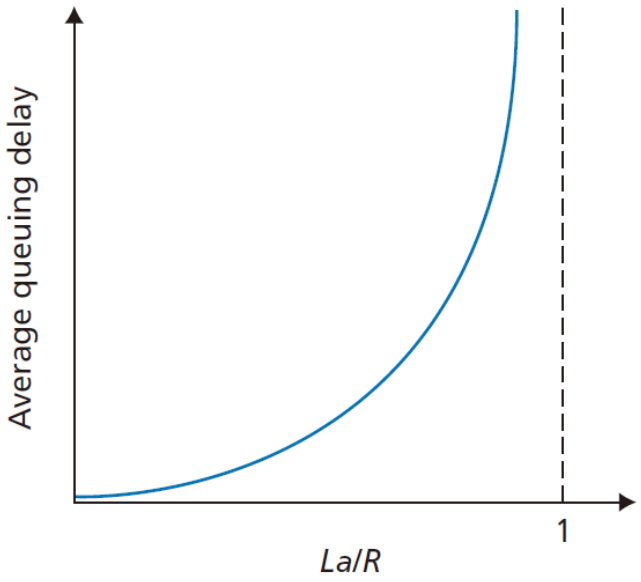


Figure 1.18 ♦ Dependence of average queuing delay on traffic intensity

图 1-18 的一个重要方面是这样一个事实：随着流量强度接近于 1，平均排队时延迅速增加。该强度的少量增加将导致时延大比例增加。也许你在公路上 经历过这种事。如果在经常拥塞的公路上像平时一样驾驶，这条路经常拥塞的事实意味着它的流量强度接近于 1,如果某些事件引起一个即便是稍微大于平常量的流量，经受的时延就可能很大。

为了实际感受一下排队时延的情况，我们再次鼓励你访问本书的 Web 网站(进入动画)。如果你将分组到达速率设置得足够大，使流量强度超过 1，那么将看到经过一段时间后，队列慢慢地建立起来。

在上述讨论中，我们已经假设队列能够容纳无穷多的分组。在现实中，一条链路前的队列只有有限的容量，尽管排队容量极大地依赖于路由器设计和成本。因为该排队容量是 有限的，随着流量强度接近 1，排队时延并不真正趋向无穷大。相反，到达的分组将发现一个满的队列。由于没有地方存储这个分组，路由器将丢弃该分组，即该分组将会 **丢包(lost)**。当流量强度大于 1 时，队列中的这种溢出也能够在配套 web 网站的动画中看到。

从端系统的角度看，上述丢包现象看起来是一个分组已经传输到网络核心，但它绝不会从网络发送到目的地。分组丢失的比例随着流量强度增加而增加。因此，一个节点的性能常常不仅根据时延来度量，而且根据丢包的概率来度量。正如我们将在后面各章中讨论的那样，丢失的分组可能基于端到端的原则重传，以确保所有的数据最终从源传送到了目的地。

1.5.3. 端到端时延

前面的讨论一直集中在节点时延上，即在单台路由器上的时延。我们现在考虑从源到目的地的总时延。为了能够理解这个概念，假定在源主机和目的主机之间有 N-1 台路由器。我们还要假设该网络此时是无

拥塞的(因此排队时延是微不足道的)，在每台路由器和源主机上的处理时延是 d_{proc}，每台路由器和源主机的输出速率是 R bps，每条链路的传播时延是 $d_{trans})$。节点时延累加起来，得到端到端时延：

$$d_{end-end} = N(d_{proc} + d_{trans} + d_{prop})$$

其中，d_{trans} = L / R，L 为分组的长度。

在各节点具有不同的时延和每个节点存在平均排队时延的情况下，需要对上式进行一般化处理。我们将有关工作留给读者。

- **Traceroute**

为了对计算机网络中的端到端时延有第一手认识，我们可以利用 Traceroute 程序。 Traceroute 是一个简单的程序，它能够在任何因特网主机上运行。当用户指定一个目的主机名字时，源主机中的该程序朝着目的地发送多个特殊的分组。当这些分组向着目的地传送时，它们通过一系列路由器。当路由器接收到这些特殊分组之一时，它向源回送一个短报文。该短报文包括路由器的名字和地址。

更具体来说，假定在源和目的地之间有 N-1 台路由器。源将向网络发送 N 个特殊的分组，其中每个分组地址指向最终目的地。这 N 个特殊分组标识为从 1 到 N，第一个分组标识为 1，最后的分组标识为 N。当第 n 台路由器接收到标识为 n 的第 n 个分组时，该路由器不是向它的目的地转发该分组，而是向源回送一个报文。当目的主机接收第 N 个分组时，它也会向源返回一个报文。该源记录了从它发送一个分组到它接收到对应返回报文所经历的时间；它也记录了返回该报文的路由器（或目的主机）的名字和地址。以这种方式，源能够重建分组从源到目的地所采用的路由，并且该源能够确定到所有中间路由器的往返时延。Traceroute 实际上对刚才描述的实验重复了 3 次，因此该源实际上向目的地发送了 3*N 个分组。RFC 1393 详细地描述了 Traceroute。

这里我们提供了一个了一个追踪本地主机到 www.baidu.com 的一个例子。

```
tracert www.baidu.com

通过最多 30 个跃点跟踪
到 www.a.shifen.com [110.242.68.3] 的路由:

  1     2 ms     2 ms     2 ms  bogon [192.168.43.141]
  2     *        *        *     请求超时。
  3     *        *        *     请求超时。
  4    22 ms    35 ms    33 ms  123.139.0.221
  5     *       62 ms    27 ms  gi0-0-rtr1-xgx-man.169cnc.net [221.11.0.1]
  6    64 ms    48 ms    40 ms  gi3-0-rtr1-dwl-man.169cnc.net [221.11.0.53]
  7    77 ms    44 ms    49 ms  219.158.111.233
  8    82 ms    76 ms    51 ms  110.242.66.178
  9     *        *        *     请求超时。
 10     *        *        *     请求超时。
 11     *        *        *     请求超时。
 12     *        *        *     请求超时。
 13    43 ms    58 ms    61 ms  110.242.68.3

跟踪完成。
```

- **其他时延**

除了处理时延、传输时延和传播时延，端系统中还有其他一些重要时延。例如，希望向共享媒体（例如在 WiFi 或电缆调制解调器情况下）传输分组的端系统可能有意地延迟它的传输，把这作为它与其他端系统共享媒体的协议的一部分；我们将在第 6 章中详细地考虑这样的协议。另一个重要的时延是媒体分组化时延，这种时延出现在 IP 语音（VoIP） 应用中。在 VoIP 中，发送方在向因特网传递分组之前必须首先用编码的数字化语音填充一个分组。这种填充一个分组的时间称为分组化时延，它可能较大，并能够影响用户感受到的 VoIP 呼叫的质量。这个问题将在本章结束的课后作业中进一步探讨。

1.5.4. 吞吐量

除了时延和丢包，计算机网络中另一个至关重要的性能测度是端到端吞吐量。为了定义吞吐量，考虑从主机 A 到主机 B 跨越计算机网络传送一个大文件。例如，也许是从一个 P2P 文件共享系统中的一个对等方向另一个对等方传送一个大视频片段。在任何时间瞬间的 **瞬时吞吐量(instantaneous throughput)** 是主机 B 接收到该文件的速率（以 bps 计）。（许多应用程序包括许多 P2P 文件共享系统，其用户界面显示了下载期间的瞬时吞吐量，也许你以前已经观察过它！）如果该文件由 F 比特组成，主机 B 接收到所有 F 比特用去 T 秒，则文件传送的 **平均吞吐量(average throughput)** 是 F/T bps。 对于某些应用程序如因特网电话，希望具有低时延和在某个阈值之上（例如，对某些因特网电话是超过 24kbps，对某些实时视频应用程序是超过 256kbps）的一致的瞬时吞吐量。对于其他应用程序，包括涉及文件传送的那些应用程序，时延不是决定性的，但是希望具有尽可能高的吞吐量。

为了进一步深入理解吞吐量这个重要概念，我们考虑几个例子。图 1-19a 显示了服务器和客户这两个端系统，它们由两条通信链路和一台路由器相连。考虑从服务器传送一个文件到客户的吞吐量。令 Rs 表

示服务器与路由器之间的链路速率；Rc 表示路由器与客户之间的链路速率。假定在整个网络中只有从该服务器到客户的比特在传送。在这种理想的情况下，我们要问该服务器到客户的吞吐量是多少？为了回答这个问题，我们可以想象比特是流体，通信链路是管道。显然，这台服务器不能以快于 Rs bps 的速率通过其链路注入比特；这台路由器也不能以快于 Rcbps 的速率转发比特。如果 Rs < Rc，则在给定的吞吐量 Rs bps 的情况下，由该服务器注入的比特将顺畅地通过路由器“流动”，并以速率 Rs bps 到达客户。另一方面，如果 Rc < Rs，则该路由器将不能像接收速率那样快地转发比特。在这种情况下，比特将以速率 Rc 离开该路由器，从而得到端到端吞吐量 Rc。（还要注意的是，如果比特继续以速率 Rs 到达路由器，继续以 Rc 离开路由器的话，在该路由器中等待传输给客户的积压比特将不断增加，这是一种最不希望的情况！）因此，对于这种简单的两链路网络，其吞吐量是 min {Rc, Rs}。这就是说，它是 **瓶颈链路(bottleneck link)** 的传输速率。在决定了吞吐量之后，我们现在近似地得到从服务器到客户传输一个 F 比特的大文件所需要的时间是 F / min{Rc, Rs}。举一个特定的例子，假你正在下载一个 F = 32 X 10^6 比特的 MP3 文件，服务器具有 Rs = 2 Mbps 的传输速率，并且你有一条 Rc = 1 Mbps 的接入链路。则传输该文件所需的时间是 32 秒。当然，这些吞吐量和传输时间的表达式仅是近似的，因为它们并没有考虑存储转发、处理时延和协议等问题。

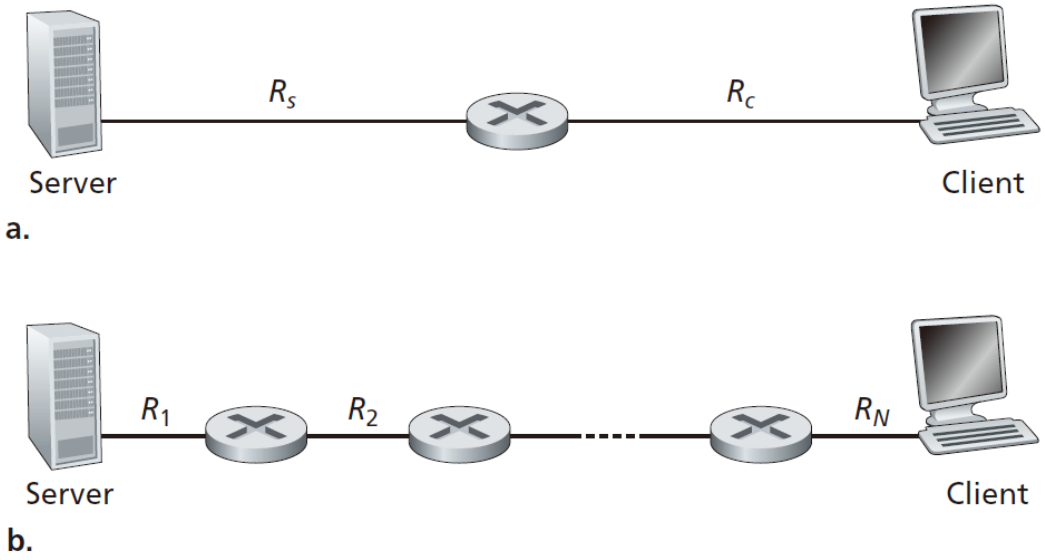


Figure 1.19 ♦ Throughput for a file transfer from server to client

图 1-19b 此时显示了一个在服务器和客户之间具有 N 条链路的网络，这 N 条链路的传输速率分别是 R1, R2, ..., Rn。应用对两条链路网络的分析方法，我们发现从服务器到客户的文件传输吞吐量是 min {R1, R2, ..., Rn}，这同样仍是沿着服务器和客户之间路径的瓶颈链路的速率。

现在考虑由当前因特网所引发的另一个例子。图 1-20a 显示了与一个计算机网络相连的两个端系统：一台服务器和一个客户。考虑从服务器向客户传送一个文件的吞吐量。服务器以速率为 Rs 的接入链路与网络相连，且客户以速率为 Rc 的接入链路与网络相连。现在假定在通信网络核心中的所有链路具有非常高的传输速率，即该速率比 Rs 和 Rc 要高得多。目前因特网的核心的确超量配置了高速率的链路，

从而很少出现拥塞。同时假定在整个网络中发送的比特都是从该服务器到该客户。在这个例子中，因为计算机网络的核心就像一个粗大的管子，所以比特从源向目的地的流动速率仍是 Rs 和 Rc 中的最小者，即吞吐量 = min {Rs, Rc}。因此，在今天因特网中对吞吐量的限制因素通常是接入网。

作为最后一个例子，考虑图 1-20b,其中有 10 台服务器和 10 个客户与某计算机网络 核心相连。在这个例子中，同时发生 10 个下载，涉及 10 个客户-服务器对。假定这 10 个下载是网络中当时的唯一流量。如该图所示，在核心中有一条所有 10 个下载通过的链路。将这条链路的传输速率表示为 R。假定所有服务器接入链路具有相同的速率 Rs，所有客户接入链路具有相同的速率 Rc，并且核心中除了速率为 R 的一条共同链路之外的所有链路，它们的传输速率都比 R、Rs 和 Rc 大得多。现在我们要问，这种下载的吞吐量是多少？显然，如果该公共链路的速率 R 很大，比如说比 Rs 和 Rc 大 100 倍，则每个下载的吞吐量将仍然是 min {Rs, Rc}。 但是如果该公共链路的速率与 Rs 和 Rc 有相同量级会怎样呢？在这种情况下其吞吐量将是多少呢？让我们观察一个特定的例子。假定 Rs = 2Mbps, Rc = 1 Mbps, R = 5Mbps,并且公共链路为 10 个下载平等划分它的传输速率。这时每个下载的瓶颈不再位于接入网中，而是位于核心中的共享链路了，该瓶颈仅能为每个下载提供 500 kbps 的吞吐量。因此每个下载的端到端吞吐量现在减少到 500 kbps。

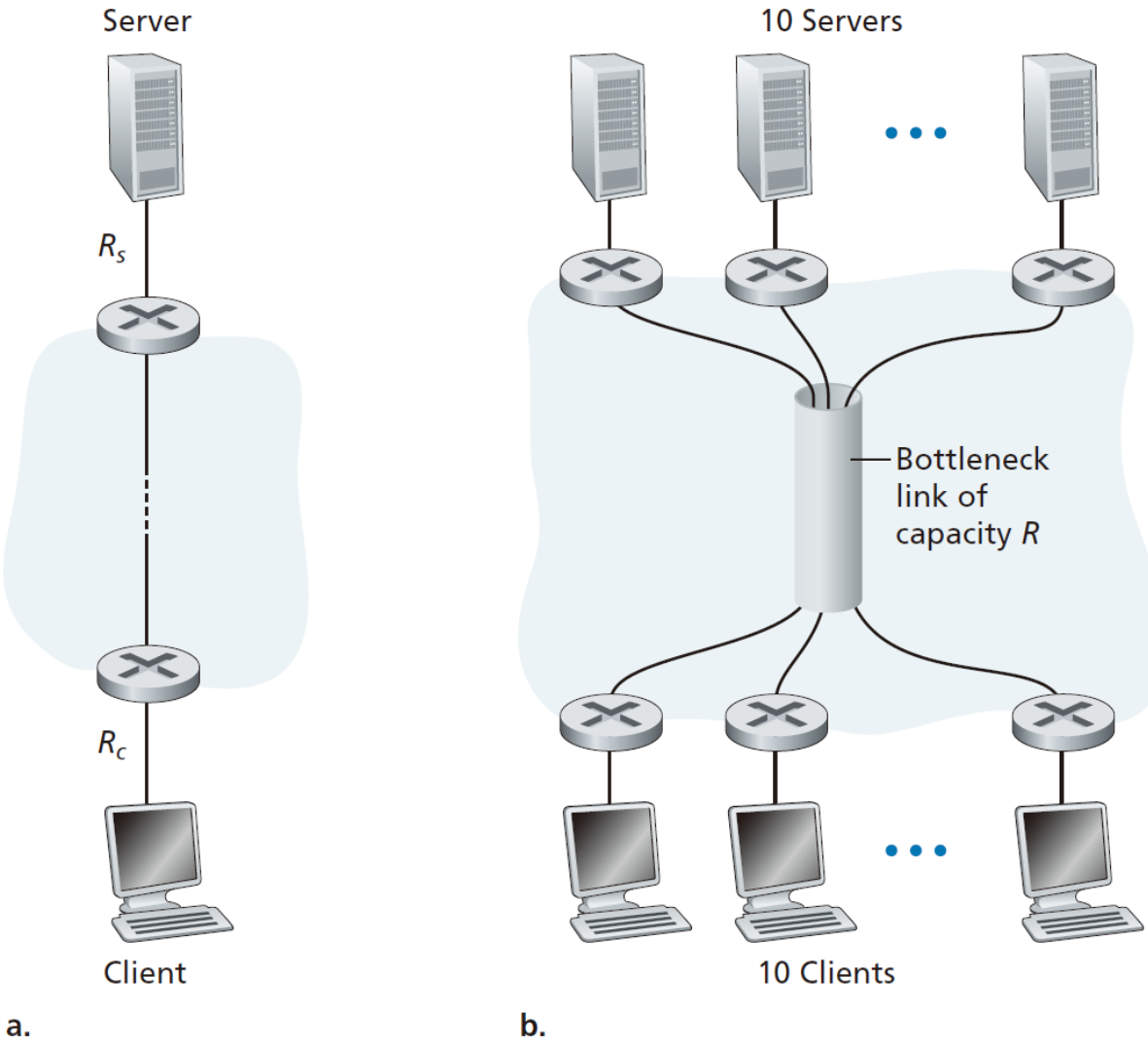


Figure 1.20 ♦ End-to-end throughput: (a) Client downloads a file from server; (b) 10 clients downloading with 10 servers

图 1-19 和图 1-20 中的例子说明吞吐量取决于数据流过的链路的传输速率。我们看到当没有其他干扰流量时，其吞吐量能够近似为沿着源和目的地之间路径的最小传输速率。图 1-20b 中的例子更一般地说明了吞吐量不仅取决于沿着路径的传输速率，而且取决于干扰流量。特别是，如果许多其他的数据流也通过这条链路流动，一条具有高传输速率的链路仍然可能成为文件传输的瓶颈链路。我们将在课后习题中和后继章节中更仔细地研究计算机网络中的吞吐量。

1.6. 计算机网络的安全性

对于今天的许多机构（包括大大小小的公司、大学和政府机关）而言，因特网已经成为与其使命密切相关的一部分了。许多人也依赖因特网从事各种职业、社会和个人活动。目前，数以亿计的物品（包括可穿戴设备和家用设备）与因特网相连。但是在所有这一切背后，存在着一个阴暗面，其中的“坏家伙”试

图对我们的日常生活进行破坏，如损坏我们与因特网相连的计算机，侵犯我们的隐私以及使我们依赖的因特网服务无法运行。

网络安全领域主要探讨以下问题：坏家伙如何攻击计算机网络，以及我们（即将成为计算机网络的专家）如何防御以免受他们的攻击，或者更好的是设计能够事先免除这样的攻击的新型体系结构。面对经常发生的各种各样的现有攻击以及新型和更具摧毁性的未来攻击的威胁，网络安全已经成为近年来计算机网络领域的中心主题。本书的特色之一是将网络安全问题放在中心位置。

因为我们在计算机网络和因特网协议方面还没有专业知识，所以这里我们将从审视某 些今天最为流行的与安全性相关的问题开始。这将刺激我们的胃口，以便我们在后续章节中进行更为充实的讨论。我们在这里以提出问题开始：什么会出现问题？计算机网络是如何受到攻击的？今天一些最为流行的攻击类型是什么？

1.6.1. 恶意软件

因为我们要从/向因特网接收/发送数据，所以我们将设备与因特网相连。这包括各种好东西，例如 Instagram 帖子、因特网搜索结果、流式音乐、视频会议、流式电影等。但不幸的是，伴随好的东西而来的还有恶意的东西，这些恶意的东西可统称为 **恶意软件(malware)**，它们能够进入并感染我们的设备。一旦恶意软件感染我们的设备，就能够做各种不正当的事情，包括删除我们的文件，安装间谍软件来收集我们的隐私信息，如身份证号、口令和击键，然后将这些（当然经因特网）发送给坏家伙。我们的受害主机也可能成为数以千计的类似受害设备网络中的一员，它们被统称为 **僵尸网络(botnet)**，坏家伙利用僵尸网络控制并有效地对目标主机展开垃圾邮件分发或分布式拒绝服务攻击（很快将讨论）。

至今为止的多数恶意软件是 **自我复制(self-replicating)** 的：一旦它感染了一台主机，就会从那台主机寻求进人因特网上的其他主机，从而形成新的感染主机，再寻求进入更多的主机。以这种方式，自我复制的恶意软件能够指数式地快速扩散。恶意软件能够以病毒或蠕虫的形式扩散。**病毒(virus)** 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。典型的例子是包含恶意可执行代码的电子邮件附件。如果用户接收并打开这样的附件，不经意间就在其设备上运行了该恶意软件。通常，这种电子邮件病毒是自我复制的: 例如，一旦执行，该病毒可能向用户地址簿上的每个接收方发送一个具有相同恶意附件的相同报文。**蠕虫(worm)** 是一种无须任何明显用户交互就能进入设备的恶意软件。例如，用户也许运行了一个攻击者能够发送恶意软件的脆弱网络应用程序。在某些情况下，没有用户的任何干预，该应用程序可能从因特网接收恶意软件并运行它，生成了蠕虫。新近感染设备中的蠕虫则能扫描因特网，搜索其他运行相同网络应用程序的易受感染的主机。当它发现其他易受感染的主机时，便向这些主机发送一个它自身的副本。今天，恶意软件无所不在且防范成本高。当你用这本书学习时，我们鼓励你思考下列问题：计算机网络设计者能够采取什么防御措施，以使与因特网连接的设备免受恶意软件的攻击？

1.6.2. 拒绝服务(DoS)

另一种宽泛类型的安全性威胁称为 **拒绝服务攻击(Denial-of-Service (DoS) attack)**。 顾名思义，DoS 攻击使得网络、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、DNS

服务器（在第 2 章中讨论）和机构网络都能够成为 DoS 攻击的目标。因特网 DoS 攻击极为常见，每年会出现数以千计的 DoS 攻击［Moore 2001］。访问数字攻击图（Digital Attack Map）站点可以可视化了世界范围内每天最厉害的 DoS 攻击［DAM 2020］。大多数因特网 DoS 攻击属于下列三种类型之一：

- 弱点攻击。这涉及向一台目标主机上运行的易受攻击的应用程序或操作系统发送精心制作的报文。如果适当顺序的多个分组发送给一个易受攻击的应用程序或操作系统，该服务器可能停止运行，或者更糟糕的是主机可能崩溃。
- 带宽洪泛。攻击者向目标主机发送大量的分组，分组数量之多使得目标的接入链路变得拥塞，使得合法的分组无法到达服务器。
- 连接洪泛。攻击者在目标主机中创建大量的半开或全开 TCP 连接(将在第 3 章中讨论 TCP 连接)。该主机因这些伪造的连接而陷入困境，并停止接受合法的连接。

我们现在更详细地研究这种带宽洪泛攻击。回顾 1-4-2 节中讨论的时延和丢包问题，显然，如果某服务器的接入速率为 R bps，则攻击者将需要以大约 R bps 的速率来产生危害。如果 R 非常大的话，单一攻击源可能无法产生足够大的流量来伤害该服务器。此外，如果从单一源发出所有流量的话，某上游路由器就能够检测出该攻击并在该流量靠近服务器之前就将其阻挡下来。在图 1-25 中显示的 **分布式 DoS(Distributed DoS, DDoS)** 中，攻击者控制多个源并让每个源向目标猛烈发送流量。使用这种方法，遍及所有受控源的聚合流量速率需要大约 R 的能力来使该服务陷入瘫痪。DDoS 攻击充分利用由数以千计的受害主机组成的僵尸网络，这在今天是屡见不鲜的［DAM 2020］。相比于来自单一主机的 DoS 攻击，DDoS 攻击更加难以检测和防范。

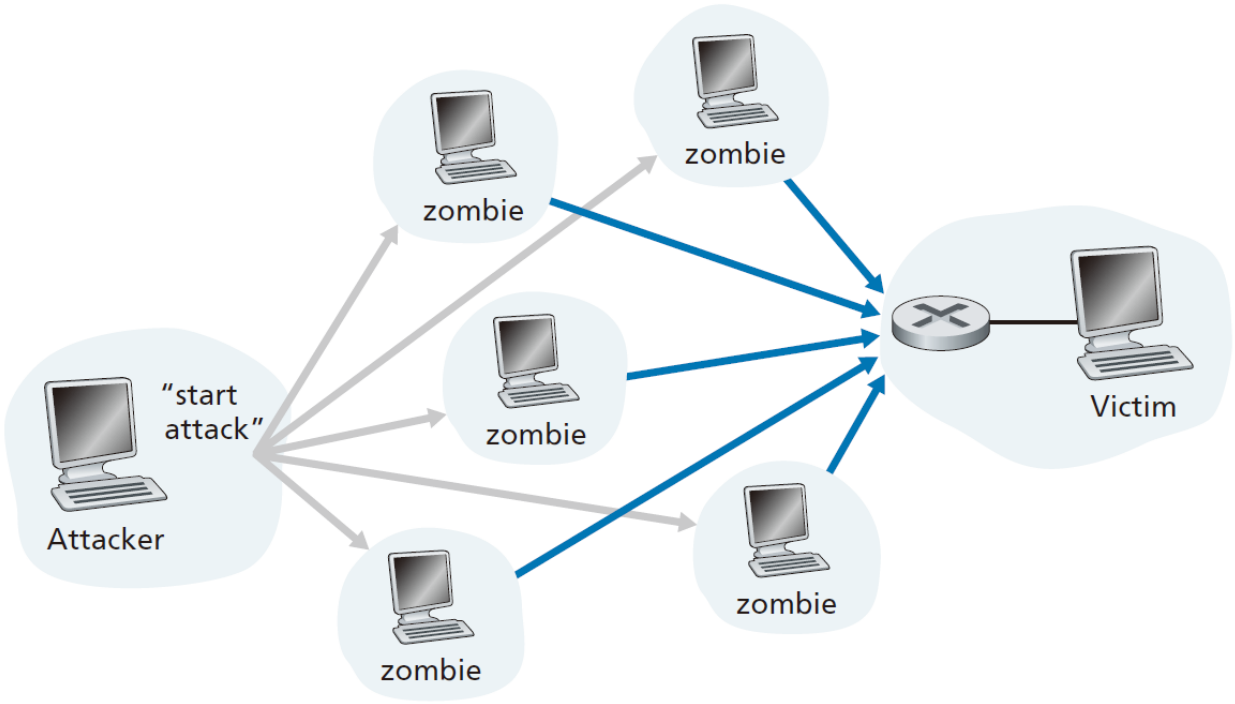


Figure 1.25 ♦ A distributed denial-of-service attack

当学习这本书时，我们鼓励你考虑下列问题：计算机网络设计者能够采取哪些措施防止 DoS 攻击？我们将看到，对于 3 种不同类型的 DoS 攻击需要采用不同的防御方法。

1.6.3. 嗅探分组

今天的许多用户经无线设备接入因特网，如 WiFi 连接的膝上计算机或使用蜂窝因特网连接的手持设备(在第 7 章中讨论)。无所不在的因特网接入极为便利并让移动用户方便地使用令人惊奇的新应用程序的同时，也产生了严重的安全脆弱性：在无线传输设备的附近放置一台被动的接收机，该接收机就能得到传输的每个分组的副本！这些分组包含了各种敏感信息，包括口令、身份证号、商业秘密和隐秘的个人信息。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(packet sniffer)**。

嗅探器也能够部署在有线环境中。在有线的广播环境中，如在许多以太网 LAN 中，分组嗅探器能够获得经该 LAN 发送的所有分组。如在前面描述的那样，电缆接入技术也广播分组，因此易于受到嗅探攻击。此外，获得某机构与因特网连接的接入路由器或接入链路访问权的坏家伙能够放置一台嗅探器以产生从该机构出入的每个分组的副本，再对嗅探到的分组进行离线分析，就能得出敏感信息。

分组嗅探软件在各种 Web 站点上可免费得到，这类软件也有商用的产品。教网络课程的教授布置的实验作业就涉及写一个分组嗅探器和应用层数据重构程序。与本书相关联的 Wireshark [Wireshark 2020] 实验（参见本章结尾处的 Wireshark 实验介绍）使用的正是这样一种分组嗅探器！

因为分组嗅探器是被动的，也就是说它们不向信道中注入分组，所以难以检测到它们。因此，当我们向无线信道发送分组时，我们必须接受这样的可能性，即某些坏家伙可能记录了我们的分组的副本。如你已经猜想的那样，最好的防御嗅探的方法基本上都与密码学有关。我们将在第 8 章研究密码学应用于网络安全的有关内容。

1.6.4. 信任伪装

生成具有任意源地址、分组内容和目的地址的分组，然后将这个人工制作的分组传输到因特网中，因特网将忠实地将该分组转发到目的地，这一切都极为容易（当你学完这本教科书后，你将很快具有这方面的知识了！）。想象某个接收到这样一个分组的不会猜疑的 接收方（比如说一台因特网路由器），将该（虚假的）源地址作为真实的，进而执行某些嵌入在该分组内容中的命令（比如说修改它的转发表）。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(IP spoofing)**，而它只是一个用户能够冒充另一个用户的许多方式中的一种。

为了解决这个问题，我们需要采用端点鉴别，即一种使我们能够确信一个报文源自我们认为它应当来自的地方的机制。当你继续学习本书各章时，再次建议你思考怎样为网络应用程序和协议做这件事。我们将在第 8 章探讨端点鉴别机制。

在本节结束时，值得思考一下因特网是如何从一开始就落入这样一种不安全的境地的。大体上讲，答案是：因特网最初就是基于“一群相互信任的用户连接到一个透明的网 络上”这样的模型［Blumenthal 2001］进行设计的，在这样的模型中，安全性是没有必要的。初始的因特网体系结构在许多方面都深刻地反映了这种相互信任的理念。例如，一个用户向任何其他用户发送分组的能力是默认的，而不是一种请求/准予的能力，还有用户身份取自所宣称的表面价值，而不是默认地需要鉴别。

但是今天的因特网无疑并不涉及“相互信任的用户”。但是，今天的用户仍然需要通信，当他们不必相互信任时，他们也许希望匿名通信，也许间接地通过第三方通信（例如我们将在第 2 章中学习的 Web 代理，我们将在第 7 章学习的移动性协助代理），也许不信任他们通信时使用的硬件、软件甚至空气。随着我们进一步学习本书，会面临许多安全性相关的挑战：我们应当寻求对嗅探、端点假冒、中间人攻击、DDoS 攻击、恶意软件等的防护办法。我们应当记住：在相互信任的用户之间的通信是一种例外而不是规则。欢迎你到现代计算机网络世界！

1.7. 计算机网络的历史

1. 分组交换的发展(1961-1972)
2. 专用网络和网络互联(1972-1980)
3. 网络的激增(1980-1990)
4. Internet 的爆炸(1990-2000)
5. 最新发展(2000 至今)

- 家庭接入网的普及和更迭

- LTE(4G)的发展和 WiFi 的普及
- 在线社交网络的发展
- 在线服务提供商部署了自己的专属网络
- 云技术的发展

1.8. 实验 1：熟悉 wireshark

1.8.1. 分组嗅探器

用来观察执行协议实体之间交换的报文的基本工具称为**分组嗅探器（packet sniffer）**。

顾名思义，一个分组嗅探器被动地拷贝（嗅探）由你的计算机发送和接收的报文；它也能显示出这些被捕获报文的各个协议字段的内容。

图 1-28 中显示了一个常见的分组嗅探器： Wireshark 的截图。

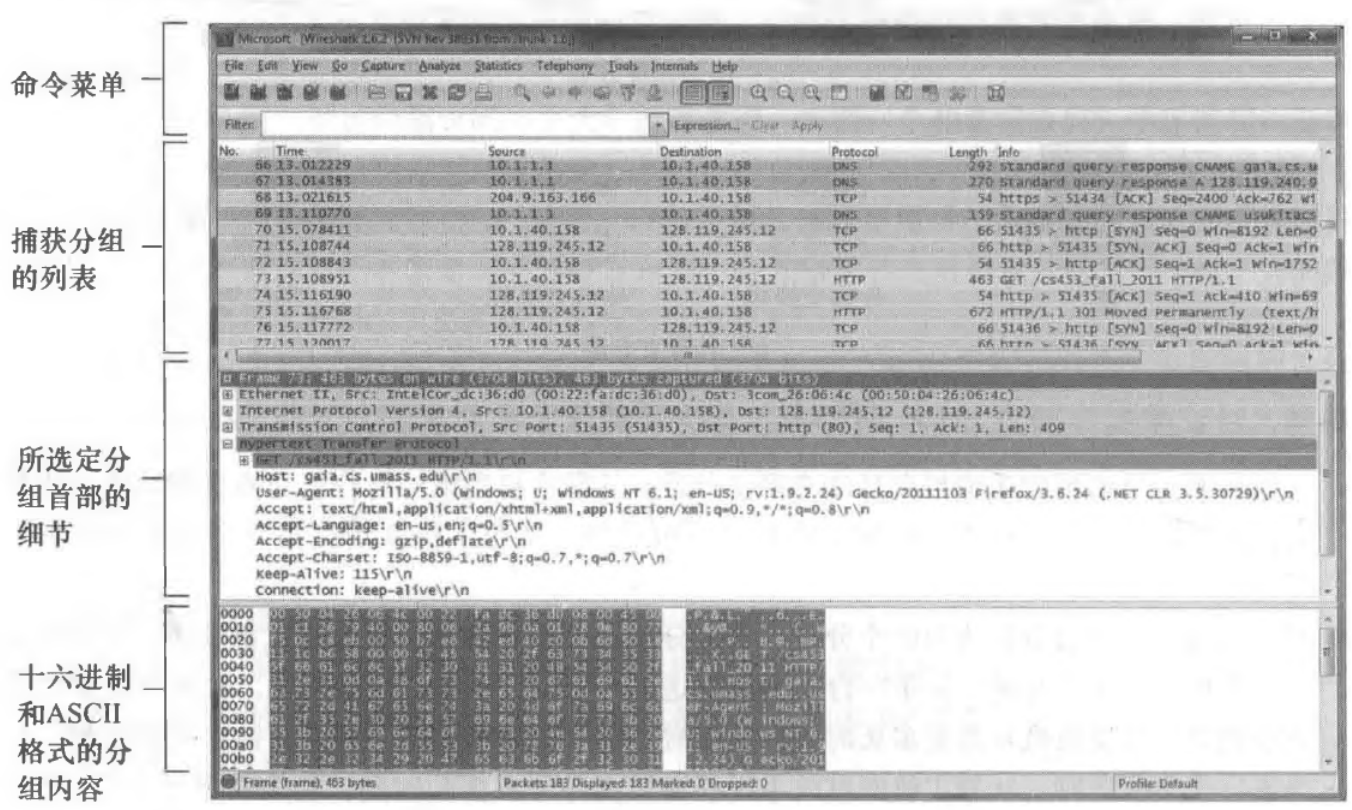


图 1-28　一个 Wireshark 屏幕快照（打印的 Wireshark 屏幕快照得到了 Wireshark 基金会的许可）

下图描述了分组嗅探器的工作原理。

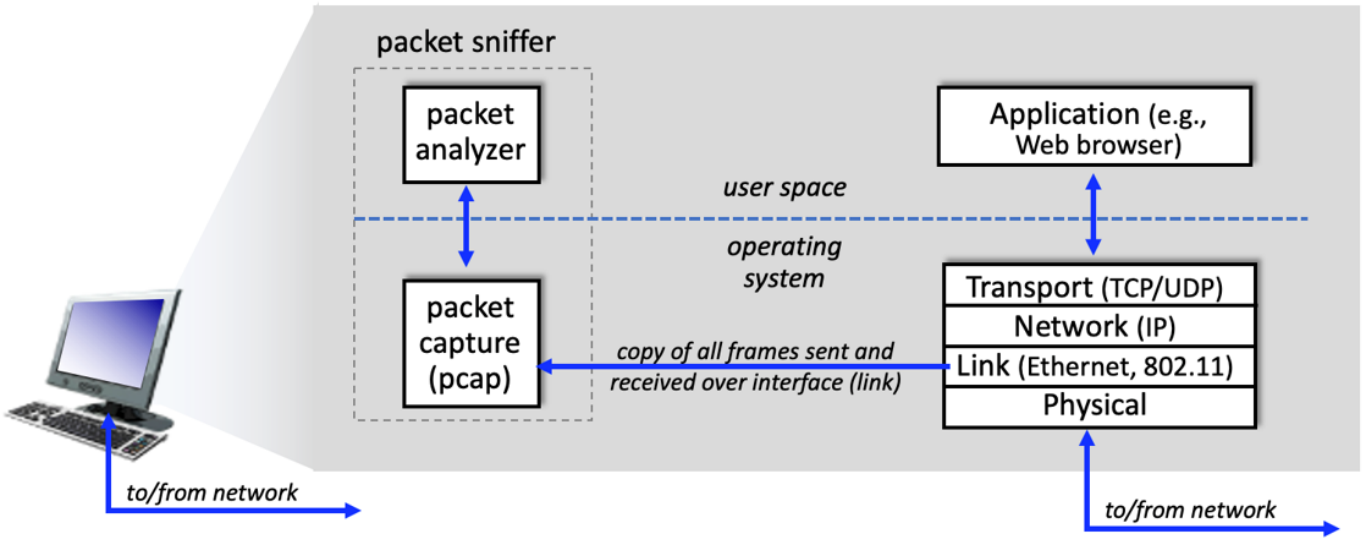


Figure 1: packet sniffer structure

观察上图，在图的右边是分布式应用(如 Web 浏览器)，下面是支撑应用的网络协议层：传输层(TCP/IP)，网络层(IP)，数据链路层(以太网协议，802.11 WiFi 协议)，最后是物理层。图片左边是分组嗅探器，它由 2 部分构成：**分组分析器(packet analyzer)** 和 **分组捕获器(packet capturer)**。

分组捕获器被动地复制来自链路层的分组，回忆一下我们在第 1.5 节讨论到的内容。这些分组称为帧。帧封装了来自网络层的首部字节 Hn，以及来自运输层的首部字节 Ht，当然还有应用层报文 M。

分组分析器可以分析出分组的结构，以便展示出一个应用层报文的内容的字段。比如，我们现在对 HTTP 报文的字段比较感兴趣。分组分析器首先要理解来自数据链路层的帧(以太网帧或 WiFi 帧)的格式，以便它识别出来自网络层的数据报。它也要理解数据报的格式，以便它识别出来自运输层的报文段(TCP 报文段)。最终，分组分析器理解了 TCP 报文段的格式，识别出来自应用层的 HTTP 报文。接着，分组分析器提取出了 HTTP 报文的字段(如 HTTP 请求报文中请求行的请求方法字段，它的值可以为 GET，POST，或 HEAD)，参考图 2.8。

1.8.2. wireshark

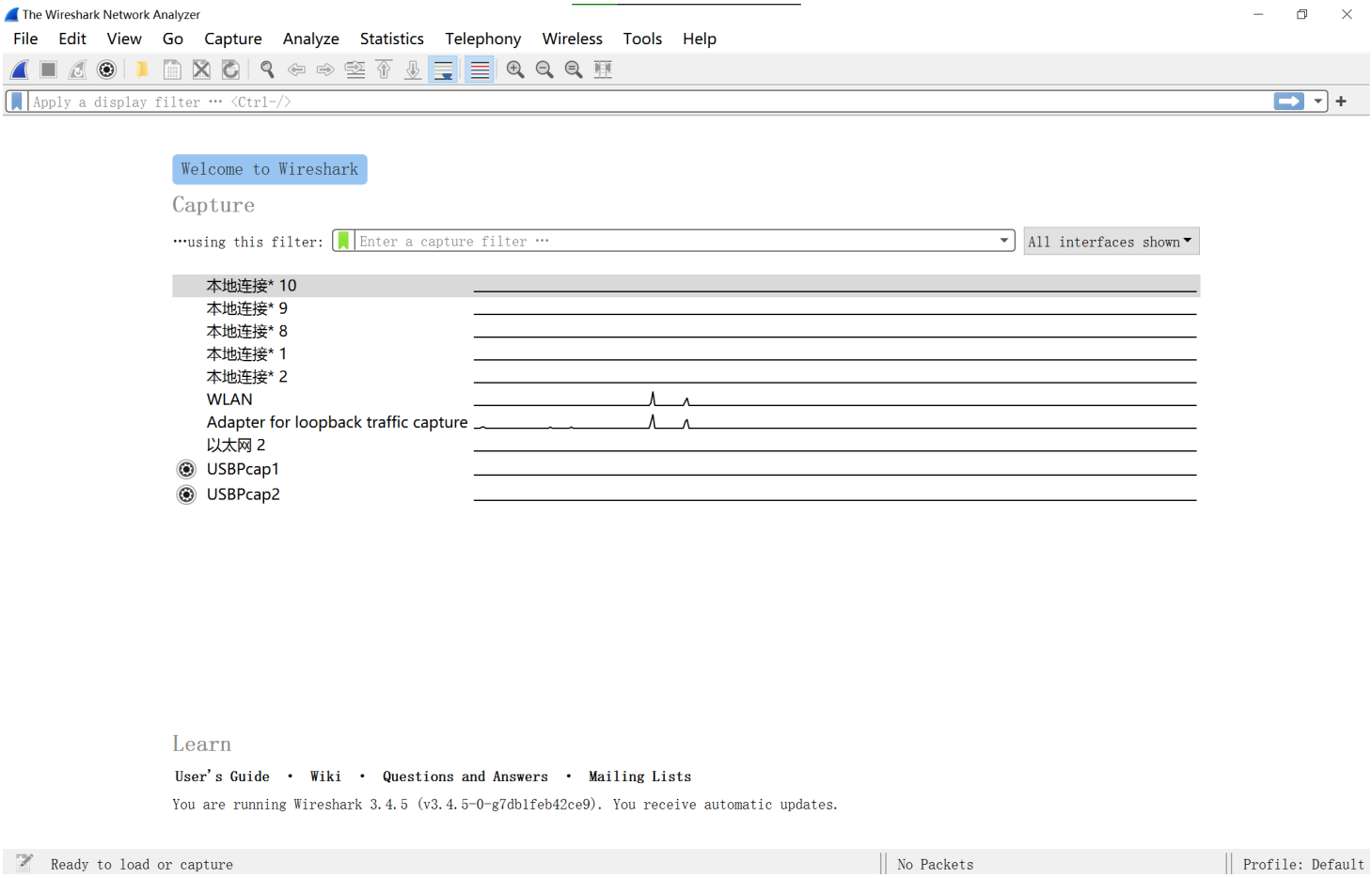
我们将使用 wireshark 分组嗅探器来实验。wireshark 是一个多平台，免费的分组嗅探器。wireshark 拥有大量的用户，完善的指导文档(你可以在 http://www.wireshark.org/ wireshark 的官网找到)和 FAQ。此外，wireshark 拥有丰富的功能和设计良好的用户界面。

1. **获取 wireshark**

你可以在这里下载 wireshark。

2. **运行 wireshark**

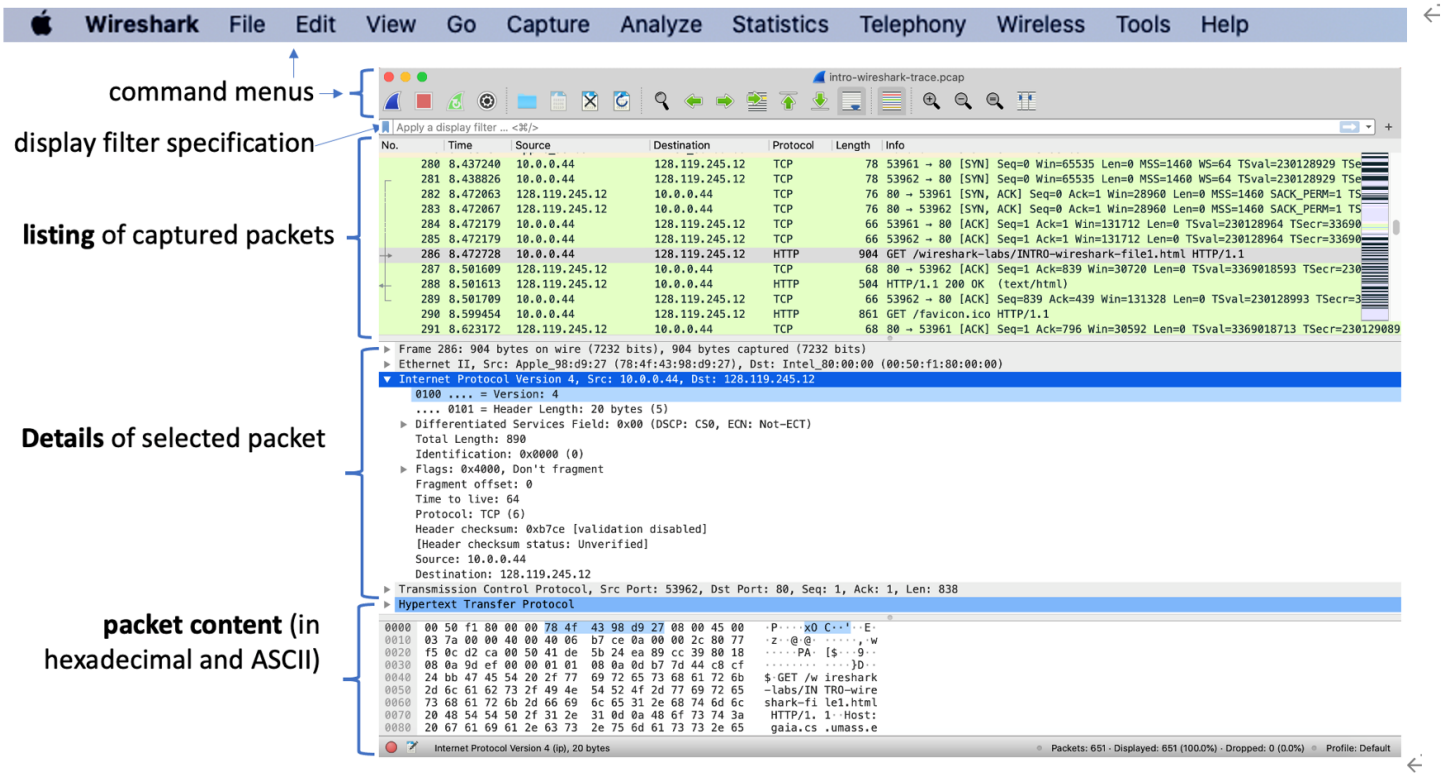
第一次运行 wireshark 后，你将会看到类似下面的屏幕截图。



不同的平台，如 windows 和 macos，不同的语言设置和 wireshark 版本都会影响第一次运行后的界面。不过 wireshark 提供的功能大部分是相同的。这张屏幕截图在 windows 10，wireshark v3.4.5 环境。

在这张图中，注意到在 Capture 下面的列表中有许多所谓的接口。其中 WLAN 表示这台电脑的 WiFi 接入。双击其中一个接口，可以捕获来自这个接口的帧。本电脑使用了 WiFi 接入，所以我们双击 WLAN，就可以开始捕获来自 WLAN 的帧。

下图展示了捕获分组时的界面。



这个界面包含了从上到下的 5 个部分：

- **命令菜单**

命令菜单包含了常见的 wireshark 命令。其中的 **File** 下拉菜单项用于打开以及保存捕获的分组信息文件。**Capture** 下拉菜单项用于选择一项网络接口进行捕获。下面的一栏包含了常用的操作，第 1 个蓝色的鲨鱼图标按钮为捕获操作，第 2 个红色的(没有处于捕获期间为灰色，不可点击)按钮为停止捕获操作，第 3 个绿色的鲨鱼图标按钮用于重启当前的捕获。接着的 4 个按钮为用于捕获文件上的操作。第 1 个按钮用于打开捕获的文件，第 2 个按钮用于保存捕获文件，第 3 个按钮关闭当前的捕获文件，第 4 个按钮用于重载捕获文件。剩下的按钮以后再讨论。

- **过滤框**

过滤框用于过滤分组列表中的信息。如，输入 http 可以只保留协议为 HTTP 的分组。

- **分组列表**

分组列表显示了捕获到的每一个分组的摘要，包含了分组的序号(由 wireshark 指定的序号)，分组捕获时的时间，源地址和目的地址，协议类型，以及协议中的简要内容信息。

- **已选分组信息窗**

这个窗口提供了选择的分组(高亮)的详细信息。这些信息包含了以太网帧(或 WLAN 帧)，IP 数据报，TCP(或 UDP)报文段，以及应用层的报文(如果使用了应用层协议)。这些信息可以点击箭头展开和收起。

- **分组详细内容窗**

这个窗口提供了帧的内容。有 2 种方式：16 进制以及 ASCII 方式。

3. **用 wireshark 做一次测试**

现在我们使用 wireshark 做一次测试。我们假设你的电脑使用了以太网接入或 WiFi 接入。

执行以下的步骤：

- 打开你的浏览器，新建一个 Tab。
- 打开 wireshark，选择你电脑的网络接口开始捕获。
- 此时，切换到浏览器的 Tab，键入 http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html 等待网页加载完成。
- 在 wireshark 中，使用过滤框，键入 http，就可以发现我们刚才的浏览器活动。
- 选择 HTTP 请求分组，在已选分组信息窗，找到 HTTP 协议，其中就有请求行中的 GET 方法字段。

如下图所示：