1

INTRODUCTION

Each of the past three centuries was dominated by a single new technology. The 18th century was the era of the great mechanical systems accompanying the Industrial Revolution. The 19th century was the age of the steam engine. During the 20th century, the key technology was information gathering, processing, and distribution. Among other developments, we saw the deployment of worldwide telephone networks, the invention of radio and television, the birth and unprecedented growth of the computer industry, the launching of communication satellites, and, of course, the Internet. Who knows what miracles the 21st century will bring?

As a result of this rapid technological progress, these areas are rapidly converging in the 21st century, and the differences between collecting, transporting, storing, and processing information are quickly disappearing. Organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As our ability to gather, process, and distribute information grows, the demand for more sophisticated information processing grows even faster.

1.1 USES OF COMPUTER NETWORKS

Although the computing industry is still young compared to other technical industries such as automobiles and air transportation, computers have made spectacular progress in a short time. During the first two decades of their existence,

computer systems were highly centralized, usually within a single room. Often, this room had glass windows, through which visitors could gawk at the great electronic wonder inside. A medium-sized company or university might have had one or two computers, while large institutions had at most a few dozen. The idea that within fifty years vastly more powerful computers smaller than postage stamps would be mass produced by the billions was science fiction.

The convergence of computers and communications has had a profound influence on the organization of computer systems. The once-dominant concept of the ''computer center'' as a room with a single large computer to which users bring their work for processing is now obsolete (although data centers holding hundreds of thousands of Internet servers are common). The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called **computer networks**. The design and organization of these networks are the subjects of this book.

Throughout the book, we will use the term ''computer network'' to mean a collection of interconnected, autonomous computing devices. Two computers are said to be interconnected if they can exchange information. Interconnection can take place over a variety of transmission media including copper wire, fiber optic cable, and radio waves (e.g., microwave, infrared, communication satellites). Networks come in many sizes, shapes, and forms, as we will explore throughout the book. They are usually connected to make larger networks, with the **Internet** being the most well-known example of a network of networks.

1.1.1 Access to Information

Access to information comes in many forms. A common method of accessing information via the Internet is using a Web browser, which allows a user to retrieve information from various Web sites, including increasingly popular social media sites. Mobile applications on smartphones now also allow users to access remote information. Topics include the arts, business, cooking, government, health, history, hobbies, recreation, science, sports, travel, and many others. Fun comes in too many ways to mention, plus some ways that are better left unmentioned.

News organizations have largely migrated online, with some even ceasing print operations entirely. Access to information, including the news, is increasingly personalizable. Some online publications even allow you to tell them that you are interested in corrupt politicians, big fires, scandals involving celebrities, and epidemics, but no football, thank you. This trend certainly threatens the employment of 12-year-old paperboys, but online distribution has allowed the distribution of news to reach far larger and broader audiences.

Increasingly, news is also being curated by social media platforms, where users can post and share news content from a variety of sources, and where the news that any given user sees is prioritized and personalized based on both explicit user

preferences and complex machine learning algorithms that predict user preferences based on the user's history. Online publishing and content curation on social media platforms supports a funding model that depends largely on highly targeted behavioral advertising, which necessarily implies gathering data about the behavior of individual users. This information has sometimes been misused.

Online digital libraries and retail sites now host digital versions of content ranging from academic journals to books. Many professional organizations, such as the ACM (*www.acm.org*) and the IEEE Computer Society (*www.computer.org*), already have all their journals and conference proceedings online. Electronic book readers and online libraries may someday make printed books obsolete. Skeptics should take note of the effect the printing press had on the medieval illuminated manuscript.

Much information on the Internet is accessed using a client-server model, where a client explicitly requests information from a server that hosts that information, as illustrated in Fig. 1-1.

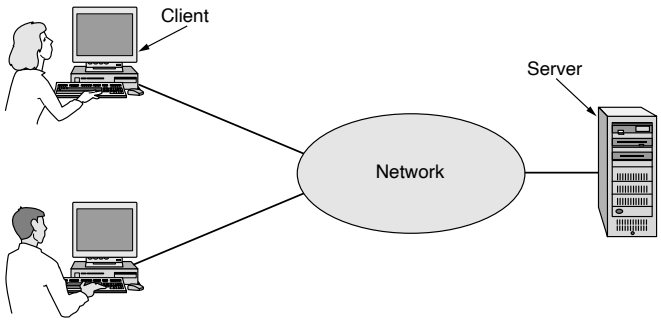

Figure 1-1. A network with two clients and one server.

The **client-server model** is widely used and forms the basis of much network usage. The most popular realization is that of a **Web application**, where a server generates Web pages based on its database in response to client requests that may update the database. The client-server model is applicable not only when the client and server are both in the same building (and belong to the same company), but also when they are far apart. For example, when a person at home accesses a page on the World Wide Web, the same model is employed, with the remote Web server being the server and the user's personal computer being the client. Under most conditions, one server can handle a large number (hundreds or thousands) of clients simultaneously.

If we look at the client-server model, to a first approximation we see that two processes (running programs) are involved, one on the client machine and one on the server machine. Communication takes the form of the client process sending a

message over the network to the server process. The client process then waits for a reply message. When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply. These messages are shown in Fig. 1-2.

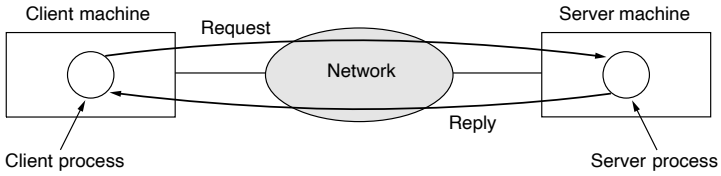

Figure 1-2. The client-server model involves requests and replies.

Another popular model for accessing information is **peer-to-peer** communication (Parameswaran et al., 2001). In this form, individuals who form a loose group can communicate with others in the group, as shown in Fig. 1-3. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients and servers.

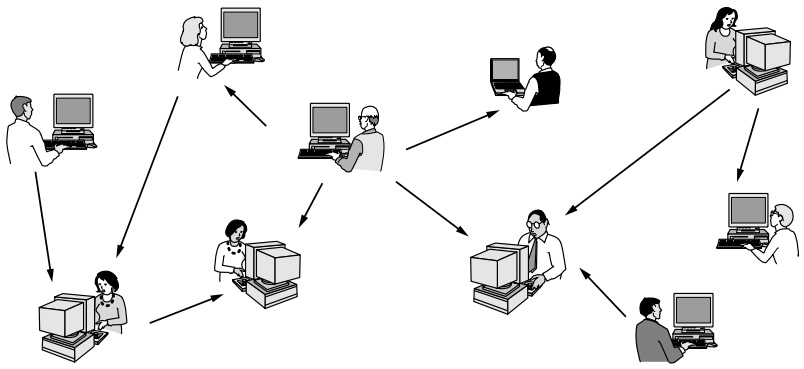

Figure 1-3. In a peer-to-peer system, there are no fixed clients and servers.

Many peer-to-peer systems, such as BitTorrent (Cohen, 2003), do not have a central database of content. Instead, each user maintains a local database of content, as well as a list of other members of the system. A new user can then go to any existing member to see what he has and get the names of other members to inspect for more content and more names. This lookup process can be repeated indefinitely to build up a large local database of what is out there. It is an activity that would get tedious for people, but computers excel at it.

Peer-to-peer communication is often used to share music and videos. It really hit the big time around 2000 with a music sharing service called Napster, which was shut down after a monumental copyright infringement case (Lam and Tan, 2001; and Macedonia, 2000). Legal applications for peer-to-peer communication now exist. These include fans sharing public domain music, families sharing photos and movies, and users downloading public software packages. In fact, one of the most popular Internet applications of all, email, is (conceptually) peer-to-peer. This form of communication is likely to grow considerably in the future.

1.1.2 Person-to-Person Communication

Person-to-person communication is the 21st century's answer to the 19th century's telephone. Email is already used on a daily basis by millions of people all over the world and its use is growing rapidly. It already routinely contains audio and video as well as text and pictures. Smell may take a while.

Many Internet users now rely on some form of **instant messaging** to communicate with other people on the Internet. This facility, derived from the UNIX *talk* program in use since around 1970, allows two people to type messages at each other in real time. There are also multi-person messaging services too, such as the **Twitter** service, which lets people send short messages (possibly including video) called ''tweets'' to their circle of friends or other followers or the whole world.

The Internet can be used by applications to carry audio (e.g., Internet radio stations, streaming music services) and video (e.g., Netflix, YouTube). Besides being an inexpensive way to communicate with your distant friends, these applications can provide rich experiences such as distance learning, meaning attending 8 A.M. classes without the inconvenience of having to get out of bed first. In the long run, the use of networks to enhance human-to-human communication may prove more important than any of the others. It may become hugely important to people who are geographically challenged, giving them the same access to services as people living in the middle of a big city.

Between person-to-person communications and accessing information are **social network** applications. In these applications, the flow of information is driven by the relationships that people declare between each other. One of the most popular social networking sites is **Facebook**. It lets people create and update their personal profiles and shares the updates with other people who they have declared to be their friends. Other social networking applications can make introductions via friends of friends, send news messages to friends, such as Twitter above, and much more.

Even more loosely, groups of people can work together to create content. A **wiki**, for example, is a collaborative Web site that the members of a community edit. The most famous wiki is the **Wikipedia**, an encyclopedia anyone can read or edit, but there are thousands of other wikis.

1.1.3 Electronic Commerce

Online shopping is already popular; users can browse the online catalogs of thousands of companies and have products shipped right to their doorsteps. After the customer buys a product electronically but cannot figure out how to use it, online technical support may be consulted.

Another area in which e-commerce is widely used is access to financial institutions. Many people already pay their bills, manage their bank accounts, and even handle their investments electronically. Financial technology or ''fintech'' applications allow users to conduct a wide variety of financial transactions online, including transferring money between bank accounts, or even between friends.

Online auctions of second-hand goods have become a massive industry. Unlike traditional e-commerce, which follows the client-server model, online auctions are peer-to-peer in the sense that consumers can act as both buyers and sellers, although there is a central server that holds the database of products for sale.

Some of these forms of e-commerce have acquired cute little tags based on the fact that ''to'' and ''2'' are pronounced the same. The most popular ones are listed in Fig. 1-4.

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books online
B2B	Business-to-business	Car manufacturer ordering tires from a supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products online
P2P	Peer-to-peer	Music or file sharing; Skype

Figure 1-4. Some forms of e-commerce.

1.1.4 Entertainment

Our fourth category is entertainment. This has made huge strides in the home in recent years, with the distribution of music, radio and television programs, and movies over the Internet beginning to rival that of traditional mechanisms. Users can find, buy, and download MP3 songs and high-definition movies and add them to their personal collection. TV shows now reach many homes via **IPTV** (**IP Television**) systems that are based on IP technology instead of cable TV or radio transmissions. Media streaming applications let users tune to Internet radio stations or watch recent episodes of their favorite TV shows or movies. Naturally, all of this content can be moved around your house between different devices, displays, and speakers, usually via a wireless network.

Soon, it may be possible to search for any movie or television program ever made, in any country, and have it be displayed on your screen instantly. New films

may become interactive, where the user is occasionally prompted for the story direction (should Macbeth murder the king or just bide his time?) with alternative scenarios provided for all cases. Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants, and so on.

Another form of entertainment is game playing. Already we have multi-person real-time simulation games, like hide-and-seek in a virtual dungeon, and flight simulators with the players on one team trying to shoot down the players on the opposing team. Virtual worlds provide a persistent setting in which thousands of users can experience a shared reality with three-dimensional graphics.

1.1.5 The Internet of Things

Ubiquitous computing entails computing that is embedded in everyday life, as in the vision of Mark Weiser (1991). Many homes are already wired with security systems that include door and window sensors. Also, there are many more sensors that can be folded into a smart home monitor, such as energy consumption. Smart electricity, gas, and water meters report usage over the network. This functionality saves the company money as there is then no need to send people to read the meters. Smoke detectors can call the fire department instead of just making a big noise (which has little value if no one is home). Smart refrigerators could order more milk when it is almost gone. As the cost of sensing and communication drops, more and more measurement and reporting will be done with networks. This ongoing revolution, often referred to as the **IoT** (**Internet of Things**), is poised to connect just about every electronic device we purchase to the Internet.

Increasingly, consumer electronic devices are networked. For example, some high-end cameras already have a wireless network capability and use it to send photos to a nearby display for viewing. Professional sports photographers can also send their photos to their editors in real-time, first wirelessly to an access point then over the Internet. Devices such as televisions that plug into the wall can use **power-line networks** to send information throughout the house over the wires that carry electricity. It may not be very surprising to have these objects on the network, but objects that we do not think of as computers may sense and communicate information too. For example, your shower may record water usage, give you visual feedback while you lather up, and report to a home environmental monitoring application when you are done to help save on your water bill.

1.2 TYPES OF COMPUTER NETWORKS

There are many distinct types of computer networks. This section provides an overview of a few of these networks, including those we commonly use to access the Internet (mobile and broadband access networks); those that house the data and

applications we use every day (data-center networks); those that connect access networks to data centers (transit networks); and those that we use on a campus, office building, or other organization (enterprise networks).

1.2.1 Broadband Access Networks

In 1977, Ken Olsen was president of the Digital Equipment Corporation, then the number two computer vendor in the world (after IBM). When asked why Digital was not going after the personal computer market in a big way, he said: ''There is no reason for any individual to have a computer in his home.'' History showed otherwise and Digital no longer exists. People initially bought computers for word processing and games. Now the prevailing reason to buy a home computer is to get Internet access. Also, many consumer electronic devices, such as set-top boxes, game consoles, television sets, and even door locks, come with embedded computers that access computer networks, especially wireless networks. Home networks are broadly used for entertainment, including listening to, looking at, and creating music, photos, and videos.

Internet access provides home users with **connectivity** to remote computers. As with companies, home users can access information, communicate with other people, and buy products and services. The main benefit now comes from connecting these devices to other destinations outside of the home. Bob Metcalfe, the inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the number of users because this is roughly the number of different connections that may be made (Gilder, 1993). This hypothesis is known as ''Metcalfe's law.'' It helps to explain how the tremendous popularity of the Internet comes from its size.

Today, broadband access networks are proliferating. In many parts of the world, broadband access is delivered to homes through copper (e.g., telephone lines), coaxial cable (e.g., cable), or optical fiber. The speeds of broadband Internet access continue to increase as well, with many broadband access providers in developed countries delivering a gigabit per second to individual homes. In some parts of the world, particularly in developing regions, the predominant mode of Internet access is mobile.

1.2.2 Mobile and Wireless Access Networks

Mobile computers, such as laptops, tablets, and smartphones, are one of the fastest-growing segments of the computer industry. Their sales have already overtaken those of desktop computers. Why would anyone want one? People on the go often want to use their mobile devices to read and send email, tweet, watch movies, download music, play games, look at maps, or simply to surf the Web for information or fun. They want to do all of the things they do at home and in the office. Naturally, they want to do them from anywhere on land, sea, or in the air.

Connectivity to the Internet enables many of these mobile uses. Since having a wired connection is impossible in cars, boats, and airplanes, there is a lot of interest in wireless networks. Cellular networks operated by telephone companies are one familiar kind of wireless network that blankets us with coverage for mobile phones. Wireless **hotspots** based on the 802.11 standard are another kind of wireless network for mobile computers and portable devices such as phones and tablets. They have sprung up everywhere that people go, resulting in a patchwork of coverage at cafes, hotels, airports, schools, trains, and planes. Anyone with a mobile device and a wireless modem can just turn on their computer and be connected to the Internet through the hotspot as though the computer were plugged into a wired network.

Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, and repair-persons for keeping in contact with their home base. For example, in many cities, taxi drivers are independent businessmen, rather than being employees of a taxi company. In some of these cities, the taxis have a display the driver can see. When a customer calls up, a central dispatcher types in the pickup and destination points. This information is displayed on the drivers' displays and a beep sounds. The first driver to hit a button on the display gets the call. The rise of mobile and wireless networking has also led to a revolution in ground transportation itself, with the ''sharing economy'' allowing drivers to use their on phones as a dispatch device, as with ride-sharing companies such as Uber and Lyft.

Wireless networks are also important to the military. If you have to be able to fight a war anywhere on Earth at short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own.

Although wireless networking and mobile computing are often related, they are not identical, as Fig. 1-5 shows. Here, we see a distinction between **fixed wireless** and **mobile wireless** networks. Even notebook computers are sometimes wired. For example, if a traveler plugs a laptop computer into the wired network jack in a hotel room, he has mobility without a wireless network. The growing pervasiveness of wireless networks is making this situation increasingly rare, although for high performance, wired networks are always better.

Wireless	Mobile	Typical applications
No	No	Desktop computers in offices
No	Yes	A laptop computer used in a hotel room
Yes	No	Networks in unwired buildings
Yes	Yes	Store inventory with a handheld computer

Figure 1-5. Combinations of wireless networks and mobile computing.

Conversely, some wireless computers are not mobile. In people's homes, and in offices or hotels that lack suitable cabling, it can be more convenient to connect desktop computers or media players wirelessly than to install wires. Installing a

wireless network may require simply buying a small box with some electronics in it, unpacking it, and plugging it in. This solution may be far cheaper than having workmen put in cable ducts to wire the building.

Finally, there are also true mobile, wireless applications, such as people walking around stores with handheld computers recording inventory. At many busy airports, car rental return clerks work in the parking lot with wireless mobile computers. They scan the barcodes or RFID chips of returning cars, and their mobile device, which has a built-in printer, calls the main computer, gets the rental information, and prints out the bill on the spot.

A key driver of mobile, wireless applications is the mobile phone. The convergence between telephones and the Internet is accelerating the growth of mobile applications. **Smartphones**, such as Apple's iPhone and Samsung's Galaxy, combine aspects of mobile phones and mobile computers. These phones connect to wireless hotspots, too, and automatically switch between networks to choose the best option for the user. **Text messaging** or **texting** (or **Short Message Service** as it is known outside the U.S.) over the cellular network was tremendously popular at its outset. It lets a mobile phone user type a short message that is then delivered by the cellular network to another mobile subscriber. Texting is extremely profitable since it costs the carrier but a tiny fraction of one cent to relay a text message, a service for which it charges far more. Typing short text messages on mobile phones was, for a time, an immense money maker for mobile carriers. Now, many alternatives that use either the phone's cellular data plan or wireless network, including WhatsApp, Signal, and Facebook Messenger, have overtaken SMS.

Other consumer electronics devices can also use cellular and hotspot networks to stay connected to remote computers. Tablets and electronic book readers can download a newly purchased book or the next edition of a magazine or today's newspaper wherever they roam. Electronic picture frames can update their displays on cue with fresh images.

Mobile phones typically know their own locations. **GPS** (**Global Positioning System**) can directly locate a device, and mobile phones often also triangulate between Wi-Fi hotspots with known locations to determine their location. Some applications are location-dependent. Mobile maps and directions are an obvious candidate as your GPS-enabled phone and car probably have a better idea of where you are than you do. So, too, are searches for a nearby bookstore or Chinese restaurant, or a local weather forecast. Other services may record location, such as annotating photos and videos with the place at which they were made. This annotation is known as **geo-tagging**.

Mobile phones are being increasingly used in **m-commerce** (**mobile-commerce**) (Senn, 2000). Short text messages from the mobile are used to authorize payments for food in vending machines, movie tickets, and other small items instead of cash and credit cards. The charge then appears on the mobile phone bill. When equipped with **NFC** (**Near Field Communication**), technology the mobile can act as an RFID smartcard and interact with a nearby reader for payment. The

driving forces behind this phenomenon are the mobile device makers and network operators, who are trying hard to figure out how to get a piece of the e-commerce pie. From the store's point of view, this scheme may save them most of the credit card company's fee, which can be several percent. Of course, this plan may backfire, since customers in a store might use the RFID or barcode readers on their mobile devices to check out competitors' prices before buying and use them to get a detailed report on where else an item can be purchased nearby and at what price.

One huge thing that m-commerce has going for it is that mobile phone users are accustomed to paying for everything (in contrast to Internet users, who expect everything to be free). If an Internet Web site charged a fee to allow its customers to pay by credit card, there would be an immense bellowing from the users. If, however, a mobile phone operator let its customers pay for items in a store by waving the phone at the cash register and then tacks on a small fee for this convenience, it would probably be accepted as normal. Time will tell.

The uses of mobile and wireless computers will grow rapidly in the future as the size of computers shrinks, probably in ways no one can now foresee. Let us take a quick look at some possibilities. **Sensor networks** have nodes that gather and relay information they sense about the state of the physical world. The nodes may be embedded in familiar devices such as cars or phones, or they may be small separate devices. For example, your car might gather data on its location, speed, vibration, and fuel efficiency from its on-board diagnostic system and upload this information to a database (Hull et al., 2006). Those data can help find potholes, plan trips around congested roads, and tell you if you are a ''gas guzzler'' compared to other drivers on the same stretch of road.

Sensor networks are revolutionizing science by providing a wealth of data on behavior that could not previously be observed. One example is tracking the migration of individual zebras by placing a small sensor on each animal (Juang et al., 2002). Researchers have packed a wireless computer into a single square cubic millimeter (Warneke et al., 2001). With mobile computers this small, even small birds, rodents, and insects can be tracked.

Wireless parking meters can accept credit or debit card payments with instant verification over the wireless link. They can also report when they are in use, which can let drivers download a recent parking map to their car so they can find an available spot more easily. Of course, when a meter expires, it might also check for the presence of a car (by bouncing a signal off it) and report the expiration to parking enforcement. It has been estimated that city governments in the U.S. alone could collect an additional $10 billion this way (Harte et al., 2000).

1.2.3 Content Provider Networks

Many Internet services are now served from ''the cloud,'' or a **data-center network**. Modern data center networks have hundreds of thousands or millions of servers in a single location, usually in a very dense configuration of rows of racks

in buildings that can be more than a kilometer long. Data center networks serve the increasingly growing demands of **cloud computing** and are designed to move large amounts of data between servers in the data center, as well as between the data center and the rest of the Internet.

Today, many of the applications and services you use, ranging from the Web sites you visit to the cloud-based document editor you use to take notes, store data in a data center network. Data center networks face challenges of scale, both for network throughput and for energy usage. One of the main network throughput challenges is the so-called ''cross-section bandwidth,'' which is the data rate that can be delivered between any two servers in the network. Early data-center network designs were based on a simple tree topology, with three layers of switches: access, aggregate, and core; this simple design did not scale well, and was also to be subject to faults.

Many popular Internet services need to deliver content to users around the world. To do so, many sites and services on the Internet use a **CDN** (**Content Delivery Network**). A CDN is a large collection of servers that are geographically distributed in such a way that content is placed as close as possible to the users that are requesting it. Large content providers such as Google, Facebook, and Netflix operate their own CDNs. Some CDNs, such as Akamai and Cloudflare, offer hosting services to smaller services that do not have their own CDN.

Content that users want to access, ranging from static files to streaming video, may be replicated in many locations across a single CDN. When a user requests content, the CDN must decide which replica it should serve to that user. This process must consider the distance from each replica to the client, the load on each CDN server, and traffic load and congestion on the network itself.

1.2.4 Transit Networks

Internet travels over many independently operated networks. The network run by your Internet service provider is typically not the same network as the one that hosts the content for the Web sites that you commonly visit. Typically, content and applications are hosted in data-center networks, and you may be accessing that content from an access network. Content must thus traverse the Internet from the data center to the access network, and ultimately to your device.

When the content provider and your **ISP** (**Internet Service Provider**) are not directly connected, they often rely on a **transit network** to carry the traffic between them. Transit networks typically charge both the ISP and the content provider for carrying traffic from end-to-end. If the network hosting the content and the access network exchange enough traffic between them, they may decide to interconnect directly. One example where direct interconnection is common is between large ISPs and large content providers, such as Google or Netflix. In these cases, the ISP and the content provider must build and maintain network infrastructure to facilitate interconnecting directly, often in many geographic locations.

Transit networks are traditionally called **backbone networks** because they have had the role of carrying traffic between two endpoints. Many years ago, transit networks were hugely profitable because every other network would rely on them (and pay them) to connect to the rest of the Internet.

The last decade, however, has witnessed two trends. The first trend is the consolidation of content in a handful of large content providers, spawned by the proliferation of cloud-hosted services and large content delivery networks. The second trend is the expansion of the footprint of individual access ISP networks: whereas access ISPs may have once been small and regional, many access ISPs have national (or even international) footprints, which has increased both the range of geographic locations where they can connect to other networks as well as their subscriber base. As the size (and negotiating power) of the access networks and the content provider networks continues to increase, the larger networks have come to rely less on transit networks to deliver their traffic, preferring often to directly interconnect and rely on the transit network only as a backup.

1.2.5 Enterprise Networks

Most organizations (e.g., companies, universities) have many computers. Each employee may use a computer to perform tasks ranging from product design to payroll. In the common case, these machines are connected on a common network, which allows the employees to share data, information, and compute resources with one another.

Resource sharing makes programs, equipment, and especially data available to other users on the network without regard to the physical location of the resource or the user. One widespread example is having a group of office workers share a common printer. Many employees do not need a private printer and a high-volume networked printer is often less expensive, faster, and easier to maintain than a large collection of individual printers.

Probably, even more important than sharing physical resources such as printers and backup systems is sharing information. Most companies have customer records, product information, inventories, financial statements, tax information, and much more online. If all of its computers suddenly went down, a bank could not last more than five minutes. A modern manufacturing plant, with a computer-controlled assembly line, would not last even five seconds. Even a small travel agency or three-person law firm is now highly dependent on computer networks for allowing employees to access relevant information and documents instantly.

For smaller companies, the computers may be located in a single office even a single building; in the case of larger companies, the computers and employees may be scattered over dozens of offices and plants in many countries. Nevertheless, a salesperson in New York might sometimes need access to a product inventory database in Singapore. Networks called **VPNs** (**Virtual Private Networks**) connect

the individual networks at different sites into one logical network. In other words, the mere fact that a user happens to be 15,000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the ''tyranny of geography.''

In the simplest of terms, one can imagine a company's information system as consisting of one or more databases with company information and some number of employees who need to access them remotely. In this model, the data are stored on powerful computers called **servers**. Often, these are centrally housed and maintained by a system administrator. In contrast, the employees have simpler machines, called **clients**, on their desks, with which they access remote data, for example, to include in spreadsheets they are constructing. (Sometimes we will refer to the human user of the client machine as the ''client,'' but it should be clear from the context whether we mean the computer or its user.) The client and server machines are connected by a network, as illustrated in Fig. 1-1. Note that we have shown the network as a simple oval, without any detail. We will use this form when we mean a network in the most abstract sense. When more detail is required, it will be provided.

A second goal of setting up an enterprise computer network has to do with people rather than information or even computers. A computer network can provide a powerful **communication medium** among employees. Virtually every company that has two or more computers now has **email** (**electronic mail**), which employees generally use for a great deal of daily communication. In fact, a common gripe around the water cooler is how much email everyone has to deal with, much of it quite meaningless because bosses have discovered that they can send the same (often content-free) message to all their subordinates at the push of a button.

Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **VoIP** (**Voice over IP**) when Internet technology is used. The microphone and speaker at each end may belong to a VoIP-enabled phone or the employee's computer. Companies find this a wonderful way to save on their telephone bills.

Other, much richer forms of communication are made possible by computer networks. Video can be added to audio so that multiple employees at distant locations can see and hear each other as they hold a meeting. This technique is a powerful tool for eliminating the cost and time previously devoted to travel. **Desktop sharing** lets remote workers see and interact with a graphical computer screen. This makes it easy for two or more people who work far apart to read and write a shared blackboard or write a report together. When one worker makes a change to an online document, the others can see the change immediately, instead of waiting several days for a letter. Such a speedup makes cooperation among far-flung groups of people easy where it previously had been impossible. More ambitious forms of remote coordination such as telemedicine are only now starting to be used (e.g., remote patient monitoring) but may become much more important. It is

sometimes said that communication and transportation are having a race, and whichever wins will make the other obsolete.

A third goal for many companies is doing business electronically, especially with customers and also suppliers. Airlines, bookstores, and other retailers have discovered that many customers like the convenience of shopping from home. Consequently, many companies provide catalogs of their goods and services online and take orders online. Manufacturers of automobiles, aircraft, and computers, among others, buy subsystems from many suppliers and then assemble the parts. Using computer networks, manufacturers can place orders electronically as needed. This reduces the need for large inventories and enhances efficiency.

1.3 NETWORK TECHNOLOGY, FROM LOCAL TO GLOBAL

Networks can range from small and personal to large and global. In this section, we explore the various networking technologies that implement networks at different sizes and scales.

1.3.1 Personal Area Networks

PANs (**Personal Area Networks**) let devices communicate over the range of a person. A common example is a wireless network that connects a computer with its peripherals. Other examples include the network that connects your wireless headphones and your watch to your smartphone. It is also often used to connect a headset to a mobile phone without cords, and it can allow your digital music player to connect to your car merely being brought within range.

Almost every computer has an attached monitor, keyboard, mouse, and printer. Without using wireless, this connection must be done with cables. Many new users have so much trouble finding the right cables and plugging them into the right little holes (even though they are usually shape and color coded) that most computer vendors offer the option of sending a technician to the user's home to do it. To help these users, some companies got together to design a short-range wireless network called **Bluetooth** to connect these components without wires. The idea is that if your devices have Bluetooth, then you do not need to deal with cables. You just put them down, turn them on, and they begin communicating. For many people, this ease of operation is a big plus.

In the simplest form, Bluetooth networks use the master-slave paradigm shown in Fig. 1-6. The system unit (the PC) is normally the master, talking to the mouse or keyboard as slaves. The master tells the slaves what addresses to use, when they can transmit, how long they can transmit, what frequencies they can use, and so on. We will discuss Bluetooth in more detail in Chap. 4.

PANs can also be built with a variety of other technologies that communicate over short ranges, as we will discuss in Chap. 4.

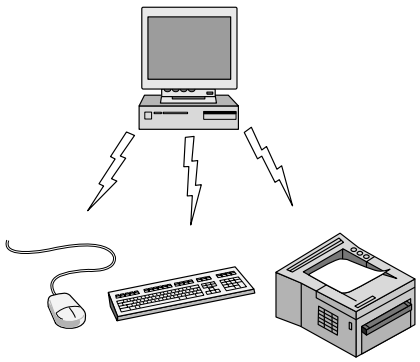

Figure 1-6. Bluetooth PAN configuration.

1.3.2 Local Area Networks

A **LAN** (**Local Area Network**) is a private network that operates within and nearby a single building such as a home, office, or factory. LANs are widely used to connect personal computers and consumer electronics to let them share resources (e.g., printers) and exchange information.

Wireless LANs are pervasive today. They initially gained popularity in homes, older office buildings, cafeterias, and other places where installing cables introduced too much cost. In these systems, every computer has a radio modem and an antenna that it uses to communicate with other computers. In most cases, each computer talks to a device called an **AP** (**Access Point**), **wireless router**, or **base station**, as shown in Fig. 1-7(a). This device relays packets between the wireless computers and also between them and the Internet. Being the AP is like being the popular kid at school because everyone wants to talk to you. Another common scenario entails nearby devices relaying packets for one another in a so-called **mesh network** configuration. In some cases, the relays are the same nodes as the endpoints; more commonly, however, a mesh network will include a separate collection of nodes whose sole responsibility is relaying traffic. Mesh network settings are common in developing regions where deploying connectivity across a region may be cumbersome or costly. They are also becoming increasingly popular for home networks, particularly in large homes.

There is a popular standard for wireless LANs called **IEEE 802.11**, commonly called WiFi . It runs at speeds from 11 Mbps (802.11b) to 7 Gbps (802.11ad). Please note that in this book we will adhere to tradition and measure line speeds in megabits/sec, where 1 Mbps is 1,000,000 bits/sec, and gigabits/sec, where 1 Gbps is 1,000,000,000 bits/sec. Powers of two are used only for storage, where a 1 MB memory is 2^{20} or 1,048,576 bytes. We will discuss 802.11 in Chap. 4.

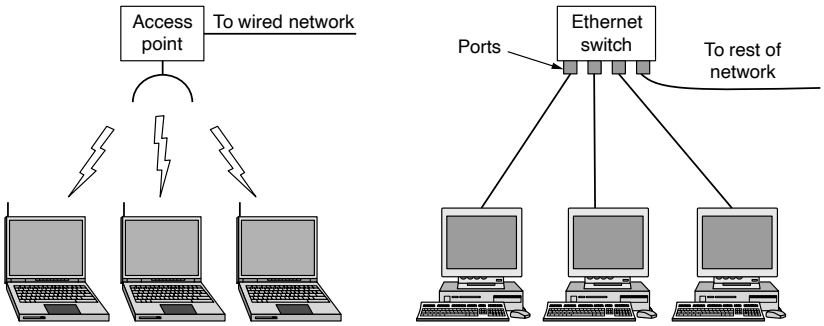

Figure 1-7. Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

Wired LANs use many different transmission technologies; common physical modes of transmission are copper, coaxial cable, and optical fiber. LANs have limited size, which means that the worst-case transmission time is bounded and known in advance. Knowing these bounds helps with the task of designing network protocols. Typically, wired LANs can run at speeds ranging from 100 Mbps to 40 Gbps. They also have low latency (never more than tens of milliseconds, and often much less) and transmission errors are infrequent. Wired LANs typically have lower latency, lower packet loss, and higher throughput than wireless LANs, but over time this performance gap has narrowed. It is far easier to send signals over a wire or through a fiber than through the air.

Many wired LANs comprise point-to-point wired links. IEEE 802.3, popularly called **Ethernet**, is by far the most common type of wired LAN. Fig. 1-7(b) shows an example **switched Ethernet** topology. Each computer speaks the Ethernet protocol and connects to a device called a **switch** with a point-to-point link. The job of the switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.

A switch has multiple **ports**, each of which can connect to one other device, such as a computer or even another switch. To build larger LANs, switches can be plugged into each other using their ports. What happens if you plug them together in a loop? Will the network still work? Luckily, someone thought of this case, and now all switches in the world use her anti-looping algorithm (Perlman, 1985). It is the job of the protocol to sort out what paths packets should travel to safely reach the intended computer. We will see how this works in Chap. 4.

It is also possible to divide one large physical LAN into two smaller logical LANs. You might wonder why this would be useful. Sometimes, the layout of the network equipment does not match the organization's structure. For example, the engineering and finance departments of a company might have computers on the same physical LAN because they are in the same wing of the building, but it might be easier to manage the system if engineering and finance logically each had its

own network **VLAN** (**Virtual LAN**). In this design, each port is tagged with a ''color,'' say green for engineering and red for finance. The switch then forwards packets so that computers attached to the green ports are separated from the computers attached to the red ports. Broadcast packets sent on a red port, for example, will not be received on a green port, just as though there were two separate physical LANs. We will cover VLANs at the end of Chap. 4.

There are other wired LAN topologies, too. In fact, switched Ethernet is a modern version of the original Ethernet design that broadcasts all packets over a single linear cable. At most one machine could successfully transmit at a time, and a distributed arbitration mechanism was used to resolve conflicts. It used a simple algorithm: computers could transmit whenever the cable was idle. If two or more packets collided, each computer just waited a random time and tried later. We will call that version **classic Ethernet** for clarity, and as you no doubt suspected, you will learn about it in Chap. 4.

Both wireless and wired broadcast LANs can allocate resources statically or dynamically. A typical static allocation would be to divide time into discrete intervals and use a round-robin algorithm, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to transmit or receive during its allocated slot, so most systems attempt to allocate the channel dynamically (i.e., on demand).

Dynamic allocation methods for a common channel are either centralized or decentralized. In a centralized channel allocation method, there is a single entity, for example, the base station in cellular networks, which determines who goes next. It might do so by accepting multiple packets and prioritizing them according to some internal algorithm. In a decentralized channel allocation method, there is no central entity; each machine must decide for itself whether to transmit. You might think that this approach would lead to chaos, but later we will study many algorithms designed to bring order out of the potential chaos—provided, of course, that all the machines obey the rules.

1.3.3 Home Networks

It is worth giving specific attention to LANs in the home, or **home networks**. Home networks are a type of LAN; they may have a broad, diverse range of Internet-connected devices, and must be particularly easy to manage, dependable, and secure, especially in the hands of nontechnical users.

Many years ago, a home network would probably have consisted of a few laptops on a wireless LAN. Today, a home network may include devices such as smartphones, wireless printers, thermostats, burglar alarms, smoke detectors, lightbulbs, cameras, televisions, stereos, smart speakers, refrigerators, and so on. The proliferation of Internet-connected appliances and consumer electronics, often called the Internet of things, makes it possible to connect just about any electronic

device (including sensors of many types) to the Internet. This huge scale and diversity of Internet connected devices introduces new challenges for designing, managing, and securing a home network. Remote monitoring of the home is becoming increasingly common, with applications ranging from security monitoring to maintenance to aging in place, as many grown children are willing to spend some money to help their aging parents live safely in their own homes.

Although the home network is just another LAN, in practice it is likely to have different properties than other LANs, for several reasons. First, the devices that people connect to their home network need to be easy to install and maintain. Wireless routers were at one point very commonly returned to stores because people bought them expecting to have a wireless network work ''out of the box'' but instead found themselves confronted with the prospect of many calls to technical support. The devices need to be foolproof and work without requiring the user to read and fully understand a 50-page manual.

Second, security and reliability have higher stakes because insecurity of the devices may introduce direct threats to consumer health and safety. Losing a few files to an email virus is one thing; having a burglar disarm your security system from his phone and then plunder your house is something quite different. The past few years have seen countless examples of insecure or malfunctioning IoT devices that have resulted in everything from frozen pipes to remote control of devices through malicious third-party scripts. The lack of serious security on many of these devices has made it possible for an eavesdropper to observe details about user activity in the home; even when the contents of the communication are encrypted, simply knowing the type of device that is communicating and the volumes and times of traffic can reveal a lot about private user behavior.

Third, home networks evolve organically, as people buy various consumer electronics devices and connect them to the network. As a result, in contrast to a more homogeneous enterprise LAN, the set of technologies connected to the home network may be significantly more diverse. Yet, despite this diversity, people expect these devices to be able to interact (e.g., they want to be able to use the voice assistant manufactured by one vendor to control the lights from another vendor). Once installed, the devices may remain connected for years (or decades). This means no interface wars: Telling consumers to buy peripherals with IEEE 1394 (FireWire) interfaces and a few years later retracting that and saying USB 3.0 is the interface-of-the-month and then switching that to 802.11g—oops, no, make that 802.11n—no wait, 802.11ac—sorry, we mean 802.11ax, is not tenable.

Finally, profit margins are small in consumer electronics, so many devices aim to be as inexpensive as possible. When confronted with a choice about which Internet-connected digital photo frame to buy, many users may opt for the less-expensive one. The pressure to reduce consumer device costs makes achieving the above goals even more difficult. Security, reliability, and interoperability all ultimately cost money. In some cases, manufacturers or consumers may need powerful incentives to make and stick to recognized standards.

Home networks typically operate over wireless networks. Convenience and cost favors wireless networking because there are no wires to fit, or worse, retrofit. As Internet-connected devices proliferate, it becomes increasingly inconvenient to drop a wired network port everywhere in the home where there is a power outlet. Wireless networks are more convenient and more cost-effective. Reliance on wireless networks in the home, however, does introduce unique performance and security challenges. First, as users exchange more traffic on their home networks and connect more devices to them, the home wireless network is increasingly becoming a performance bottleneck. When the home network is performing poorly, a common pastime is to blame the ISP for the poor performance. ISPs tend not to like this so much.

Second, wireless radio waves can travel through walls (in the popular 2.4 GHz band, but less so at 5 GHz). Although wireless security has improved substantially over the last decade, it still has been subject to many attacks that allow eavesdropping, and certain aspects of the traffic, such as device hardware addresses and traffic volume, remain unencrypted. In Chap. 8, we will study how encryption can be used to provide security, but it is easier said than done with inexperienced users.

Power-line networks can also let devices that plug into outlets broadcast information throughout the house. You have to plug in the TV anyway, and this way it can get Internet connectivity at the same time. These networks carry both power and data signals at the same time; part of the solution is to run these two functions on different frequency bands.

1.3.4 Metropolitan Area Networks

A **MAN** (**Metropolitan Area Network**) covers a city. The best-known examples of MANs are the cable television networks. These systems grew from earlier community antenna systems used in areas with poor over-the-air television reception. In those early systems, a large antenna was placed on top of a nearby hill and a signal was then piped to the subscribers' houses.

At first, these networks were locally designed, ad hoc systems. Then, companies began jumping into the business, getting contracts from local governments to wire up entire cities. The next step was television programming and even entire channels designed for cable only. Often, these channels were highly specialized, such as all news, all sports, all cooking, all gardening, and so on. But from their inception until the late 1990s, they were intended for television reception only.

When the Internet began attracting a mass audience, the cable TV network operators began to realize that with some changes to the system, they could provide two-way Internet service in unused parts of the spectrum. At that point, the cable TV system began to morph from simply a way to distribute television to a metropolitan area network. To a first approximation, a MAN might look something like the system shown in Fig. 1-8. In this figure, we see both television signals and Internet being fed into the centralized **cable head-end**, (or cable modem termination

system) for subsequent distribution to people's homes. We will come back to this subject in detail in Chap. 2.

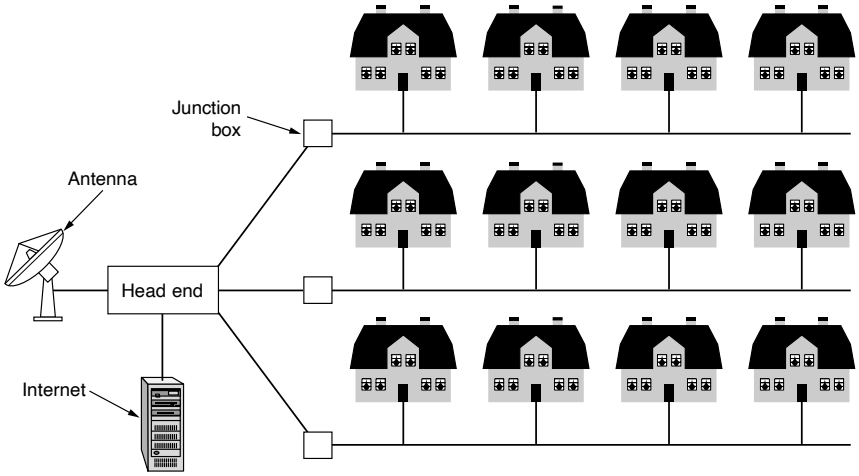

Figure 1-8. A metropolitan area network based on cable TV.

Cable television is not the only MAN. Recent developments in high-speed wireless Internet access have resulted in another MAN, which has been standardized as IEEE 802.16 and is popularly known as **WiMAX**. It does not seem to be catching on, however. Other wireless technologies, **LTE** (**Long Term Evolution**) and 5G, will also be covered there.

1.3.5 Wide Area Networks

A **WAN** (**Wide Area Network**) spans a large geographical area, often a country, a continent, or even multiple continents. A WAN may serve a private organization, as in the case of an enterprise WAN, or it may be a commercial service offering, as in the case of a transit network.

We will begin our discussion with wired WANs, using the example of a company with branch offices in different cities. The WAN in Fig. 1-9 connects offices in Perth, Melbourne, and Brisbane. Each of these offices contains computers intended for running user (i.e., application) programs. We will follow conventional usage and call these machines **hosts**. The rest of the network that connects these hosts is then called the **communication subnet**, or just **subnet** for short. The subnet carries messages from host to host, just as the telephone system carries words (really just sounds) from speaker to listener.

In most WANs, the subnet consists of two distinct components: transmission lines and switching elements. **Transmission lines** move bits between machines.

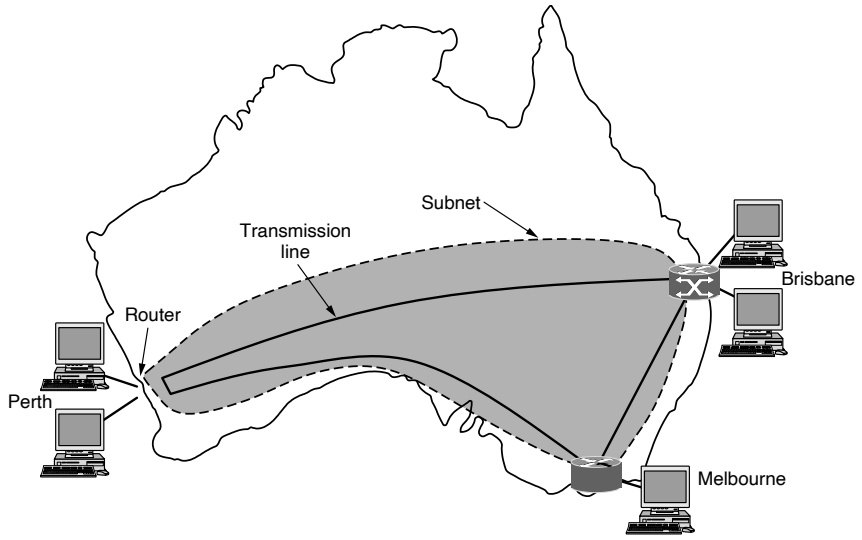

Figure 1-9. WAN that connects three branch offices in Australia.

They can be made of copper wire, coaxial cable, optical fiber, or radio links. Most organizations do not have transmission lines lying about, so instead they use the lines from a telecommunications company. **Switching elements**, or **switches**, are specialized devices that connect two or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line on which to forward them. These switching computers have been called by various names in the past; the name **router** is now most commonly used. Unfortunately, some people pronounce it ''rooter'' while others have it rhyme with ''doubter.'' Determining the correct pronunciation will be left as an exercise for the reader. (Note: the perceived correct answer may depend on where you live.)

In most WANs, the network contains many transmission lines, each connecting a pair of routers. Two routers that do not share a transmission line must do so via other routers. There may be many paths in the network that connect these two routers. How the network makes the decision as to which path to use is called a **routing algorithm**. How each router makes the decision as to where to send a packet next is called a **forwarding algorithm**. We will study some of both types in detail in Chap. 5.

A short comment about the term ''subnet'' is in order here. Originally, its *only* meaning was the collection of routers and communication lines that moved packets from the source host to the destination host. Readers should be aware that it has acquired a second, more recent meaning in conjunction with network addressing.

We will discuss that meaning in Chap. 5 and stick with the original meaning (a collection of lines and routers) until then.

The WAN as we have described it looks similar to a large wired LAN, but there are some important differences that go beyond long wires. Usually in a WAN, the hosts and subnet are owned and operated by different people. In our example, the employees might be responsible for their own computers, while the company's IT department is in charge of the rest of the network. We will see clearer boundaries in the coming examples, in which the network provider or telephone company operates the subnet. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts) greatly simplifies the overall network design.

A second difference is that the routers will usually connect different kinds of networking technology. The networks inside the offices may be switched Ethernet, for example, while the long-distance transmission lines may be SONET links (which we will cover in Chap. 2). Some device needs to join them. The astute reader will notice that this goes beyond our definition of a network. This means that many WANs will in fact be **internetworks**, or composite networks that comprise more than one network. We will have more to say about internetworks in the next section.

A final difference is in what is connected to the subnet. This could be individual computers, as was the case for connecting to LANs, or it could be entire LANs. This is how larger networks are built from smaller ones. As far as the subnet is concerned, it does the same job.

Virtual Private Networks and SD-WANs

Rather than lease dedicated transmission lines, an organization might rely on Internet connectivity to connect its offices. This allows connections to be made between the offices as virtual links that use the underlying capacity of the Internet. As mentioned earlier, this arrangement, shown in Fig. 1-10, is called a virtual private network. In contrast to a network with dedicated physical links, a VPN has the usual advantage of virtualization, which is that it provides flexible reuse of a resource (Internet connectivity). A VPN also has the usual disadvantage of virtualization, which is a lack of control over the underlying resources. With a dedicated line, the capacity is clear. With a VPN, performance may vary with that of the underlying Internet connectivity. The network itself may also be operated by a commercial Internet service provider (ISP). Fig. 1-11 shows this structure, which connects the WAN sites to each other, as well as to the rest of the Internet.

Other kinds of WANs make heavy use of wireless technologies. In satellite systems, each computer on the ground has an antenna through which it can exchange data with a satellite in orbit. All computers can hear the output *from* the satellite, and in some cases, they can also hear the upward transmissions of their

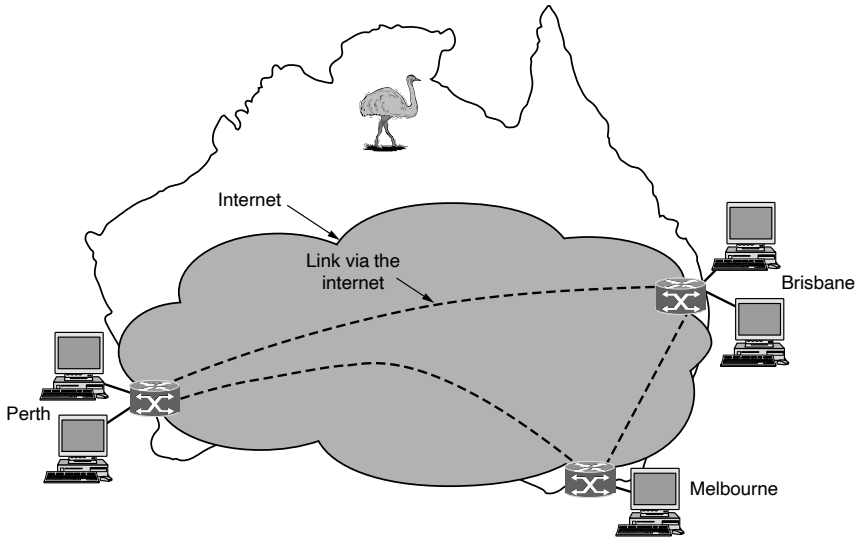

Figure 1-10. WAN using a virtual private network.

fellow computers *to* the satellite as well. Satellite networks are inherently broadcast and are most useful when broadcast is important or no ground-based infrastructure is present (think: oil companies exploring in an isolated desert).

The cellular telephone network is another example of a WAN that uses wireless technology. This system has already gone through five generations. The first generation was analog and for voice only. The second generation was digital and for voice only. The third generation is digital and is for both voice and data. The fourth generation is purely digital, even for voice. The fifth generation is also pure digital and much faster than the fourth, with lower delays as well.

Each cellular base station covers a distance much larger than a wireless LAN, with a range measured in kilometers rather than tens of meters. The base stations are connected to each other by a backbone network that is usually wired. The data rates of cellular networks are often on the order of 100 Mbps, much smaller than a wireless LAN that can range up to on the order of 7 Gbps. We will have a lot to say about these networks in Chap. 2.

More recently, organizations that are distributed across geographic regions and need to connect sites are designing and deploying so-called **software-defined WANs** or **SD-WANs**, which use different, complementary technologies to connect disjoint sites but provide a single **SLA** (**Service-Level Agreement**) across the network. For example, a network might possibly use a combination of more-expensive dedicated leased lines to connect multiple remote locations and complementary,

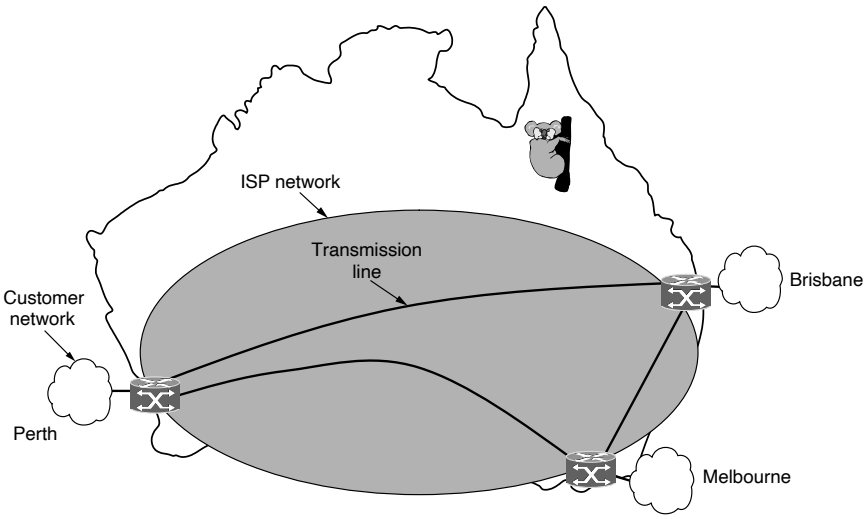

Figure 1-11. WAN using an ISP network.

less-expensive commodity Internet connectivity to connect these locations. Logic written in software reprograms the switching elements in real time to optimize the network for both cost and performance. SD-WANs are one example of an **SDN** (**Software-Defined Network**), a technology that has gained momentum over the last decade and generally describes network architectures that control the network using a combination of programmable switches with control logic implemented as a separate software program.

1.3.6 Internetworks

Many networks exist in the world, and they often use different hardware and software technologies. People connected to one network often want to communicate with people attached to a different one. The fulfillment of this desire requires that different, and frequently incompatible, networks be connected. A collection of interconnected networks is called an **internetwork** or **internet**. We will use these terms in a generic sense, in contrast to the global **Internet** (which is one specific internet), which we will always capitalize. The Internet connects content providers, access networks, enterprise networks, home networks, and many other networks to one another. We will look at the Internet in great detail later in this book.

A network comprises the combination of a subnet and its hosts. However, the word ‘‘network’’ is often used in a loose (and confusing) sense as well. A subnet might be described as a network, as in the case of the ‘‘ISP network’’ of Fig. 1-11.

An internetwork might also be described as a network, as in the case of the WAN in Fig. 1-9. We will follow similar practice, and if we are distinguishing a network from other arrangements, we will stick with our original definition of a collection of computers interconnected by a single technology.

An internet entails the interconnection of distinct, independently operated networks. In our view, connecting a LAN and a WAN or connecting two LANs is the usual way to form an internetwork, but there is little agreement over terminology in this area. Generally speaking, if two or more independently operated networks pay to interconnect, or if two or more networks use fundamentally different underlying technology (e.g., broadcast versus point-to-point and wired versus wireless), we probably have an internetwork.

The device that makes a connection between two or more networks and provides the necessary translation, both in terms of hardware and software, is a **gateway**. Gateways are distinguished by the layer at which they operate in the protocol hierarchy. We will have much more to say about layers and protocol hierarchies in the next section, but for now imagine that higher layers are more tied to applications, such as the Web, and lower layers are more tied to transmission links, such as Ethernet. Because the benefit of forming an internet is to connect computers across networks, we do not want to use too low-level a gateway or we will be unable to make connections between different kinds of networks. We do not want to use too high-level a gateway either, or the connection will only work for particular applications. The level in the middle that is ''just right'' is often called the network layer, and a router is a gateway that switches packets at the network layer. Generally speaking, an internetwork will be connected by network-layer gateways, or routers; however, even a single large network often contains many routers.

1.4 EXAMPLES OF NETWORKS

The subject of computer networking covers many different kinds of networks, large and small, well known and less well known. They have different goals, scales, and technologies. In the following sections, we will look at some examples, to get an idea of the variety one finds in the area of computer networking.

We will start with the Internet, probably the best-known ''network,'' and look at its history, evolution, and technology. Then, we will consider the mobile phone network. Technically, it is quite different from the Internet. Next, we will introduce IEEE 802.11, the dominant standard for wireless LANs.

1.4.1 The Internet

The Internet is a vast collection of different networks that use certain common protocols and provide certain common services. It is an unusual system in that it was not planned by any single organization, and it is not controlled by any single

organization, either. To better understand it, let us start from the beginning and see how it has developed and why. For a wonderful history of how the Internet developed, John Naughton's (2000) book is highly recommended. It is one of those rare books that is not only fun to read but also has 20 pages of *ibid*.'s and *op. cit*.'s for the serious historian. Some of the material in this section is based on this book. For a more recent history, try Brian McCullough's book (2018).

Of course, countless technical books have been written about the Internet, its history, and its protocols as well. For more information, see, for example, Severance (2015).

The ARPANET

The story begins in the late 1950s. At the height of the Cold War, the U.S. DoD (Department of Defense) wanted a command-and-control network that could survive a nuclear war. At that time, all military communications used the public telephone network, which was considered vulnerable. The reason for this belief can be gleaned from Fig. 1-12(a). Here the black dots represent telephone switching offices, each of which was connected to thousands of telephones. These switching offices were, in turn, connected to higher-level switching offices (toll offices), to form a national hierarchy with only a small amount of redundancy. The vulnerability of the system was that the destruction of a few key toll offices could fragment it into many isolated islands so that generals in the Pentagon could not call a base in Los Angeles.

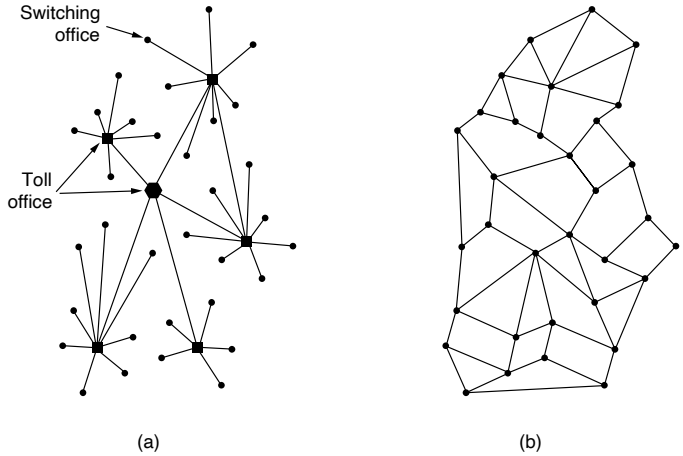

Figure 1-12. (a) Structure of the telephone system. (b) Baran's proposal.

Around 1960, the DoD awarded a contract to the RAND Corporation to find a solution. One of its employees, Paul Baran, came up with the highly distributed

and fault-tolerant design of Fig. 1-12(b). Since the paths between any two switching offices were now much longer than analog signals could travel without distortion, Baran proposed using digital packet-switching technology. Baran wrote several reports for the DoD describing his ideas in detail (Baran, 1964). Officials at the Pentagon liked the concept and asked AT&T, then the U.S.' national telephone monopoly, to build a prototype. AT&T dismissed Baran's ideas out of hand. The biggest and richest corporation in the world was not about to allow some young whippersnapper (out in California, no less—AT&T was then an East Coast company) tell it how to build a telephone system. They said Baran's network could not be built and the idea was killed.

Several years went by and still the DoD did not have a better command-and--control system. To understand what happened next, we have to go back all the way to October 1957, when the Soviet Union beat the U.S. into space with the launch of the first artificial satellite, Sputnik. When President Dwight Eisenhower tried to find out who was asleep at the switch, he was appalled to find the Army, Navy, and Air Force squabbling over the Pentagon's research budget. His immediate response was to create a single defense research organization, **ARPA**, the **Advanced Research Projects Agency**. ARPA had no scientists or laboratories; in fact, it had nothing more than an office and a small (by Pentagon standards) budget. It did its work by issuing grants and contracts to universities and companies whose ideas looked promising to it.

For the first few years, ARPA tried to figure out what its mission should be. In 1967, the attention of Larry Roberts, a program manager at ARPA who was trying to figure out how to provide remote access to computers, turned to networking. He contacted various experts to decide what to do. One of them, Wesley Clark, suggested building a packet-switched subnet, connecting each host to its own router.

After some initial skepticism, Roberts bought the idea and presented a somewhat vague paper about it at the ACM SIGOPS Symposium on Operating System Principles held in Gatlinburg, Tennessee, in late 1967 (Roberts, 1967). Much to Roberts' surprise, another paper at the conference described a similar system that had not only been designed but actually fully implemented under the direction of Donald Davies at the National Physical Laboratory in England. The NPL system was not a national system by any means. It just connected several computers on the NPL campus. Nevertheless, it convinced Roberts that packet switching could be made to work. Furthermore, it cited Baran's now discarded earlier work. Roberts came away from Gatlinburg determined to build what later became known as the **ARPANET**.

In the plan that was developed, the subnet would consist of minicomputers called **IMPs** (**Interface Message Processors**) connected by then-state-of-the-art 56-kbps transmission lines. For high reliability, each IMP would be connected to at least two other IMPs. Each packet sent across the subnet was to contain the full destination address, so if some lines and IMPs were destroyed, subsequent packets could be automatically rerouted along alternative paths.

Each node of the network was to consist of an IMP and a host, in the same room, connected by a short wire. A host could send messages of up to 8063 bits to its IMP, which would then break these up into packets of at most 1008 bits and forward them independently toward the destination. Each packet was received in its entirety before being forwarded, so the subnet was the first electronic store-and--forward packet-switching network.

ARPA then put out a tender for building the subnet. Twelve companies bid for it. After evaluating all the proposals, ARPA selected BBN, a consulting firm based in Cambridge, Massachusetts, and in December 1968 awarded it a contract to build the subnet and write the subnet software. BBN chose to use specially modified Honeywell DDP-316 minicomputers with 12K 16-bit words of magnetic core memory as the IMPs. The IMPs did not have disks since moving parts were considered unreliable. The IMPs were interconnected by 56-kbps lines leased from telephone companies. Although 56 kbps is now often the only choice of people in rural areas, back then, it was the best money could buy.

The software was split into two parts: subnet and host. The subnet software consisted of the IMP end of the host-IMP connection, the IMP-IMP protocol, and a source IMP to destination IMP protocol designed to improve reliability. The original ARPANET design is shown in Fig. 1-13.

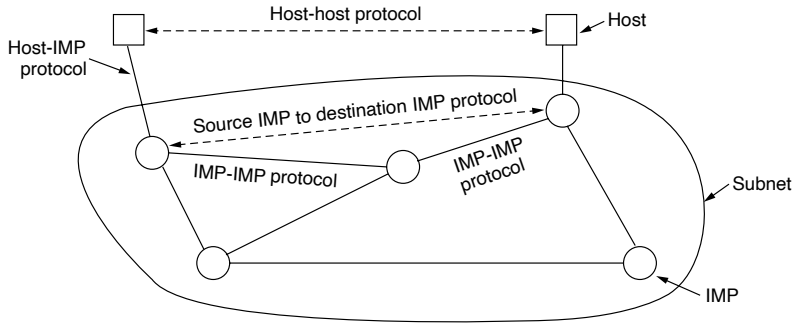

Figure 1-13. The original ARPANET design.

Outside the subnet, software was also needed, namely, the host end of the host-IMP connection, the host-host protocol, and the application software. It soon became clear that BBN was of the opinion that when it had accepted a message on a host-IMP wire and placed it on the host-IMP wire at the destination, its job was done.

Roberts had a problem, though: the hosts needed software too. To deal with it, he convened a meeting of network researchers, mostly graduate students, at Snowbird, Utah, in the summer of 1969. The graduate students expected some network

expert to explain the grand design of the network and its software to them and then assign each of them the job of writing part of it. They were astounded when there was no network expert and no grand design. They had to figure out what to do on their own.

Nevertheless, somehow an experimental network went online in December 1969 with four nodes: at UCLA, UCSB, SRI, and the University of Utah. These four were chosen because all had a large number of ARPA contracts, and all had different and completely incompatible host computers (just to make it more fun). The first host-to-host message had been sent two months earlier from the UCLA node by a team led by Len Kleinrock (a pioneer of the theory of packet switching) to the SRI node. The network grew quickly as more IMPs were delivered and installed; it soon spanned the United States. Figure 1-14 shows how rapidly the ARPANET grew in the first 3 years.

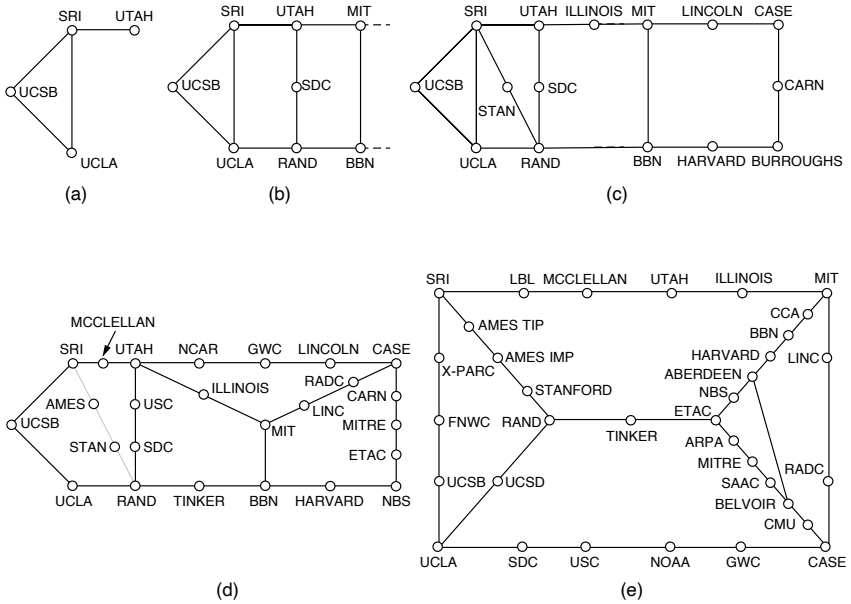

Figure 1-14. Growth of the ARPANET. (a) December 1969. (b) July 1970. (c) March 1971. (d) April 1972. (e) September 1972.

In addition to helping the fledgling ARPANET grow, ARPA also funded research on the use of satellite networks and mobile packet radio networks. In one now-famous demonstration, a big truck driving around in California used the packet radio network to send messages to SRI, which were then forwarded over the ARPANET to the East Coast, where they were then shipped to University College

in London over the satellite network. This allowed a researcher in the truck to use a computer in London while driving around in California.

This experiment also demonstrated that the existing ARPANET protocols were not suitable for running over different networks. This observation led to more research on protocols, culminating with the invention of the TCP/IP protocols (Cerf and Kahn, 1974). TCP/IP was specifically designed to handle communication over internetworks, something becoming increasingly important as more and more networks were hooked up to the ARPANET.

To encourage adoption of these new protocols, ARPA awarded several contracts to implement TCP/IP on different computer platforms, including IBM, DEC, and HP systems, as well as for Berkeley UNIX. Researchers at the University of California at Berkeley rewrote TCP/IP with a new programming interface called **sockets** for the upcoming 4.2BSD release of Berkeley UNIX. They also wrote many application, utility, and management programs to show how convenient it was to use the network with sockets.

The timing was perfect. Many universities had just acquired a second or third VAX computer and a LAN to connect them, but they had no networking software. When 4.2BSD came along, with TCP/IP, sockets, and many network utilities, the complete package was adopted immediately. Furthermore, with TCP/IP, it was easy for the LANs to connect to the ARPANET, and many did. As a result, TCP/IP use grew rapidly during the mid-1970s.

NSFNET

By the late 1970s, NSF (the U.S. National Science Foundation) saw the enormous impact the ARPANET was having on university research, allowing scientists across the country to share data and collaborate on research projects. However, to get on the ARPANET a university had to have a research contract with the DoD. Many did not have a contract. NSF's initial response was to fund **CSNET** (**Computer Science Network**) in 1981. It connected computer science departments and industrial research labs to the ARPANET via dial-up and leased lines. In the late 1980s, the NSF went further and decided to design a successor to the ARPANET that would be open to all university research groups.

To have something concrete to start with, NSF decided to build a backbone network to connect its six supercomputer centers, in San Diego, Boulder, Champaign, Pittsburgh, Ithaca, and Princeton. Each supercomputer was given a little brother, consisting of an LSI-11 microcomputer called a **fuzzball**. The fuzzballs were connected with 56-kbps leased lines and formed the subnet, the same hardware technology the ARPANET used. The software technology was different, however: the fuzzballs spoke TCP/IP right from the start, making it the first TCP/IP WAN.

NSF also funded some (eventually about 20) regional networks that connected to the backbone to allow users at thousands of universities, research labs, libraries,

and museums to access any of the supercomputers and to communicate with one another. The complete network, including backbone and the regional networks, was called **NSFNET** (**National Science Foundation Network**). It connected to the ARPANET through a link between an IMP and a fuzzball in the Carnegie-Mellon machine room. The first NSFNET backbone is illustrated in Fig. 1-15 superimposed on a map of the United States.

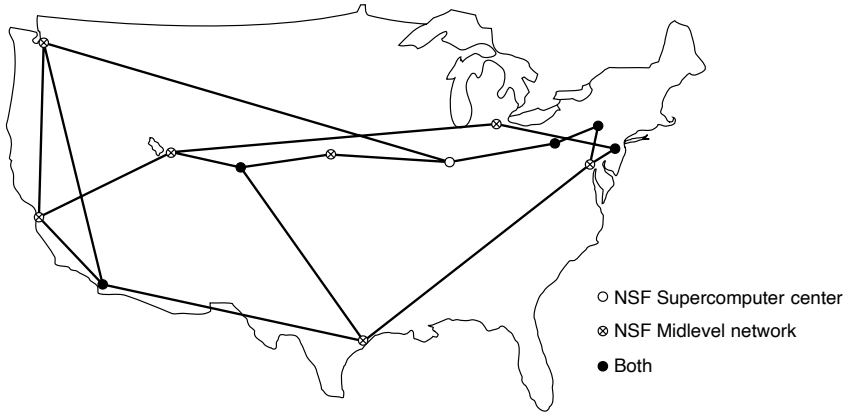

Figure 1-15. The NSFNET backbone in 1988.

NSFNET was an instantaneous success and was overloaded from the word go. NSF immediately began planning its successor and awarded a contract to the Michigan-based MERIT consortium to run it. Fiber optic channels at 448 kbps were leased from MCI (which was purchased by Verizon in 2006) to provide the version 2 backbone. IBM PC-RTs were used as routers. This, too, was soon overwhelmed, and by 1990, the second backbone was upgraded to 1.5 Mbps.

As growth continued, NSF realized that the government could not continue financing networking forever. Furthermore, commercial organizations wanted to join but were forbidden by NSF's charter from using networks NSF paid for. Consequently, NSF encouraged MERIT, MCI, and IBM to form a nonprofit corporation, **ANS** (**Advanced Networks and Services**), as the first step along the road to commercialization. In 1990, ANS took over NSFNET and upgraded the 1.5-Mbps links to 45 Mbps to form **ANSNET**. This network operated for 5 years and was then sold to America Online. But by then, various companies were offering commercial IP service and it was clear that the government should now get out of the networking business.

To ease the transition and make sure every regional network could communicate with every other regional network, NSF awarded contracts to four different network operators to establish a **NAP** (**Network Access Point**). These operators

were PacBell (San Francisco), Ameritech (Chicago), MFS (Washington, D.C.), and Sprint (New York City, where for NAP purposes, Pennsauken, New Jersey counts as New York City). Every network operator that wanted to provide backbone service to the NSF regional networks had to connect to all the NAPs.

This arrangement meant that a packet originating on any regional network had a choice of backbone carriers to get from its NAP to the destination's NAP. Consequently, the backbone carriers were forced to compete for the regional networks' business on the basis of service and price, which was the idea, of course. As a result, the concept of a single default backbone was replaced by a commercially driven competitive infrastructure. Many people like to criticize the federal government for not being innovative, but in the area of networking, it was DoD and NSF that created the infrastructure that formed the basis for the Internet and then handed it over to industry to operate. This happened because when DoD asked AT&T to build the ARPANET, it saw no value in computer networks and refused to do it.

During the 1990s, many other countries and regions also built national research networks, often patterned on the ARPANET and NSFNET. These included EuropaNET and EBONE in Europe, which started out with 2-Mbps lines and then upgraded to 34-Mbps lines. Eventually, the network infrastructure in Europe was handed over to industry as well.

The Internet has changed a great deal since those early days. It exploded in size with the emergence of the World Wide Web (WWW) in the early 1990s. Recent data from the Internet Systems Consortium puts the number of visible Internet hosts at over 600 million. This guess is only a low-ball estimate, but it far exceeds the few million hosts that were around when the first conference on the WWW was held at CERN in 1994.

The way we use the Internet has also changed radically. Initially, applications such as email-for-academics, newsgroups, remote login, and file transfer dominated. Later, it switched to email-for-everyman, then the Web, and peer-to-peer content distribution, such as the now-shuttered Napster. Now real-time media distribution and social media (e.g., Twitter, Facebook) are mainstays. The dominant form of traffic on the Internet now is, by far, streaming video (e.g., Netflix and YouTube). These developments brought richer kinds of media to the Internet and hence much more traffic, which have also had implications for the Internet architecture itself.

The Internet Architecture

The architecture of the Internet has also changed a great deal as it has grown explosively. In this section, we will attempt to give a brief overview of what it looks like today. The picture is complicated by continuous upheavals in the businesses of telephone companies (telcos), cable companies, and ISPs that often make it hard to tell who is doing what. One driver of these upheavals is convergence in

the telecommunications industry, in which one network is used for previously different uses. For example, in a ''triple play,'' one company sells you telephony, TV, and Internet service over the same network connection for a lower price than the three services would cost individually. Consequently, the description given here will be a simplified version of reality. And what is true today may not be true tomorrow.

Fig. 1-16 shows a high-level overview of the Internet architecture. Let us examine this figure piece by piece, starting with a computer at home (at the edges of the figure). To join the Internet, the computer is connected to an internet service provider from whom the user purchases Internet access. This lets the computer exchange packets with all of the other accessible hosts on the Internet. There are many kinds of Internet access, and they are usually distinguished by how much bandwidth they provide and how much they cost, but the most important attribute is connectivity.

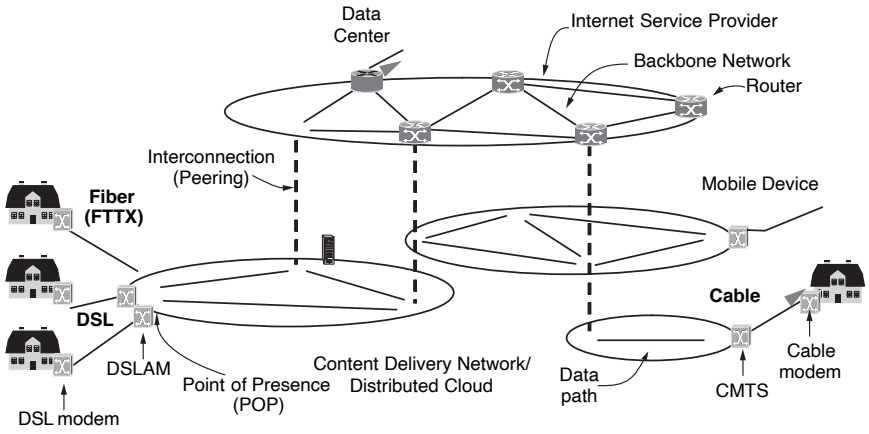

Figure 1-16. Overview of the Internet architecture.

A common method for connecting to the Internet from your home is to send signals over the cable television infrastructure. The cable network, sometimes called an **HFC** (**Hybrid Fiber-Coaxial**) network, is a single integrated infrastructure that uses a packet-based transport called **DOCSIS** (**Data Over Cable Service Interface Specification**) to transmit a variety of data services, including television channels, high-speed data, and voice. The device at the home end is called a **cable modem**, and the device at the **cable headend** is called the **CMTS** (**Cable Modem Termination System**). The word **modem** is short for ''*mo*dulator *dem*odulator'' and refers to any device that converts between digital bits and analog signals.

Access networks are limited by the bandwidth of the ''last mile'' or last leg of transmission. Over the last decade, the DOCSIS standard has advanced to enable

significantly higher throughput to home networks. The most recent standard, DOCSIS 3.1 full duplex, introduces support for symmetric upstream and downstream data rates, with a maximum capacity of 10 Gbps. Another option for last-mile deployment involves running optical fiber to residences using a technology called **FTTH** (**Fiber to the Home**). For businesses in commercial areas, it may make sense to lease a dedicated high-speed transmission line from the offices to the nearest ISP. In large cities in some parts of the world, leased lines of up to 10 Gbps are available; lower speeds are also available. For example, a T3 line runs at roughly 45 Mbps. In other parts of the world, especially in developing regions, there is neither cable nor fiber deployed; some of these regions are jumping straight to higher-speed wireless or mobile networks as the predominant means of Internet access. We will provide an overview of mobile Internet access in the next section.

We can now move packets between the home and the ISP. We call the location at which customer packets enter the ISP network for service the ISP's **POP** (**Point of Presence**). We will next explain how packets are moved between the POPs of different ISPs. From this point on, the system is fully digital and packet switched.

ISP networks may be regional, national, or international. We have already seen that their architecture includes long-distance transmission lines that interconnect routers at POPs in the different cities that the ISPs serve. This equipment is called the **backbone** of the ISP. If a packet is destined for a host served directly by the ISP, that packet is routed over the backbone and delivered to the host. Otherwise, it must be handed over to another ISP.

ISPs connect their networks to exchange traffic at **IXPs** (**Internet eXchange Points**). The connected ISPs are said to **peer** with each other. There are many IXPs in cities around the world. They are drawn vertically in Fig. 1-16 because ISP networks overlap geographically. Basically, an IXP is a building full of routers, at least one per ISP. A very fast optical LAN in the room connects all the routers, so packets can be forwarded from any ISP backbone to any other ISP backbone. IXPs can be large and independently owned facilities that compete with each other for business. One of the largest is the Amsterdam Internet Exchange (AMS-IX), to which over 800 ISPs connect and through which they exchange over 4000 gigabits (4 terabits) worth of traffic *every second*.

Peering at IXPs depends on the business relationships between ISPs. There are many possible relationships. For example, a small ISP might pay a larger ISP for Internet connectivity to reach distant hosts, much as a customer purchases service from an Internet provider. In this case, the small ISP is said to pay for **transit**. Alternatively, two large ISPs might decide to exchange traffic so that each ISP can deliver some traffic to the other ISP without having to pay for transit. One of the many paradoxes of the Internet is that ISPs who publicly compete with one another for customers often privately cooperate to do peering (Metz, 2001).

The path a packet takes through the Internet depends on the peering choices of the ISPs. If the ISP that is delivering a packet peers with the destination ISP, it might deliver the packet directly to its peer. Otherwise, it might route the packet to

the nearest place at which it connects to a paid transit provider so that provider can deliver the packet. Two example paths across ISPs are shown in Fig. 1-16. Often, the path a packet takes will not be the shortest path through the Internet. It could be the least congested or the cheapest for the ISPs.

A small handful of **transit providers**, including AT&T and Level 3, operate large international backbone networks with thousands of routers connected by high-bandwidth fiber-optic links. These ISPs do not pay for transit. They are usually called **tier-1** ISPs and are said to form the backbone of the Internet, since everyone else must connect to them to be able to reach the entire Internet.

Companies that provide lots of content, such as Facebook and Netflix, locate their servers in **data centers** that are well-connected to the rest of the Internet. These data centers are designed for computers, not humans, and may be filled with rack upon rack of machines. Such an installation is called a **server farm**. **Colocation** or **hosting** data centers let customers put equipment such as servers at ISP POPs so that short, fast connections can be made between the servers and the ISP backbones. The Internet hosting industry has become increasingly virtualized so that it is now common to rent a virtual machine that is run on a server farm instead of installing a physical computer. These data centers are so large (hundreds of thousands or millions of machines) that electricity is a major cost, so data centers are sometimes built in areas where electricity is cheap. For example, Google built a $2 billion data center in The Dalles, Oregon, because it is close to a huge hydroelectric dam on the mighty Columbia River that supplies it with cheap green electric power.

Conventionally, the Internet architecture has been viewed as a hierarchy, with the tier-1 providers at the top of the hierarchy and other networks further down the hierarchy, depending on whether they are large regional networks or smaller access networks, as shown in Fig. 1-17. Over the past decade, however, this hierarchy has evolved and ''flattened'' dramatically, as shown in Fig. 1-18. The impetus for this shakeup has been the rise of ''hyper-giant'' content providers, including Google, Netflix, Twitch, and Amazon, as well as large, globally distributed CDNs such as Akamai, Limelight, and Cloudflare. They have changed the Internet architecture once again. Whereas in the past, these content providers would have had to rely on transit networks to deliver content to local access ISPs, both the access ISPs and the content providers have proliferated and become so large that they often connect directly to one another in many distinct locations. In many cases, the common Internet path will be directly from your access ISP to the content provider. In some cases, the content provider will even host servers inside the access ISP's network.

1.4.2 Mobile Networks

Mobile networks have more than five billion subscribers worldwide. To put this number in perspective, it is roughly 65% of the world's population. Many, if not most, of these subscribers have Internet access using their mobile device (ITU,

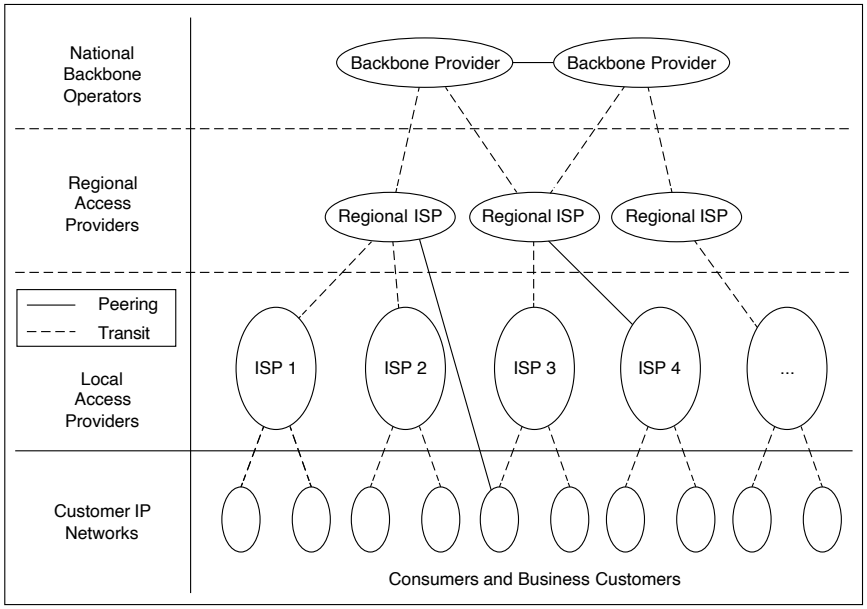

Figure 1-17. The Internet architecture through the 1990s followed a hierarchical structure.

2016). In 2018, mobile Internet traffic became more than half of global online traffic. Consequently, studying the mobile phone system is up next.

Mobile Network Architecture

The architecture of the mobile phone network is very different than that of the Internet. It has several parts, as shown in the simplified version of the 4G LTE architecture in Fig. 1-19. This is one of the more common mobile network standards and will continue to be until it is replaced by 5G, the fifth generation network. We will discuss the history of the various generations shortly.

First, there is the **E-UTRAN** (**Evolved UMTS Terrestrial Radio Access Network**) which is a fancy name for the radio communication protocol that is used over the air between the mobile device (e.g., the cell phone) and the **cellular base station**, which is now called an **eNodeB**. **UMTS** (**Universal Mobile Telecommunications System**) is the formal name for the cellular phone network. Advances in the air interface over the past decades have greatly increased wireless data rates (and are still increasing them). The air interface is based on **CDMA** (**Code Division Multiple Access**), a technique that we will study in Chap. 2.

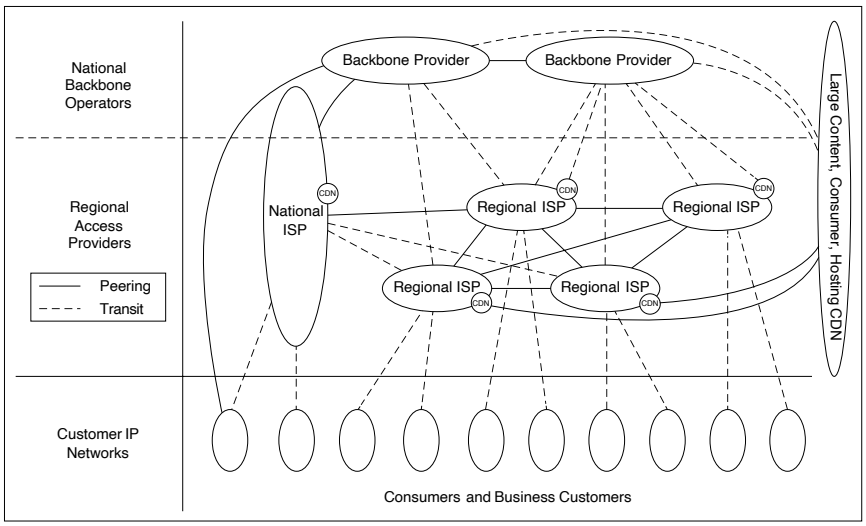

Figure 1-18. Flattening of the Internet hierarchy.

The cellular base station together with its controller forms the **radio access network**. This part is the wireless side of the mobile phone network. The controller node or **RNC** (**Radio Network Controller**) controls how the spectrum is used. The base station implements the air interface.

The rest of the mobile phone network carries the traffic for the radio access network. It is called the **core network**. In 4G networks, the core network became packet-switched, and is now called the **EPC** (**Evolved Packet Core**). The 3G UMTS core network evolved from the core network used for the 2G GSM system that came before it; the 4G EPC completed the transition to a fully packet-switched core network. The 5G system is also fully digital, too. There is no going back now. Analog is as dead as the dodo.

Data services have become a much more important part of the mobile phone network than they used to be, starting with text messaging and early packet data services such as **GPRS** (**General Packet Radio Service**) in the GSM system. These older data services ran at tens of kbps, but users wanted even higher speeds.. Newer mobile phone networks support rates of multiple Mbps. For comparison, a voice call is carried at a nominal rate of 64 kbps, typically 3–4x less with compression.

To carry all of this data, the UMTS core network nodes connect directly to a packet-switched network. The **S-GW** (**Serving Network Gateway**) and the **P-GW** (**Packet Data Network Gateway**) deliver data packets to and from mobiles and interface to external packet networks such as the Internet.

Figure 1-19. Simplified 4G LTE network architecture.

This transition is set to continue in future mobile phone networks. Internet protocols are even used on mobiles to set up connections for voice calls over a packet data network, in the manner of voice over IP. IP and packets are used all the way from the radio access through to the core network. Of course, the way that IP networks are designed is also changing to support better quality of service. If it did not, then problems with chopped-up audio and jerky video would not impress paying customers. We will return to this subject in Chap. 5.

Another difference between mobile phone networks and the conventional Internet is mobility. When a user moves out of the range of one cellular base station and into the range of another one, the flow of data must be re-routed from the old to the new cell base station. This technique is known as **handover** or **handoff**, and it is illustrated in Fig. 1-20.

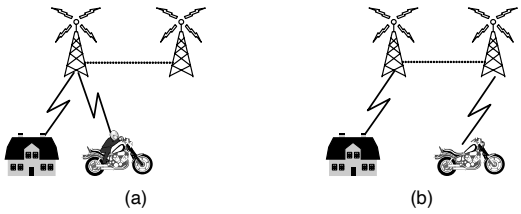

Figure 1-20. Mobile phone handover (a) before. (b) after.

Either the mobile device or the base station may request a handover when the quality of the signal drops. In some cell networks, usually those based on CDMA

technology, it is possible to connect to the new base station before disconnecting from the old base station. This improves the connection quality for the mobile because there is no break in service; the mobile is actually connected to two base stations for a short while. This way of doing a handover is called a **soft handover** to distinguish it from a **hard handover**, in which the mobile disconnects from the old base station before connecting to the new one.

A related issue is how to find a mobile in the first place when there is an incoming call. Each mobile phone network has a **HSS** (**Home Subscriber Server**) in the core network that knows the location of each subscriber, as well as other profile information that is used for authentication and authorization. In this way, each mobile can be found by contacting the HSS.

A final area to discuss is security. Historically, phone companies have taken security much more seriously than Internet companies because they needed to bill for service and avoid (payment) fraud. Unfortunately, that is not saying much. Nevertheless, in the evolution from 1G through 5G technologies, mobile phone companies have been able to roll out some basic security mechanisms for mobiles.

Starting with the 2G GSM system, the mobile phone was divided into a handset and a removable chip containing the subscriber's identity and account information. The chip is informally called a **SIM card**, short for **Subscriber Identity Module**. SIM cards can be switched to different handsets to activate them, and they provide a basis for security. When GSM customers travel to other countries on vacation or business, they often bring their handsets but buy a new SIM card for few dollars upon arrival in order to make local calls with no roaming charges.

To reduce fraud, information on SIM cards is also used by the mobile phone network to authenticate subscribers and check that they are allowed to use the network. With UMTS, the mobile also uses the information on the SIM card to check that it is talking to a legitimate network.

Privacy is another important consideration. Wireless signals are broadcast to all nearby receivers, so to make it difficult to eavesdrop on conversations, cryptographic keys on the SIM card are used to encrypt transmissions. This approach provides much better privacy than in 1G systems, which were easily tapped, but is not a panacea due to weaknesses in the encryption schemes.

Packet Switching and Circuit Switching

Since the beginning of networking, a war has been going on between the people who support packet-switched networks (which are connectionless) and the people who support circuit-switched networks (which are connection-oriented). The main proponents of **packet switching** come from the Internet community. In a connectionless design, every packet is routed independently of every other packet. As a consequence, if some routers go down during a session, no harm will be done as long as the system can dynamically reconfigure itself so that subsequent packets can find some other route to the destination, even if it is different from that which

previous packets used. In a packet-switched network, if too many packets arrive at the a router during a particular time interval, the router will choke and probably lose packets. The sender will eventually notice this and resend the data, but the quality of service may be poor unless the applications account for this variability.

The **circuit switching** camp comes from the world of telephone companies. In the telephone system, a caller must dial the called party's number and wait for a connection before talking or sending data. This connection setup establishes a route through the telephone system that is maintained until the call is terminated. All words or packets follow the same route. If a line or switch on the path goes down, the call is aborted, making it less fault tolerant than a connectionless design.

Circuit switching can support quality of service more easily. By setting up a connection in advance, the subnet can reserve link bandwidth, switch buffer space, and CPU time. If an attempt is made to set up a call and insufficient resources are available, the call is rejected and the caller gets a kind of busy signal. In this way, once a connection has been set up, the connection will get good service.

The surprise in Fig. 1-19 is that there is both packet- and circuit-switched equipment in the core network. This shows that the mobile phone network is in transition, with mobile phone companies able to implement one or sometimes both of the alternatives. Older mobile phone networks used a circuit-switched core in the style of the traditional phone network to carry voice calls. This legacy is seen in the UMTS network with the **MSC** (**Mobile Switching Center**), **GMSC** (**Gateway Mobile Switching Center**), and **MGW** (**Media Gateway**) elements that set up connections over a circuit-switched core network such as the **PSTN** (**Public Switched Telephone Network**).

Early Generation Mobile Networks: 1G, 2G, and 3G

The architecture of the mobile network has changed greatly over the past 50 years along with its tremendous growth. First-generation mobile phone systems transmitted voice calls as continuously varying (analog) signals rather than sequences of (digital) bits. **AMPS** (**Advanced Mobile Phone System**), which was deployed in the United States in 1982, was a widely used first-generation system. Second-generation mobile phone systems switched to transmitting voice calls in digital form to increase capacity, improve security, and offer text messaging. **GSM** (**Global System for Mobile communications**), which was deployed starting in 1991 and has become widely used worldwide. It is a 2G system.

The third generation, or 3G, systems were initially deployed in 2001 and offer both digital voice and broadband digital data services. They also come with a lot of jargon and many different standards to choose from. 3G is loosely defined by the ITU (an international standards body we will discuss later on in this chapter)) as providing rates of at least 2 Mbps for stationary or walking users and 384 kbps in a moving vehicle. UMTS is the main 3G system that is deployed worldwide. It is also the basis for its various successors. It can provide up to 14 Mbps on the

downlink and almost 6 Mbps on the uplink. Future releases will use multiple antennas and radios to provide even greater speeds for users.

The scarce resource in 3G systems, as in 2G and 1G systems before them, is radio spectrum. Governments license the right to use parts of the spectrum to the mobile phone network operators, often using a spectrum auction in which network operators submit bids. Having a piece of licensed spectrum makes it easier to design and operate systems, since no one else is allowed to transmit on that spectrum, but it often costs a serious amount of money. In the United Kingdom in 2000, for example, five 3G licenses were auctioned for a total of about $40 billion.

It is the scarcity of spectrum that led to the **cellular network** design shown in Fig. 1-21 that is now used for mobile phone networks. To manage the radio interference between users, the coverage area is divided into cells. Within a cell, users are assigned channels that do not interfere with each other and do not cause too much interference for adjacent cells. This allows for good reuse of the spectrum, or **frequency reuse**, in the neighboring cells, which increases the capacity of the network. In 1G systems, which carried each voice call on a specific frequency band, the frequencies were carefully chosen so that they did not conflict with neighboring cells. In this way, a given frequency might only be reused once in several cells. Modern 3G systems allow each cell to use all frequencies, but in a way that results in a tolerable level of interference to the neighboring cells. There are variations on the cellular design, including the use of directional or sectored antennas on cell towers to further reduce interference, but the basic idea is the same.

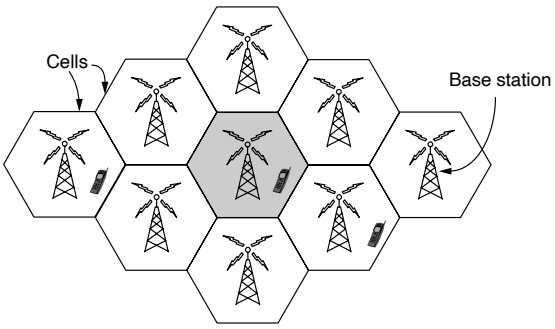

Figure 1-21. Cellular design of mobile phone networks.

Modern Mobile Networks: 4G and 5G

Mobile phone networks are destined to play a big role in future networks. They are now more about mobile broadband applications (e.g., accessing the Web from a phone) than voice calls, and this has major implications for the air interfaces, core

network architecture, and security of future networks. The 4G, later 4G (LTE (Long Term Evolution) technologies offer faster speeds, emerged in the late 2000s.

4G LTE networks very quickly became the predominant mode of mobile Internet access in the late 2000s, outpacing competitors like 802.16, sometimes called **WiMAX**. 5G technologies are promising faster speeds—up to 10 Gbps—and are now set for large-scale deployment in the early 2020s. One of the main distinctions between these technologies is the frequency spectrum that they rely on. For example, 4G uses frequency bands up to 20 MHz; in contrast, 5G is designed to operate in much higher frequency bands, of up to 6 GHz. The challenge when moving to higher frequencies is that the higher frequency signals do not travel as far as lower frequencies, so the technology must account for signal attenuation, interference, and errors using newer algorithms and technologies, including multiple input multiple output (MIMO) antenna arrays. The short microwaves at these frequencies are also absorbed easily by water, requiring special efforts to have them work when it is raining.

1.4.3 Wireless Networks (WiFi)

Almost as soon as laptops appeared, many people dreamed of walking into an office and magically having their laptop computer be connected to the Internet. Various groups worked for years to accomplish this goal. The most practical approach is to equip both the office and the laptop computers with short-range radio transmitters and receivers to allow them to talk.

Work in this field rapidly led to wireless LANs being marketed by a variety of companies. The trouble was that no two of them were compatible. The proliferation of standards meant that a computer equipped with a brand *X* radio would not work in a room equipped with a brand *Y* base station. In the mid 1990s, the industry decided that a wireless LAN standard might be a good idea, so the IEEE committee that had standardized wired LANs was given the task of drawing up a wireless LAN standard.

The first decision was the easiest: what to call it. All the other LAN standards produced by IEEE's 802 standards committee had numbers like 802.1, 802.2, and 802.3, up to 802.10, so the wireless LAN standard was dubbed 802.11. Truly brilliant. A common slang name for it is **WiFi**, but it is an important standard and deserves respect, so we will call it by its more formal name, 802.11. Many variants and versions of the 802.11 standard have emerged and evolved over the years.

After settling on the name, the rest was harder. The first problem was to find a suitable frequency band that was available, preferably worldwide. The approach taken was the opposite of that used in mobile phone networks. Instead of expensive, licensed spectrum, 802.11 systems operate in unlicensed bands such as the **ISM** (**Industrial, Scientific, and Medical**) bands defined by ITU-R (e.g., 902-928 MHz, 2.4-2.5 GHz, 5.725-5.825 GHz). All devices are allowed to use this

spectrum provided that they limit their transmit power to let different devices coexist. Of course, this means that 802.11 radios may find themselves competing with cordless phones, garage door openers, and microwave ovens. So unless designers think people want to call to their garage doors, it is important to get this right.

802.11 networks have clients, such as laptops and mobile phones, as well as infrastructure called **APs** (**access points**) that is installed in buildings. Access points are sometimes called **base stations**. The access points connect to the wired network, and all communication between clients goes through an access point. It is also possible for clients that are in radio range to talk directly, such as two computers in an office without an access point. This arrangement is called an **ad hoc network**. It is used much less often than the access point mode. Both modes are shown in Fig. 1-22.

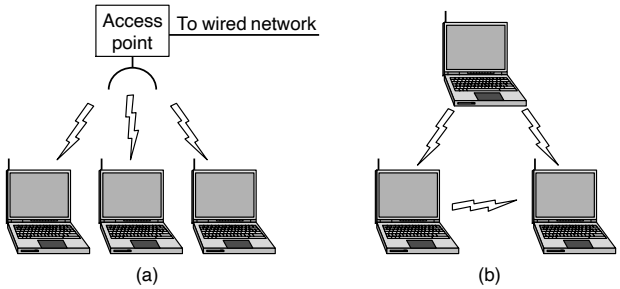

Figure 1-22. (a) Wireless network with an access point. (b) Ad hoc network.

802.11 transmission is complicated by wireless conditions that vary with even small changes in the environment. At the frequencies used for 802.11, radio signals can be reflected off solid objects so that multiple echoes of a transmission may reach a receiver along different paths. The echoes can cancel or reinforce each other, causing the received signal to fluctuate greatly. This phenomenon is called **multipath fading**, and it is shown in Fig. 1-23.

The key idea for overcoming variable wireless conditions is **path diversity**, or the sending of information along multiple, independent paths. In this way, the information is likely to be received even if one of the paths happens to be poor due to a fade. These independent paths are typically built into the digital modulation scheme used in the hardware. Options include using different frequencies across the allowed band, following different spatial paths between different pairs of antennas, or repeating bits over different periods of time.

Different versions of 802.11 have used all of these techniques. The initial (1997) standard defined a wireless LAN that ran at either 1 Mbps or 2 Mbps by hopping between frequencies or spreading the signal across the allowed spectrum. Almost immediately, people complained that it was too slow, so work began on faster standards. The spread spectrum design was later extended and became the

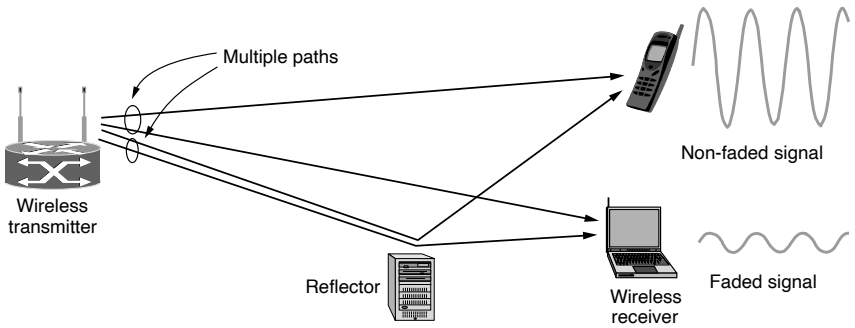

Figure 1-23. Multipath fading.

802.11b standard (1999) running at rates up to 11 Mbps. The 802.11a (1999) and 802.11g (2003) standards then switched to a different modulation scheme called **OFDM** (**Orthogonal Frequency Division Multiplexing**). It divides a wide band of spectrum into many narrow slices over which different bits are sent in parallel. This improved scheme, which we will study in Chap. 2, boosted the 802.11a/g bit rates up to 54 Mbps. That is a significant increase, but people still wanted more throughput to support more demanding uses. More recent versions of the standard offer higher data rates. The commonly deployed 802.11ac can run at 3.5 Gbps. The newer 802.11ad can run at 7 Gbps, but only indoors within a single room since the radio waves at the frequencies it uses do not penetrate walls very well.

Since wireless is inherently a broadcast medium, 802.11 radios also have to deal with the problem that multiple transmissions that are sent at the same time will collide, which may interfere with reception. To handle this problem, 802.11 uses a **CSMA** (**Carrier Sense Multiple Access**) scheme that draws on ideas from classic wired Ethernet, which, ironically, drew from an early wireless network developed in Hawaii called **ALOHA**. Computers wait for a short random interval before transmitting and defer their transmissions if they hear that someone else is already transmitting. This scheme makes it less likely that two computers will send at the same time. It does not work as well as in the case of wired networks, though. To see why, examine Fig. 1-24. Suppose that computer *A* is transmitting to computer *B*, but the radio range of *A*'s transmitter is too short to reach computer *C*. If *C* wants to transmit to *B*, it can listen before starting, but the fact that it does not hear anything does not mean that its transmission will succeed. The inability of *C* to hear *A* before starting causes some collisions to occur. After any collision, the sender then waits another, longer, random delay and retransmits the packet. Despite this and some other issues, the scheme works well enough in practice.

Mobility presents another challenge. If a mobile client is moved away from the access point it is using and into the range of a different access point, some way

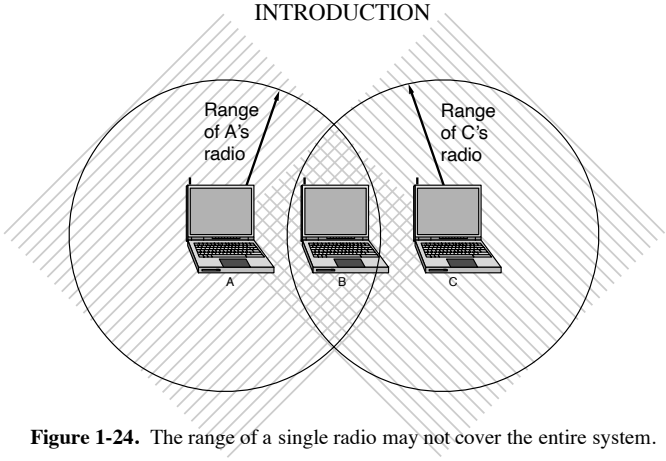

Figure 1-24. The range of a single radio may not cover the entire system.

of handing it off is needed. The solution is that an 802.11 network can consist of multiple cells, each with its own access point, and a distribution system that connects the cells. The distribution system is often switched Ethernet, but it can use any technology. As the clients move, they may find another access point with a better signal than the one they are currently using and change their association. From the outside, the entire system looks like a single wired LAN.

That said, mobility in 802.11 has been of limited value so far compared to mobility in the mobile phone network. Typically, 802.11 is used by nomadic clients that go from one fixed location to another, rather than being used on-the-go. Mobility is not really needed for nomadic usage. Even when 802.11 mobility is used, it extends over a single 802.11 network, which might cover at most a large building. Future schemes will need to provide mobility across different networks and across different technologies (e.g., 802.21, which deals with the handover between wired and wireless networks).

Finally, there is the problem of security. Since wireless transmissions are broadcast, it is easy for nearby computers to receive packets of information that were not intended for them. To prevent this, the 802.11 standard included an encryption scheme known as **WEP** (**Wired Equivalent Privacy**). The idea was to make wireless security like that of wired security. It is a good idea, but unfortunately, the scheme was flawed and soon broken (Borisov et al., 2001). It has since been replaced with newer schemes that have different cryptographic details in the 802.11i standard, called **WiFi Protected Access**, initially called **WPA** (**WiFi Protected Access**) but now replaced by **WPA2**, and even more sophisticated protocols such as **802.1X**, which allows certificated-based authentication of the access point to the client, as well as a variety of different ways for the client to authenticate itself to the access point.

802.11 has caused a revolution in wireless networking that is set to continue. Beyond buildings, it is now prevalent in trains, planes, boats, and automobiles so that people can surf the Internet wherever they go. Mobile phones and all manner

of consumer electronics, from game consoles to digital cameras, can communicate with it. There is even a convergence of 802.11 with other types of mobile technologies; a prominent example of this convergence is **LTE-Unlicensed** (**LTE-U**) which is an adaptation of 4G LTE cellular network technology that would allow it to operate in the unlicensed spectrum, as an alternative to ISP-owned WiFi ''hotspots.'' We will return to all of these mobile and cellular network technologies in Chap. 4.

1.5 NETWORK PROTOCOLS

We begin this section with a discussion of the design goals of various network protocols. We then explore a central concept in network protocol design: layering. Then, we talk about connection-oriented vs. connectionless services, as well as the specific service primitives that support these services.

1.5.1 Design Goals

Network protocols often share a common set of design goals, which include reliability (the ability to recover from errors, faults, or failures); resource allocation (sharing access to a common, limited resource); evolvability (allowing for incremental deployment of protocol improvements over time); and security (defending the network against various types of attacks). In this section, we explore each of these goals at a high level.

Reliability

Some of the key design issues that occur in computer networks will come up in layer after layer. Below, we will briefly mention the more important ones.

Reliability is the design issue of making a network that operates correctly even though it is comprised of a collection of components that are themselves unreliable. Think about the bits of a packet traveling through the network. There is a chance that some of these bits will be received damaged (inverted) due to fluke electrical noise, random wireless signals, hardware flaws, software bugs, and so on. How is it possible that we find and fix these errors?

One mechanism for finding errors in received information uses codes for **error detection**. Information that is incorrectly received can then be retransmitted until it is received correctly. More powerful codes allow for **error correction**, where the correct message is recovered from the possibly incorrect bits that were originally received. Both of these mechanisms work by adding redundant information. They are used at low layers, to protect packets sent over individual links, and high layers, to check that the right contents were received.

Another reliability issue is finding a working path through a network. Often, there are multiple paths between a source and destination, and in a large network,

there may be some links or routers that are broken. Suppose for example, that the network is down in Berlin. Packets sent from London to Rome via Berlin will not get through, but we could instead send packets from London to Rome via Paris. The network should automatically make this decision. This topic is called **routing**.

Resource Allocation

A second design issue is resource allocation. When networks get large, new problems arise. Cities can have traffic jams, a shortage of telephone numbers, and it is easy to get lost. Not many people have these problems in their own neighborhood, but citywide they may be a big issue. Designs that continue to work well when the network gets large are said to be **scalable**. Networks provide a service to hosts using their underlying resources, such as the capacity of transmission lines. To do this well, they need mechanisms that divide their resources so that one host does not interfere with another too much.

Many designs share network bandwidth dynamically, according to the short-term needs of hosts, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use. This design is called **statistical multiplexing**, meaning sharing based on the statistics of demand. It can be applied at low layers for a single link, or at high layers for a network or even applications that use the network.

An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This subject is called **flow control**. Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, and the network cannot deliver it all. This overloading of the network is called **congestion**. One strategy is for each computer to reduce its demand for resources (e.g., bandwidth) when it experiences congestion. It, too, can be used in all layers.

It is interesting to observe that the network has more resources to offer than simply bandwidth. For uses such as carrying live video, the timeliness of delivery matters a great deal. Most networks must provide service to applications that want this **real-time** delivery at the same time that they provide service to applications that want high throughput. **Quality of service** is the name given to mechanisms that reconcile these competing demands.

Evolvability

Another design issue concerns the evolution of the network. Over time, networks grow larger and new designs emerge that need to be connected to the existing network. We have recently seen the key structuring mechanism used to support change by dividing the overall problem and hiding implementation details: **protocol layering**. There are many other strategies available to designers as well.

Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message. This mechanism is called **addressing** or **naming**, in the low and high layers, respectively.

An aspect of growth is that different network technologies often have different limitations. For example, not all communication channels preserve the order of messages sent on them, leading to solutions that number messages. Another example is differences in the maximum size of a message that the networks can transmit. This leads to mechanisms for disassembling, transmitting, and then reassembling messages. This overall topic is called **internetworking**.

Security

The last major design issue is to secure the network by defending it against different kinds of threats. One of the threats we have mentioned previously is that of eavesdropping on communications. Mechanisms that provide **confidentiality** defend against this threat, and they are used in multiple layers. Mechanisms for **authentication** prevent someone from impersonating someone else. They might be used to tell fake banking Web sites from the real one, or to let the cellular network check that a call is really coming from your phone so that you will pay the bill. Other mechanisms for **integrity** prevent surreptitious changes to messages, such as altering ''debit my account $10'' to ''debit my account $1000.'' All of these designs are based on cryptography, which we shall study in Chap. 8.

1.5.2 Protocol Layering

To reduce their design complexity, most networks are organized as a stack of **layers** or **levels**, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is actually a familiar one and is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

When layer n on one machine carries on a conversation with layer n on another machine, the rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a **protocol** is an agreement between the communicating parties on how communication is to proceed. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn,

may decide to either shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

A five-layer network is illustrated in Fig. 1-25. The entities comprising the corresponding layers on different machines are called **peers**. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

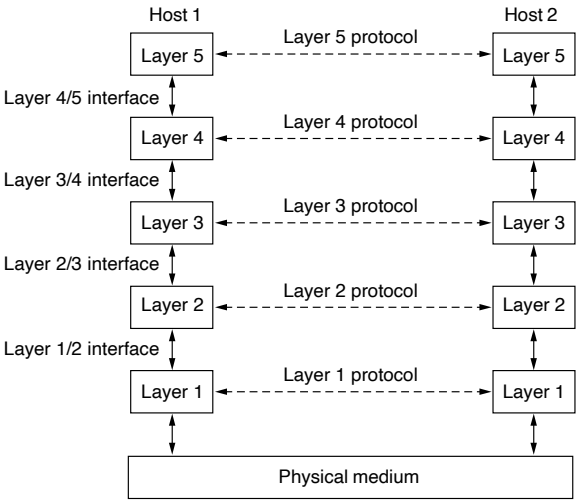

Figure 1-25. Layers, protocols, and interfaces.

In reality, no data are directly transferred from layer *n* on one machine to layer *n* on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the **physical medium** through which actual communication occurs. In Fig. 1-25, virtual communication is shown by dashed lines and physical communication by solid lines.

Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer performs a specific collection of well-understood functions. In addition to minimizing the amount of information that must be passed between layers, clear

interfaces also make it simpler to replace one layer with a completely different protocol or implementation. For example, imagine replacing all the telephone lines by satellite channels because all that is required of the new protocol or implementation is that it offers exactly the same set of services to its upstairs neighbor as the old one did. It is common that different hosts use different implementations of the same protocol (often written by different companies) In fact, the protocol itself can change in some layer without the layers above and below it even noticing.

A set of layers and protocols is called a **network architecture**. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. However, neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of the protocols used by a certain system, one protocol per layer, is called a **protocol stack**. Network architectures, protocol stacks, and the protocols themselves are the principal subjects of this book.

An analogy may help explain the idea of multilayer communication. Imagine two philosophers (peer processes in layer 3), one of whom speaks Urdu and English and one of whom speaks Chinese and French. Since they have no common language, they each engage a translator (peer processes at layer 2), each of whom in turn contacts a secretary (peer processes in layer 1). Philosopher 1 wishes to convey his affection for *oryctolagus cuniculus* to his peer. To do so, he passes a message (in English) across the 2/3 interface to his translator, saying ''I like rabbits,'' as illustrated in Fig. 1-26. The translators have agreed on a neutral language known to both of them, Dutch, so the message is converted to ''Ik vind konijnen leuk.'' The choice of the language is the layer 2 protocol and is up to the layer 2 peer processes.

The translator then gives the message to a secretary for transmission, for example, by fax (the layer 1 protocol). When the message arrives at the other secretary, it is passed to the local translator, who translates it into French and passes it across the 2/3 interface to the second philosopher. Note that each protocol is completely independent of the other ones as long as the interfaces are not changed. The translators can switch from Dutch to, say, Finnish, at will, provided that they both agree and neither changes his interface with either layer 1 or layer 3. Similarly, the secretaries can switch from email to telephone without disturbing (or even informing) the other layers. Each process may add some information intended only for its peer. This information is not passed up to the layer above.

Now consider a more technical example: how to provide communication to the top layer of the five-layer network in Fig. 1-27. A message, *M*, is produced by an application process running in layer 5 and given to layer 4 for transmission. Layer 4 puts a **header** in front of the message to identify the message and then passes the

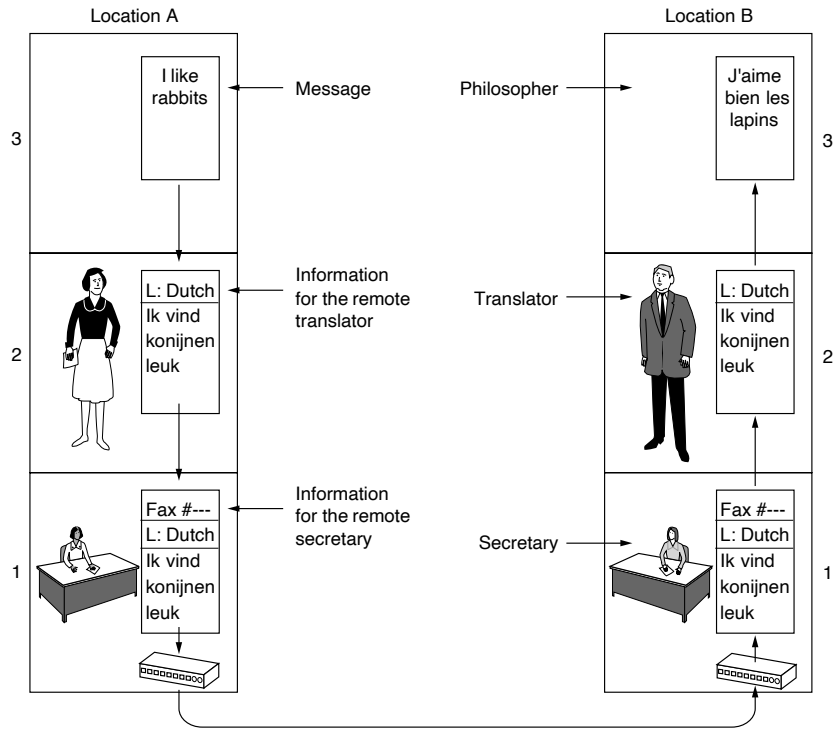

Figure 1-26. The philosopher-translator-secretary architecture.

result to layer 3. The header includes control information, such as addresses, to allow layer 4 on the destination machine to deliver the message. Other examples of control information used in some layers are sequence numbers (in case the lower layer does not preserve message order), sizes, and times.

In many networks, no limit is placed on the size of messages transmitted in the layer 4 protocol, but there is nearly always a limit imposed by the layer 3 protocol. Consequently, layer 3 must break up the incoming messages into smaller units, packets, prepending a layer 3 header to each packet. In this example, M is split into two parts, M_1 and M_2, that will be transmitted separately.

Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2. Layer 2 adds to each piece not only a header but also a trailer and gives the resulting unit to layer 1 for physical transmission. At the receiving machine, the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below n are passed up to layer n.

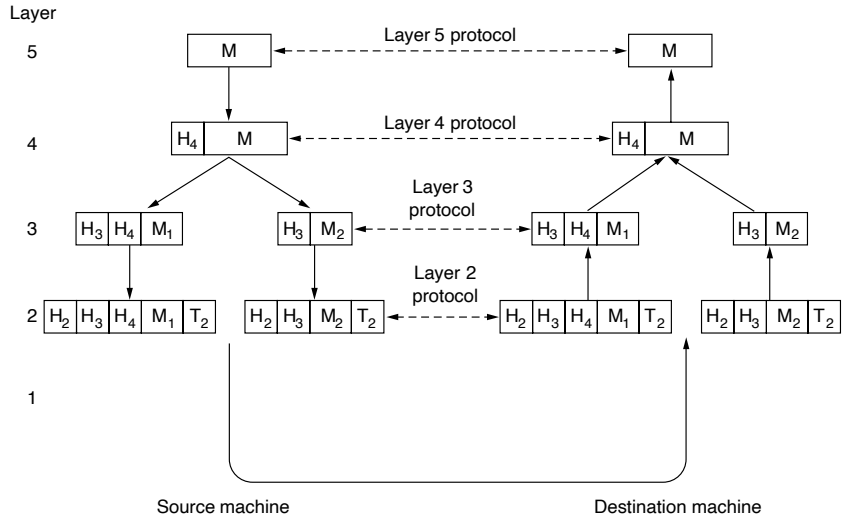

Figure 1-27. Example information flow supporting virtual communication in layer 5.

The important thing to understand about Fig. 1-27 is the relation between the virtual and actual communication and the difference between protocols and interfaces. The peer processes in layer 4, for example, conceptually think of their communication as being ''horizontal,'' using the layer 4 protocol. Each one is likely to have procedures called something like *SendToOtherSide* and *GetFromOtherSide*, even though these procedures actually communicate with lower layers across the 3/4 interface, and not with the other side.

The peer process abstraction is crucial to all network design. Using it, the unmanageable task of designing the complete network can be broken into several smaller, manageable design problems, namely, the design of the individual layers. As a consequence, all real networks use layering.

It is worth pointing out that the lower layers of a protocol hierarchy are frequently implemented in hardware or firmware. Nevertheless, complex protocol algorithms are involved, even if they are embedded (in whole or in part) in hardware.

1.5.3 Connections and Reliability

Layers offer two types of service to the layers above them: connection-oriented and connectionless. They may also offer various levels of reliability.

Connection-Oriented Service

Connection-oriented service is modeled after the telephone system. To talk to someone, you pick up the phone, key in the number, talk, and then hang up. Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out at the other end. In most cases, the order is preserved so that the bits arrive in the order they were sent.

In some cases when a connection is established, the sender, receiver, and subnet conduct a **negotiation** about the parameters to be used, such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a counterproposal. A **circuit** is another name for a connection with associated resources, such as a fixed bandwidth. This dates from the telephone network in which a circuit was a path over copper wire that carried a phone conversation.

Connectionless Service

In contrast to connection-oriented service, **connectionless** service is modeled after the postal system. Each message (letter) carries the full destination address, and each one is routed through the intermediate nodes inside the system independent of all the subsequent messages. There are different names for messages in different contexts; a **packet** is a message at the network layer. When the intermediate nodes receive a message in full before sending it on to the next node, this is called **store-and-forward switching**. The alternative, in which the onward transmission of a message at a node starts before it is completely received by the node, is called **cut-through switching**. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.

Not all applications require connections. For example, spammers send electronic junk mail to many recipients. Unreliable (meaning not acknowledged) connectionless service is often called **datagram** service, in analogy with telegram service, which also does not return an acknowledgement to the sender.

Reliability

Connection-oriented and connectionless services can each be characterized by their reliability. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but sometimes the price that has to be paid for reliability is too high.

A typical situation when a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly and in the same order they were sent. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it were much faster.

Reliable connection-oriented service has two minor variations: message sequences and byte streams. In the former variant, the message boundaries are preserved. When two 1024-byte messages are sent, they arrive as two distinct 1024-byte messages, never as one 2048-byte message. In the latter, the connection is simply a stream of bytes, with no message boundaries. When 2048 bytes arrive at the receiver, there is no way to tell if they were sent as one 2048-byte message, two 1024-byte messages, or 2048 1-byte messages. If the pages of a book are sent over a network to a photo-typesetter as separate messages, it might be important to preserve the message boundaries. On the other hand, to download a movie, a byte stream from the server to the user's computer is all that is needed. Message boundaries (different scenes) within the movie are not relevant.

In some situations, the convenience of not having to establish a connection to send one message is desired, but reliability is essential. The **acknowledged datagram** service can be provided for these applications. It is like sending a registered letter and requesting a return receipt. When the receipt comes back, the sender is absolutely sure that the letter was delivered to the intended party and not lost along the way. Text messaging on mobile phones is an example.

The concept of using unreliable communication may be confusing at first. After all, why would anyone actually prefer unreliable communication to reliable communication? First of all, reliable communication (in our sense, that is, acknowledged) may not be available in a given layer. For example, Ethernet does not provide reliable communication. Packets can occasionally be damaged in transit. It is up to higher protocol levels to recover from this problem. In particular, many reliable services are built on top of an unreliable datagram service. Second, the delays inherent in providing a reliable service may be unacceptable, especially in real-time applications such as multimedia. For these reasons, both reliable and unreliable communication coexist.

In some applications, the transit delays introduced by acknowledgements are unacceptable. One such application is digitized voice traffic (VoIP). It is less disruptive for VoIP users to hear a bit of noise on the line from time to time than to experience a delay waiting for acknowledgements. Similarly, when transmitting a video conference, having a few pixels wrong is no problem, but having the image jerk along as the flow stops and starts to correct errors, or having to wait longer for a perfect video stream to arrive, is irritating.

Still another service is the **request-reply** service. In this service, the sender transmits a single datagram containing a request; the reply contains the answer. Request-reply is commonly used to implement communication in the client-server model: the client issues a request and then the server responds to it. For example, a

mobile phone client might send a query to a map server asking for a list of nearby Chinese restaurants, with the server sending the list.

Figure 1-28 summarizes the types of services discussed above.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

Figure 1-28. Six different types of service.

1.5.4 Service Primitives

A service is formally specified by a set of **primitives** (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets.

The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service. As a minimal example of the service primitives that might provide a reliable byte stream, consider the primitives listed in Fig. 1-29. They will be familiar to fans of the Berkeley socket interface, as the primitives are a simplified version of that interface.

These primitives might be used for a request-reply interaction in a client-server environment. To illustrate how, we sketch a simple protocol that implements the service using acknowledged datagrams.

First, the server executes LISTEN to indicate that it is prepared to accept incoming connections. A common way to implement LISTEN is to make it a blocking system call. After executing the primitive, the server process is blocked (suspended) until a request for connection appears.

Next, the client process executes CONNECT to establish a connection with the server. The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address. The operating system then typically sends a

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Figure 1-29. Six service primitives that provide a simple connection-oriented service.

packet to the peer asking it to connect, as shown by (1) in Fig. 1-30. The client process is suspended until there is a response.

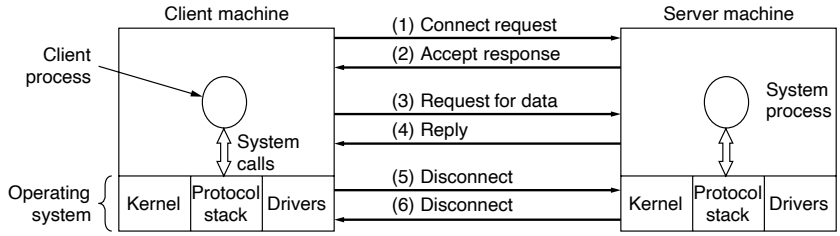

Figure 1-30. A simple client-server interaction using acknowledged datagrams.

When the packet arrives at the server, the operating system sees that the packet is requesting a connection. It checks to see if there is a listener, and if so, it unblocks the listener. The server process can then establish the connection with the ACCEPT call. This sends a response (2) back to the client process to accept the connection. The arrival of this response then releases the client. At this point, the client and server are both running and they have a connection established.

An obvious analogy between this protocol and real life is a customer (client) calling a company's customer service manager. At the start of the day, the service manager sits next to her telephone in case it rings. Later, a client places a call. When the manager picks up the phone, the connection is established.

The next step is for the server to execute RECEIVE to prepare to accept the first request. Normally, the server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the server.

Then the client executes SEND to transmit its request (3) followed by the execution of RECEIVE to get the reply. The arrival of the request packet at the server machine unblocks the server so it can handle the request. After it has done the work,

the server uses SEND to return the answer to the client (4). The arrival of this packet unblocks the client, which can now inspect the answer. If the client has additional requests, it can make them now.

When the client is done, it executes DISCONNECT to terminate the connection (5). Usually, an initial DISCONNECT is a blocking call, suspending the client, and sending a packet to the server saying that the connection is no longer needed. When the server gets the packet, it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6). When the server's packet gets back to the client machine, the client process is released and the connection is broken. In a nutshell, this is how connection-oriented communication works.

Of course, life is not so simple. Many things can go wrong here. The timing can be wrong (e.g., the CONNECT is done before the LISTEN), packets can get lost, and much more. We will look at these issues in great detail later, but for the moment, Fig. 1-30 briefly summarizes how client-server communication might work with acknowledged datagrams so that we can ignore lost packets.

Given that six packets are required to complete this protocol, one might wonder why a connectionless protocol is not used instead. The answer is that in a perfect world it could be, in which case only two packets would be needed: one for the request and one for the reply. However, in the face of large messages in either direction (e.g., a megabyte file), transmission errors, and lost packets, the situation changes. If the reply consisted of hundreds of packets, some of which could be lost during transmission, how would the client know if some pieces were missing? How would the client know whether the last packet actually received was really the last packet sent? Suppose the client wanted a second file. How could it tell packet 1 from the second file from a lost packet 1 from the first file that suddenly found its way to the client? In short, in the real world, a simple request-reply protocol over an unreliable network is often inadequate. In Chap. 3, we will study a variety of protocols in detail that overcome these and other problems. For the moment, suffice it to say that having a reliable, ordered byte stream between processes is sometimes very convenient.

1.5.5 The Relationship of Services to Protocols

Services and protocols are distinct concepts. This distinction is so important that we emphasize it again here. A *service* is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is able to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user. The service uses the lower layer to allow the upper layer to do its work.

A *protocol*, in contrast, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols in order to implement their service definitions. They are free

to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled. This is a key concept that any network designer should understand well.

To repeat this crucial point, services relate to the interfaces between layers, as illustrated in Fig. 1-31. In contrast, protocols relate to the packets sent between peer entities on different machines. It is very important not to confuse the two.

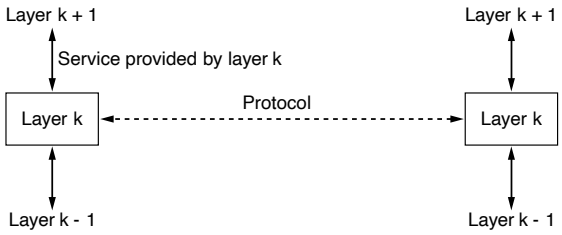

Figure 1-31. The relationship between a service and a protocol.

An analogy with programming languages is worth making. A service is like an abstract data type or an object in an object-oriented language. It defines operations that can be performed on an object but does not specify how these operations are implemented. In contrast, a protocol relates to the *implementation* of the service and as such is not visible to the user of the service.

Many older protocols did not distinguish the service from the protocol. In effect, a typical layer might have had a service primitive SEND PACKET with the user providing a pointer to a fully assembled packet. This arrangement meant that all changes to the protocol were immediately visible to the users. Most network designers now regard such a design as a serious blunder.

1.6 REFERENCE MODELS

Layered protocol design is one of the key abstractions in network design. One of the main questions is defining the functionality of each layer and the interactions between them. Two prevailing models are the TCP/IP reference model and the OSI reference model. We discuss each of them below, as well as the model we use for the rest of this book, which strikes a middle ground between them.

1.6.1 The OSI Reference Model

The OSI model (minus the physical medium) is shown in Fig. 1-32. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used

in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). It is called the ISO **OSI** (**Open Systems Interconnection**) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will call it the **OSI model** for short.

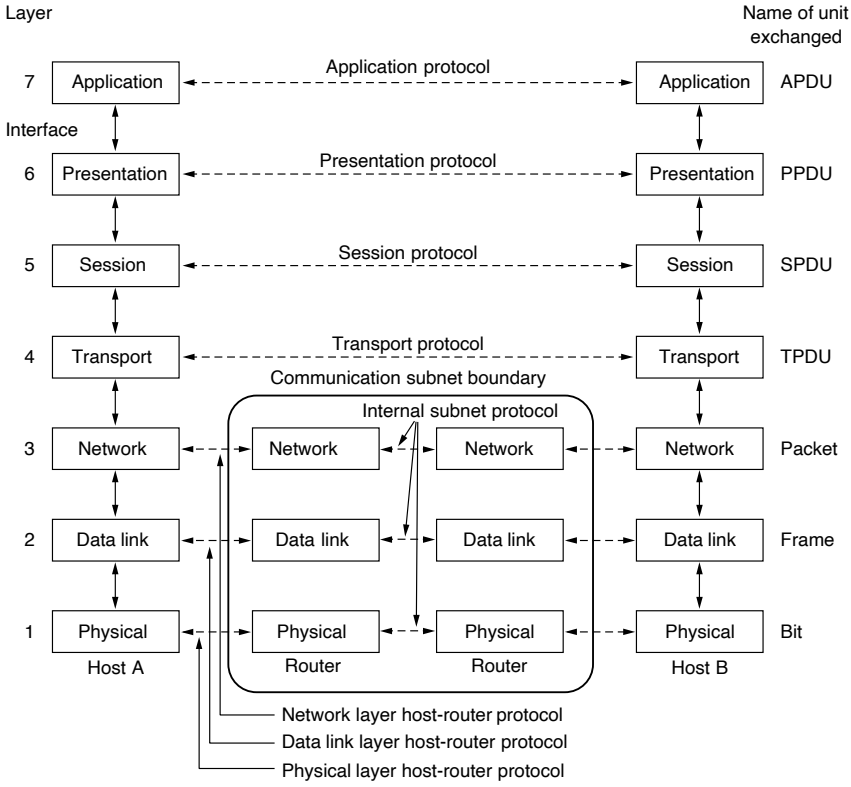

Figure 1-32. The OSI reference model.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.

5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

Three concepts are central to the OSI model:

1. Services.
2. Interfaces.
3. Protocols.

Probably, the biggest contribution of the OSI model is that it makes the distinction between these three concepts explicit. Each layer performs some *services* for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works.

The TCP/IP model did not originally clearly distinguish between services, interfaces, and protocols, although people have tried to retrofit it after the fact to make it more OSI-like.

1.6.2 The TCP/IP Reference Model

The TCP/IP reference model is used in the grandparent of all wide area computer networks, the ARPANET, and its successor, the worldwide Internet. As described earlier, the ARPANET was a research network sponsored by the DoD. It eventually connected hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. Thus, from nearly the beginning, the ability to connect multiple networks in a seamless way was one of the major design goals. This architecture later became known as the **TCP/IP Reference Model**, after its two primary protocols. It was first described by Cerf and Kahn (1974), and later refined and defined as a standard in the Internet community (Braden, 1989). The design philosophy behind the model is discussed by Clark (1988).

Given the DoD's worry that some of its precious hosts, routers, and internetwork gateways might get blown to pieces at a moment's notice by an attack from the Soviet Union, another major goal was that the network be able to survive the loss of subnet hardware, without existing conversations being broken off. In other words, the DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, since applications with divergent requirements were envisioned, ranging from transferring files to real-time speech transmission, a flexible architecture was needed.

The Link Layer

These requirements led to the choice of a packet-switching network based on a connectionless layer that runs across different networks. The lowest layer in the model, the **link layer**, describes what links such as serial lines and classic Ethernet must do to meet the needs of this connectionless internet layer. It is not really a layer at all, in the normal sense of the term, but rather an interface between hosts and transmission links. Early material on the TCP/IP model ignored it.

The Internet Layer

The **internet layer** is the linchpin that holds the whole architecture together. It is shown in Fig. 1-33. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a completely different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that ''internet'' is used here in a generic sense, even though this layer is present in the Internet.

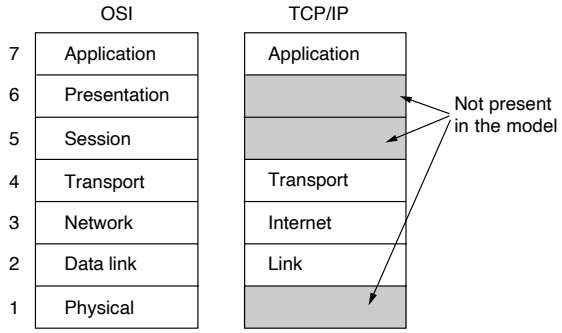

Figure 1-33. The TCP/IP reference model.

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mailbox in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. The letters will probably travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, the fact that each country (i.e., each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users.

The internet layer defines an official packet format and protocol called **IP** (**Internet Protocol**), plus a companion protocol called **ICMP** (**Internet Control Message Protocol**) that helps it function. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly a major issue

here, as is congestion management. The routing problem has largely been solved, but congestion can only be handled with help from higher layers.

The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the **transport layer**. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, **TCP** (**Transmission Control Protocol**), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It segments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

The second protocol in this layer, **UDP** (**User Datagram Protocol**), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own (if any). It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig. 1-34. Since the model was developed, IP has been implemented on many other networks.

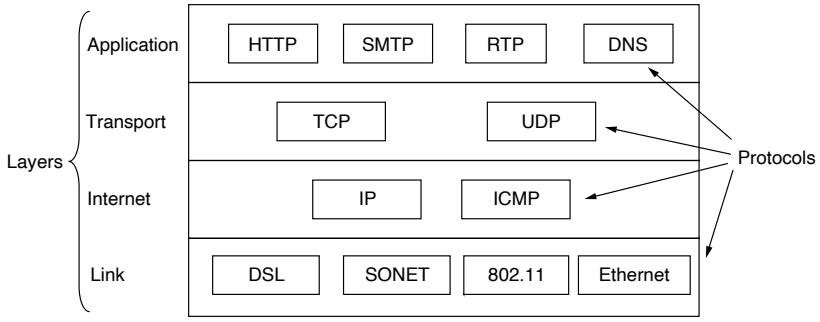

Figure 1-34. The TCP/IP model with some protocols we will study.

The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived. Instead, applications simply include any session and presentation functions that they require. Experience has proven this view correct: these layers are of little use to most applications so they are basically gone forever.

On top of the transport layer is the **application layer**. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP). Many other protocols have been added to these over the years. Some important ones that we will study, shown in Fig. 1-34, include the Domain Name System (DNS), for mapping host names onto their network addresses, HTTP, the protocol for fetching pages on the World Wide Web, and RTP, the protocol for delivering real-time media such as voice or movies.

1.6.3 A Critique of the OSI Model and Protocols

Neither the OSI model and its protocols nor the TCP/IP model and its protocols are perfect. Quite a bit of criticism can be, and has been, directed at both of them. In this section, and the next one, we will look at some of these criticisms. We will begin with OSI and examine TCP/IP afterward.

At the time the second edition of this book was published (1989), it appeared to many experts in the field that the OSI model and its protocols were going to take over the world and push everything else out of their way. This did not happen. Why? A look back at some of the reasons may be useful. They can be summarized as: bad timing, bad design, bad implementations, and bad politics.

Bad Timing

First let us look at reason one: bad timing. The time at which a standard is established is absolutely critical to its success. David Clark of M.I.T. has a theory of standards that he calls the *apocalypse of the two elephants*, which is illustrated in Fig. 1-35.

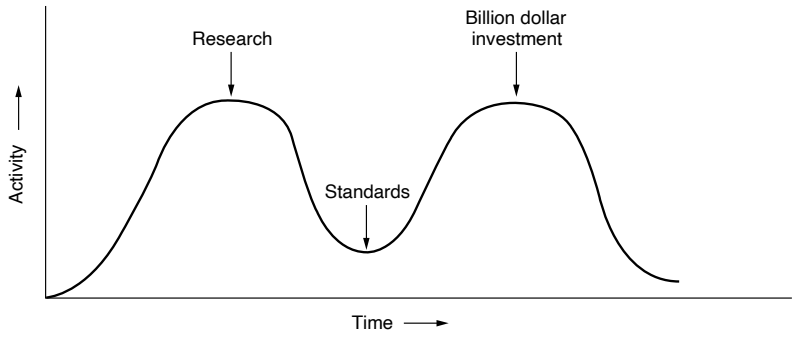

Figure 1-35. The apocalypse of the two elephants.

This figure shows the amount of activity surrounding a new subject. When the subject is first discovered, there is a giant burst of research activity in the form of

research, discussions, papers, and meetings. After a while this activity subsides, corporations discover the subject, and the billion-dollar wave of investment hits.

It is essential that the standards be written in the trough in between the two ''elephants.'' If they are written too early (before the research results are well established), the subject may still be poorly understood; the result is a bad standard. If they are written too late, so many companies may have already made major investments in different ways of doing things that the standards are effectively ignored. If the interval between the two elephants is very short (because everyone is in a hurry to get started), the people developing the standards may get crushed.

It now appears that the standard OSI protocols got crushed. The competing TCP/IP protocols were already in widespread use by research universities by the time the OSI protocols appeared. While the billion-dollar wave of investment had not yet hit, the academic market was large enough that many vendors had begun cautiously offering TCP/IP products. When OSI came around, they did not want to support a second protocol stack until they were forced to, so there were no initial offerings. With every company waiting for every other company to go first, no company went first and OSI never happened.

Bad Design

The second reason that OSI never caught on is that both the model and the protocols are flawed. The choice of seven layers was more political than technical, and two of the layers (session and presentation) are nearly empty, whereas two other ones (data link and network) are overfull.

The OSI model, along with its associated service definitions and protocols, is extraordinarily complex. When piled up, the printed standards occupy a significant fraction of a meter of paper. They are also difficult to implement and inefficient in operation. In this context, a riddle posed by Paul Mockapetris and cited by Rose (1993) comes to mind:

Q: What do you get when you cross a mobster with an international standard?
A: Someone who makes you an offer you can't understand.

In addition to being incomprehensible, another problem with OSI is that some functions, such as addressing, flow control, and error control, reappear again and again in each layer. Saltzer et al. (1984), for example, have pointed out that to be effective, error control must be done in the highest layer, so that repeating it over and over in each of the lower layers is often unnecessary and inefficient.

Bad Implementations

Given the enormous complexity of the model and the protocols, it will come as no surprise that the initial implementations were huge, unwieldy, and slow. Everyone who tried them got burned. It did not take long for people to associate ''OSI''

with ''poor quality.'' Although the products improved in the course of time, the image stuck. Once people think something is bad, its goose is cooked.

In contrast, one of the first implementations of TCP/IP was part of Berkeley UNIX and was quite good (not to mention, free). People began using it quickly, which led to a large user community, which led to improvements and which led to an even larger community. Here, the spiral was upward instead of downward.

Bad Politics

On account of the initial implementation, many people, especially in academia, thought of TCP/IP as part of UNIX, and UNIX in the 1980s in academia was not unlike parenthood (then incorrectly called motherhood) and apple pie.

OSI, on the other hand, was widely thought to be the creature of the European telecommunication ministries, the European Community, and later the U.S. Government. This belief was only partly true, but the very idea of a bunch of government bureaucrats trying to shove a technically inferior standard down the throats of the poor researchers and programmers down in the trenches actually developing computer networks did not aid OSI's cause. Some people viewed this development in the same light as IBM announcing in the 1960s that PL/I was the language of the future, or the DoD correcting this later by announcing that it was actually Ada.

1.6.4 A Critique of the TCP/IP Reference Model and Protocols

The TCP/IP model and protocols also have their problems. First, the model does not clearly distinguish the concepts of services, interfaces, and protocols. Good software engineering practice requires differentiating between the specification and the implementation, something that OSI does very carefully, but TCP/IP does not. Consequently, the TCP/IP model is not much of a guide for designing new networks using new technologies.

Second, the TCP/IP model is not at all general and is poorly suited to describing any protocol stack other than TCP/IP. Trying to use the TCP/IP model to describe Bluetooth, for example, is completely impossible.

Third, the link layer is not really a layer at all in the normal sense of the term as used in the context of layered protocols. It is an interface (between the network and data link layers). The distinction between an interface and a layer is crucial, and one should not be sloppy about it.

Fourth, the TCP/IP model does not distinguish between the physical and data link layers. These are completely different. The physical layer has to do with the transmission characteristics of copper wire, fiber optics, and wireless communication. The data link layer's job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability. A proper model should include both as separate layers. The TCP/IP model does not do this.

Finally, although the IP and TCP protocols were carefully thought out and well implemented, many of the other early protocols were ad hoc, generally produced

by a couple of graduate students hacking away until they got tired. The protocol implementations were then distributed free, which resulted in them becoming widely used, deeply entrenched, and thus hard to replace. Some of them are a bit of an embarrassment now. For example, the virtual terminal protocol, TELNET was designed for a ten-character-per-second mechanical Teletype terminal. It knows nothing of graphical user interfaces and mice. Nevertheless, it is still in use 50 years later.

1.6.5 The Model Used in This Book

As mentioned earlier, the strength of the OSI reference model is the *model* itself (minus the presentation and session layers), which has proven to be exceptionally useful for discussing computer networks. In contrast, the strength of the TCP/IP reference model is the *protocols*, which have been widely used for many years. Since computer scientists like to have their cake and eat it, too, we will use the hybrid model of Fig. 1-36 as the framework for this book.

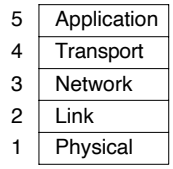

Figure 1-36. The reference model used in this book.

This model has five layers, running from the physical layer up through the link, network and transport layers to the application layer. The physical layer specifies how to transmit bits across different kinds of media as electrical (or other analog) signals. The link layer is concerned with how to send finite-length messages between directly connected computers with specified levels of reliability. Ethernet and 802.11 are examples of link layer protocols.

The network layer deals with how to combine multiple links into networks, and networks of networks, into internetworks so that we can send packets between distant computers. This includes the task of finding the path along which to send the packets. IP is the main example protocol we will study for this layer. The transport layer strengthens the delivery guarantees of the Network layer, usually with increased reliability, and provide delivery abstractions, such as a reliable byte stream, that match the needs of different applications. TCP is an important example of a transport layer protocol.

Finally, the application layer contains programs that make use of the network. Many, but not all, networked applications have user interfaces, such as a Web browser. Our concern, however, is with the portion of the program that uses the network. This is the HTTP protocol in the case of the Web browser. There are also

important support programs in the application layer, such as the DNS, that are used by many applications. These form the glue that makes the network function.

Our chapter sequence is based on this model. In this way, we retain the value of the OSI model for understanding network architectures, but concentrate primarily on protocols that are important in practice, from TCP/IP and related protocols to newer ones such as 802.11, SONET, and Bluetooth.

1.7 STANDARDIZATION

Innovation in Internet technology often depends as much on policy and legal issues as it does on the technology itself. Traditionally, Internet protocols have advanced through a standardization process, which we will now explore.

1.7.1 Standardization and Open Source

Many network vendors and suppliers exist, each with its own ideas of how things should be done. Without coordination, there would be complete chaos, and users would get nothing done. The only way out is to agree on some network standards. Not only do good standards allow different computers to communicate, but they also increase the market for products adhering to the standards. A larger market leads to mass production, economies of scale in manufacturing, better implementations, and other benefits that decrease price and further increase acceptance.

In this section, we will take a quick look at the important but little-known, world of international standardization. But let us first discuss what belongs in a standard. A reasonable person might assume that a standard tells you how a protocol should work so that you can do a good job of implementing it. That person would be wrong.

Standards define what is needed for interoperability: no more, no less. That lets the larger market emerge and also lets companies compete on the basis of how good their products are. For example, the 802.11 standard defines many transmission rates but does not say when a sender should use which rate, which is a key factor in good performance. That is up to whoever makes the product. Often getting to interoperability this way is difficult, since there are many implementation choices and standards that usually define many options. For 802.11, there were so many problems that, in a strategy that has become common practice, a trade group called the **WiFi Alliance** was started to work on interoperability within the 802.11 standard. In the context of software-defined networking, the **ONF** (**Open Networking Foundation**) aims to develop both standards and open-source software implementations of those standards to ensure the interoperability of protocols to control programmable network switches.

A protocol standard defines the protocol over the wire but not the service interface inside the box, except to help explain the protocol. Real service interfaces are

often proprietary. For example, the way TCP interfaces to IP within a computer does not matter for talking to a remote host. It only matters that the remote host speaks TCP/IP. In fact, TCP and IP are commonly implemented together without any distinct interface. That said, good service interfaces, like good **API**s (**Application Programming Interfaces**). are valuable for getting protocols used, and the best ones (such as Berkeley sockets) can become very popular.

Standards fall into two categories: de facto and de jure. **De facto** (Latin for ''from the fact'') standards are those that have just happened, without any formal plan. HTTP, the protocol on which the Web runs, started life as a de facto standard. It was part of early WWW browsers developed by Tim Berners-Lee at CERN, and its use took off with the growth of the Web. Bluetooth is another example. It was originally developed by Ericsson but now everyone is using it.

De jure (Latin for ''by law'') standards, in contrast, are adopted through the rules of some formal standardization body. International standardization authorities are generally divided into two classes: those established by treaty among national governments and those comprising voluntary, non-treaty organizations. In the area of computer network standards, there are several organizations of each type, notably ITU, ISO, IETF, and IEEE, all of which we will discuss below.

In practice, the relationships between standards, companies, and standardization bodies are complicated. De facto standards often evolve into de jure standards, especially if they are successful. This happened in the case of HTTP, which was quickly picked up by IETF. Standards bodies often ratify each others' standards, in what looks like patting one another on the back, to increase the market for a technology. These days, many ad hoc business alliances that are formed around particular technologies also play a significant role in developing and refining network standards. For example, **3GPP** (**Third Generation Partnership Project**) was a collaboration among telecommunications associations that drives the UMTS 3G mobile phone standards.

1.7.2 Who's Who in the Telecommunications World

The legal status of the world's telephone companies varies considerably from country to country. At one extreme is the United States, which has many (mostly very small) privately owned telephone companies. A few more were added with the breakup of AT&T in 1984 (which was then the world's largest corporation, providing telephone service to about 80 percent of America's telephones), and the Telecommunications Act of 1996 that overhauled regulation to foster competition. The idea of fostering competition didn't turn out as planned though. Large telephone companies bought up smaller ones until in most areas there was only one (or at most, two) left.

At the other extreme are countries in which the national government has a complete legal monopoly on all communication, including the mail, telegraph,

telephone, and often radio and television. Much of the world falls into this category. In some cases, the telecommunication authority is a nationalized company, and in others it is simply a branch of the government, usually known as the **PTT** (**Post, Telegraph & Telephone administration**). Worldwide, the trend is toward liberalization and competition and away from government monopoly. Most European countries have now (partially) privatized their PTTs, but elsewhere the process is still only slowly gaining steam.

With all these different suppliers of services, there is clearly a need to provide compatibility on a worldwide scale to ensure that people (and computers) in one country can call their counterparts in another one. Actually, this need has existed for a long time. In 1865, representatives from many European governments met to form the predecessor to today's **ITU** (**International Telecommunication Union**). Its job was to standardize international telecommunications, which in those days meant telegraphy.

Even then it was clear that if half the countries used Morse code and the other half used some other code, there was going to be a problem. When the telephone was put into international service, ITU took over the job of standardizing telephony (pronounced te-LEF-ony) as well. In 1947, ITU became an agency of the United Nations.

ITU has about 200 governmental members, including almost every member of the United Nations. Since the United States does not have a PTT, somebody else had to represent it in ITU. This task fell to the State Department, probably on the grounds that ITU had to do with foreign countries, the State Department's specialty. ITU also has more than 700 sector and associate members. They include telephone companies (e.g., AT&T, Vodafone, Sprint), telecom equipment manufacturers (e.g., Cisco, Nokia, Nortel), computer vendors (e.g., Microsoft, Dell, Toshiba), chip manufacturers (e.g., Intel, Motorola, TI), and other interested companies (e.g., Boeing, CBS, VeriSign).

ITU has three main sectors. We will focus primarily on **ITU-T**, the Telecommunications Standardization Sector, which is concerned with telephone and data communication systems. Before 1993, this sector was called **CCITT**, which is an acronym for its French name, Comité Consultatif International Télégraphique et Téléphonique. **ITU-R**, the Radiocommunications Sector, is concerned with coordinating the use by competing interest groups of radio frequencies worldwide. The other sector is ITU-D, the Development Sector. It promotes the development of information and communication technologies in order to narrow the ''digital divide'' among countries with effective access to the information technologies and countries with limited access.

ITU-T's task is to make technical recommendations about telephone, telegraph, and data communication interfaces. These often become internationally recognized standards, though technically the recommendations are only suggestions that governments can adopt or ignore, as they wish (because governments are like 13-year-old boys—they do not take kindly to being given orders). In practice,

a country that wishes to adopt a telephone standard different from that used by the rest of the world is free to do so, but at the price of cutting itself off from everyone else so no one can call in and no one can call out. This might work for North Korea, but elsewhere it would be a real problem.

The real work of ITU-T is done in its Study Groups. There are currently 11 Study Groups, often as large as 400 people, that cover topics ranging from telephone billing to multimedia services to security. SG 15, for example, standardizes fiber-optic connections to the home. This makes it possible for manufacturers to produce products that work anywhere. To make it possible to get anything at all done, the Study Groups are divided into Working Parties, which are in turn divided into Expert Teams, which are in turn divided into ad hoc groups. Once a bureaucracy, always a bureaucracy.

Despite all this, ITU-T actually does get things done. Since its inception, it has produced more than 3000 recommendations, many of which are widely used in practice. For example, Recommendation H.264 (also an ISO standard known as MPEG-4 AVC) is widely used for video compression, and X.509 public key certificates are used for secure Web browsing and digitally signed email.

As the field of telecommunications completes the transition started in the 1980s from being entirely national to being entirely global, standards will become increasingly important, and more and more organizations will want to become involved in setting them. For more information about ITU, see Irmer (1994).

1.7.3 Who's Who in the International Standards World

International standards are produced and published by **ISO** (**International Standards Organization**†), a voluntary non-treaty organization founded in 1946. Its members are the national standards organizations of the 161 member countries. These members include ANSI (U.S.), BSI (Great Britain), AFNOR (France), DIN (Germany), and 157 others.

ISO issues standards on a truly vast number of subjects, ranging from nuts and bolts (literally) to telephone pole coatings [not to mention cocoa beans (ISO 2451), fishing nets (ISO 1530), women's underwear (ISO 4416), and quite a few other subjects one might not think were subject to standardization]. On issues of telecommunication standards, ISO and ITU-T often cooperate (ISO is a member of ITU-T) to avoid the irony of two official and mutually incompatible international standards.

Over 21,000 standards have been issued, including the OSI standards. ISO has over 200 Technical Committees (TCs), numbered in the order of their creation, each dealing with some specific subject. TC1 literally deals with the nuts and bolts (standardizing screw thread pitches). JTC1 deals with information technology, including networks, computers, and software. It is the first (and so far only) Joint Technical Committee, created in 1987 by merging TC97 with activities in IEC, yet

another standardization body. Each TC has multiple subcommittees (SCs) that are divided into working groups (WGs).

The real work is done largely in the WGs by over 100,000 volunteers worldwide. Many of these ''volunteers'' are assigned to work on ISO matters by their employers, whose products are being standardized. Others are government officials keen on having their country's way of doing things become the international standard. Academic experts also are active in many of the WGs.

The procedure used by ISO for adopting standards has been designed to achieve as broad a consensus as possible. The process begins when one of the national standards organizations feels the need for an international standard in some area. A working group is then formed to come up with a **CD** (**Committee Draft**). The CD is then circulated to all the member bodies, which get 6 months to criticize it. If a substantial majority approves, a revised document, called a **DIS** (**Draft International Standard**), is produced and circulated for comments and voting. Based on the results of this round, the final text of the **IS** (**International Standard**) is prepared, approved, and published. In areas of great controversy, a CD or DIS may have to go through several versions before acquiring enough votes. The whole process can take years.

NIST (**National Institute of Standards and Technology**) is part of the U.S. Department of Commerce. It used to be called the National Bureau of Standards. It issues standards that are mandatory for purchases made by the U.S. Government, except for those of the Department of Defense, which defines its own standards.

Another major player in the standards world is **IEEE** (**Institute of Electrical and Electronics Engineers**), the largest professional organization in the world. In addition to publishing scores of journals and running hundreds of conferences each year, IEEE has a standardization group that develops standards in the area of electrical engineering and computing. IEEE's 802 committee has standardized many kinds of LANs. We will study some of its output later in this book. The actual work is done by a collection of working groups, which are listed in Fig. 1-37. The success rate of the various 802 working groups has been low; having an 802.x number is no guarantee of success. Still, the impact of the success stories (especially 802.3 and 802.11) on the industry and the world has been enormous.

1.7.4 Who's Who in the Internet Standards World

The worldwide Internet has its own standardization mechanisms, very different from those of ITU-T and ISO. The difference can be crudely summed up by saying that the people who come to ITU or ISO standardization meetings wear suits, while the people who come to Internet standardization meetings wear jeans (except when they meet in San Diego, when they wear shorts and T-shirts).

ITU-T and ISO meetings are populated by corporate officials and government civil servants for whom standardization is their job. They regard standardization as

Number	Topic
802.1	Overview and architecture of LANs
802.2	Logical link control
802.3 *	Ethernet
802.4 †	Token bus (was briefly used in manufacturing plants)
802.5 †	Token ring (IBM's entry into the LAN world)
802.6 †	Dual queue dual bus (early metropolitan area network)
802.7 †	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber-optic technologies
802.9 †	Isochronous LANs (for real-time applications)
802.10 †	Virtual LANs and security
802.11 *	Wireless LANs (WiFi)
802.12 †	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number; nobody wanted it
802.14 †	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth, Zigbee)
802.16 †	Broadband wireless (WiMAX)
802.17 †	Resilient packet ring
802.18	Technical advisory group on radio regulatory issues
802.19	Technical advisory group on coexistence of all these standards
802.20	Mobile broadband wireless (similar to 802.16e)
802.21	Media independent handoff (for roaming over technologies)
802.22	Wireless regional area network

Figure 1-37. The 802 working groups. The important ones are marked with *. The ones marked with † gave up and stopped.

a Good Thing and devote their lives to it. Internet people, on the other hand, prefer anarchy as a matter of principle. However, with hundreds of millions of people all doing their own thing, little communication can occur. Thus, standards, however regrettable, are sometimes needed. In this context, David Clark of M.I.T. once made a now-famous remark about Internet standardization consisting of ''rough consensus and running code.''

When the ARPANET was set up, DoD created an informal committee to oversee it. In 1983, the committee was renamed the **IAB** (**Internet Activities Board**) and was given a slighter broader mission, namely, to keep the researchers involved with the ARPANET and the Internet pointed more or less in the same direction, an activity not unlike herding cats. The meaning of the acronym ''IAB'' was later changed to **Internet Architecture Board**.

Each of the approximately ten members of the IAB headed a task force on some issue of importance. The IAB met several times a year to discuss results and

to give feedback to the DoD and NSF, which were providing most of the funding at this time. When a standard was needed (e.g., a new routing algorithm), the IAB members would thrash it out and then announce the change so the graduate students (who were the heart of the software effort) could implement it. Communication was done by a series of technical reports called **RFCs** (**Request For Comments**). RFCs are stored online and can be fetched by anyone interested in them from *www.ietf.org/rfc*. They are numbered in chronological order of creation. Over 8000 now exist. We will refer to many RFCs in this book.

By 1989, the Internet had grown so large that this highly informal style no longer worked. Many vendors by then offered TCP/IP products and did not want to change them just because ten researchers had thought of a better idea. In the summer of 1989, the IAB was reorganized again. The researchers were moved to the **IRTF** (**Internet Research Task Force**), which was made subsidiary to IAB, along with the **IETF** (**Internet Engineering Task Force**). The IAB was populated with people representing a broader range of organizations than just the research community. It was initially a self-perpetuating group, with members serving for a 2-year term and new members being appointed by the old ones. Later, the **Internet Society** was created, populated by people interested in the Internet. The Internet Society is thus in a sense comparable to ACM or IEEE. It is governed by elected trustees who appoint the IAB's members.

The idea of this split was to have the IRTF concentrate on long-term research while the IETF dealt with short-term engineering issues. That way they would stay outof each other's way. The IETF was divided up into working groups, each with a specific problem to solve. The chairs of these working groups initially met as a steering committee to direct the engineering effort. The working group topics include new applications, user information, OSI integration, routing and addressing, security, network management, and standards. Eventually, so many working groups were formed (more than 70) that they were grouped into areas and the area chairs met as the steering committee.

In addition, a more formal standardization process was adopted, patterned after ISOs. To become a **Proposed Standard**, the basic idea must be explained in an RFC and have sufficient interest in the community to warrant consideration. To advance to the **Draft Standard** stage, a working implementation must have been rigorously tested by at least two independent sites for at least 4 months. If the IAB is convinced that the idea is sound and the software works, it can declare the RFC to be an **Internet Standard**. Some Internet Standards have become DoD standards (MIL-STD), making them mandatory for DoD suppliers.

For Web standards, the **World Wide Web Consortium** (**W3C**) develops protocols and guidelines to facilitate the long-term growth of the Web. It is an industry consortium led by Tim Berners-Lee and set up in 1994 as the Web really begun to take off. W3C now has almost 500 companies, universities, and other organizations as members and has produced well over 100 W3C Recommendations, as its standards are called, covering topics such as HTML and Web privacy.

1.8 POLICY, LEGAL, AND SOCIAL ISSUES

Like the printing press 500 years ago, computer networks allow ordinary citizens to distribute and view content in ways that were not previously possible. But along with the good comes the bad, as these new capabilities are accompanied by many unsolved social, political, and ethical issues. We will provide a brief survey in this section; in each chapter in the book, we will provide some specific policy, legal, and social issues that pertain to specific technologies, where appropriate. Here, we introduce some of the higher level policy and legal concerns that are now affecting a range of areas in Internet technology, including traffic prioritization, data collection and privacy, and control over free speech online.

1.8.1 Online Speech

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Furthermore, opinions need not be limited to text; people can easily share high-resolution color photographs and video clips on these platforms. In some cases, such as child pornography or incitement to terrorism, the speech may also be illegal.

The ability of social media and so-called **user-generated content** platforms to act as a conduit for illegal or offensive speech has raised important questions concerning the role of these platforms in moderating the content that is hosted on these platforms. For a long time, platforms such as Facebook, Twitter, YouTube, and other user-generated content platforms have enjoyed considerable immunity from prosecution when this content is hosted on their sites. In the United States, for example, Section 230 of the **Communications Decency Act** protects these platforms from federal criminal prosecution should any illegal content be found on their sites. For many years, these social media platforms have claimed that they are merely a platform for information, akin to a printing press, and should not be held liable for the content that they host. As these platforms have increasingly curated, prioritized, and personalized the content that they show to individual users, however, the argument that these sites are merely ''platforms'' has begun to erode.

In both the United States and Europe, for example, the pendulum is beginning to swing, with laws being passed that would hold these platforms accountable for certain genres of illegal online content, such as that related to online sex trafficking. The rise of automated, machine-learning-based content classification algorithms is also leading some advocates to hold the social media platforms accountable for a wider range of content, since these algorithms purport to be able to

automatically detect unwanted content, from copyright violations to hate speech. The reality, however, is more complicated because these algorithms can generate false positives. If a platform's algorithm falsely classifies content as offensive or illegal and automatically takes it down, this action may be considered an censorship or an affront to free speech. If the laws mandate that the platforms take these types of automated actions, then they may ultimately be automating censorship.

The recording and film industries often advocate for laws that would require the use of automated content moderation technologies. In the United States, representatives from these industries regularly issue **DMCA takedown notices** (after the **Digital Millennium Copyright Act**), which threaten legal action if the party in question does not take action and remove the content. Importantly, the ISP or content provider is not held liable for copyright infringement if they pass on the takedown notice to the person who infringed. The ISP or content provider does not actively have to seek out content that violates copyright—that onus falls on the copyright holder (e.g., the record label or movie producer). Because it is challenging to find and identify copyrighted content, the copyright holders understandably continue to push for laws that would shift the onus back to the ISPs and content providers.

1.8.2 Net Neutrality

One of the more prominent legal and policy questions over the past fifteen years has been the extent to which ISPs can block or prioritize content on their own networks. The notion that ISPs should provide equal quality of service to a given type of application traffic, regardless of who is sending that content, is often referred to as **network neutrality** (Wu, 2003).

The basic tenets of net neutrality amount to the following four rules: (1) No blocking, (2) No throttling, (3) No paid prioritization, and (4) Transparency about reasonable network management practices that might be seen as violating any of the first three rules. Note that net neutrality does not prevent an ISP from prioritizing any traffic. As we will see in later chapters, in some cases it may make sense for an ISP to prioritize real-time traffic (e.g., gaming and video conferencing) over other non-interactive traffic (e.g., a large file backup). The rules typically make exception for such ''reasonable network management practices.'' What is a ''reasonable'' network management practice may be arguable, of course. What the rules are intended to prevent are situations where an ISP blocks or throttles traffic as an anti-competitive practice. Specifically, the rules are intended to prevent an ISP from blocking or throttling VoIP traffic if it competes with its own Internet telephony offering (as occurred when AT&T blocked Apple's FaceTime), or when a video service (e.g., Netflix) competes with its own video-on-demand offering.

Although at first the principle of net neutrality may appear straightforward, the legal and policy nuances are significantly more complicated, especially given how

laws and networks differ between countries. For example, one of the legal questions in the United States concerns who has the authority to enforce net neutrality rules. For example, various court rulings over the past decade have granted and subsequently revoked the authority of the Federal Communications Commission (FCC) to enforce net neutrality rules on ISPs. Much of the debate in the United States centers on whether an ISP should be classified as a ''common carrier'' service, akin to a public utility, or whether it should be considered an information service, with the likes of Google and Facebook. As many of these companies offer products in an increasingly diverse set of markets, it is becoming harder to classify a company into one category or another. On June 11, 2018, net neutrality was abolished in the entire United States by order of the FCC. However, some states may adopt their own net neutrality rules statewide.

A topic that relates to network neutrality and is prominent in many countries around the world is the practice of **zero rating**, whereby an ISP might charge its subscribers according to data usage but grant an exemption (i.e., ''zero rate'') for a particular service. For example, the ISP might charge its subscribers for streaming Netflix, but allow unlimited streaming of other video services that it wants to promote. In some countries, mobile carriers use zero rating as a differentiator: for example, a mobile carrier might zero rate Twitter as a promotion to try to attract subscribers from other carriers. Another example of zero rating is Facebook's ''Free Basics'' service, which allows ISP subscribers free, unmetered access to a bundle of sites and services that Facebook packages as part of a free offering. Many parties see these offerings as running afoul of net neutrality, since they offer preferential access to some services and applications over others.

1.8.3 Security

The Internet was designed so that anyone could easily connect to it and begin sending traffic. This open design not only spurred a wave of innovation, but it also has made the Internet a platform for attacks of unprecedented scale and scope. We will explore security in detail in Chap. 8.

One of the most prevalent and pernicious type of attack is a **DDoS** (**Distributed Denial of Service**) attack, whereby many machines on the network send traffic towards a victim machine in an attempt to exhaust its resources. There are many different types of DDoS attacks. The simplest form of DDoS attack is one where a large number of compromised machines, sometimes referred to as a **botnet**, all send traffic towards a single victim. DDoS attacks have typically been launched from compromised general-purpose machines (e.g., laptops and servers), but the proliferation of insecure IoT devices has now created a brand-new vector for launching DDoS attacks. Can a coordinated attack by a million Internet-connected smart toasters take down Google? Unfortunately, much of the IoT industry in particular is unconcerned with software security, and so defending against attacks coming from these highly insecure devices currently falls on network operators.

New incentive or regulatory structures may be necessary to discourage users from connecting insecure IoT devices to the network. In general, many Internet security problems are related to incentives.

Spam email (or unwanted electronic mail) now constitutes more than 90% of all email traffic because spammers have collected millions of email addresses and would-be marketers can cheaply send computer-generated messages to them. Fortunately, filtering software is able to read and discard the spam generated by other computers. Early spam filtering software relied largely on the contents of email messages to differentiate unwanted spam from legitimate emails, but spammers quickly found their way around those filters, since it is relatively easy to generate 100 ways of spelling Viagra. On the other hand, properties of the email message such as the IP address of the sender and receiver, as well as email sending patterns, turn out to be useful distinguishing characteristics that are much more robust to evasion.

Some email spam is simply annoying. Other email messages, on the other hand, may be attempts to launch large-scale scams or steal your personal information, such as your passwords or bank account information. **Phishing** messages masquerade as originating from a trustworthy party, for example, your bank, to try to trick you into revealing sensitive information, for example, credit card numbers. Identity theft is becoming a serious problem as thieves collect enough information about a victim to obtain credit cards and other documents in the victim's name.

1.8.4 Privacy

As computer networks and the devices that we connect to them proliferate, it is becoming increasingly easier for various parties to collect data about how each of us uses the network. Computer networks make it very easy to communicate, but they also make it easy for the people who run the network to snoop on the traffic. A wide range of parties can collect data about your Internet use, including your Internet service provider, your mobile phone carrier, applications, Web sites, cloud hosting services, content delivery networks, device manufacturers, advertisers, and Web tracking software vendors.

One prominent practice by many Web sites and application providers is the practice of **profiling** and **tracking** users by collecting data about their network behavior over time. One way that advertisers track users is by placing small files called **cookies** that Web browsers store on users' computers. Cookies allow advertisers and tracking companies to track users' browsing behavior and activities from one site to another. More sophisticated tracking mechanisms have also been developed in recent years, such as **browser fingerprinting**; it turns out that the configuration of your browser is unique enough to you that a company can use code on its Web page to extract your browser settings and determine your unique identity with high probability. Companies that provide Web-based services also maintain large amounts of personal information about their users that allows them to study user

activities directly. For example, Google can read your email and show you advertisements based on your interests if you use its email service, **Gmail**.

The rise of mobile services has also made **location privacy** a growing concern (Beresford and Stajano, 2003). Your mobile operating system vendor has access to precise location information, including your geographic coordinates and even your altitude, by virtue of the readings from the phone's barometric pressure sensor. For example, a vendor of the Android mobile phone operating system, Google, can determine that your precise location within a building or shopping mall so that it can serve you advertisements based on the store that you're walking past. Mobile carriers can also get information about your geographic location by determining which cellular tower that your phone is communicating with.

Various technologies, ranging from VPNs to anonymous browsing software such as the Tor browser, aim to improve user privacy by obfuscating the source of user traffic. The level of protection that each of these systems provides depends on the properties of the system. For example, a VPN provider may prevent your ISP from seeing any of your unencrypted Internet traffic, but the operator of the VPN service can still see the unencrypted traffic. Tor may offer an additional layer of protection, but there are varying assessments of its effectiveness, and many researchers have noted its weaknesses, particularly when a single entity controls large parts of the infrastructure. Anonymous communication may provide students, employees, and citizens a way to blow the whistle on illegal behavior without fear of reprisal. On the other hand, in the United States and most other democracies, the law specifically permits an accused person the right to confront and challenge his accuser in court so anonymous accusations cannot be used as evidence. Computer networks raise new legal problems when they interact with old laws. One interesting ongoing legal question concerns access to data. For example, what determines whether a government should be able to access data about its citizens? If the data resides in another country, is that data protected from search? If data traverses a country, to what extent does it become subject to those countries' laws? Microsoft grappled with these questions in a Supreme Court case, where the U.S. government is attempting to gain access about U.S. citizens on Microsoft servers located in Ireland. It is likely that the ''borderless'' nature of the Internet will continue to raise questions at the intersection of law and technology for years to come.

1.8.5 Disinformation

The Internet makes it possible to find information quickly, but a great deal of it is ill-considered, misleading, or downright wrong. That medical advice you plucked from the Internet about the pain in your chest may have come from a Nobel Prize winner or from a high-school dropout. There is increasing concern about how citizens around the world find information about news and current events. The 2016 presidential election in the United States, for example, saw the

rise of so-called ''fake news,'' whereby certain parties explicitly crafted false stories with the goal of tricking readers into believing things that never happened. **Disinformation** campaigns have presented network and platform operators with new challenges. First, how does one define disinformation in the first place? Second, can disinformation be reliably detected? Finally, what should a network or platform operator do about it once it is detected?

1.9 METRIC UNITS

To avoid any confusion, it is worth stating explicitly that in this book, as in computer science in general, metric units are used instead of traditional English units (the furlong-stone-fortnight system). The principal metric prefixes are listed in Fig. 1-38. The prefixes are typically abbreviated by their first letters, with the units greater than 1 capitalized (KB, MB, etc.). One exception (for historical reasons) is kbps for kilobits/sec. Thus, a 1-Mbps communication line transmits 10^6 bits/sec and a 100-psec (or 100-ps) clock ticks every 10^{-10} seconds. Since milli and micro both begin with the letter ''m,'' a choice had to be made. Normally, ''m'' is used for milli and ''μ'' (the Greek letter mu) is used for micro.

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figure 1-38. The principal metric prefixes.

It is also worth pointing out that for measuring memory, disk, file, and database sizes, in common industry practice, the units have slightly different meanings. There, kilo means 2^{10} (1024) rather than 10^3 (1000) because memories are always a power of two. Thus, a 1-KB memory contains 1024 bytes, not 1000 bytes. Note also the capital ''B'' in that usage to mean ''bytes'' (units of eight bits), instead of a lowercase ''b'' that means ''bits.'' Similarly, a 1-MB memory contains 2^{20} (1,048,576) bytes, a 1-GB memory contains 2^{30} (1,073,741,824) bytes, and a 1-TB database contains 2^{40} (1,099,511,627,776) bytes. However, a 1-kbps communication line transmits 1000 bits per second and a 10-Mbps LAN runs at 10,000,000 bits/sec because these speeds are not powers of two. Unfortunately, many people

tend to mix up these two systems, especially for disk sizes. To avoid ambiguity, in this book, we will use the symbols KB, MB, GB, and TB for 2^{10}, 2^{20}, 2^{30}, and 2^{40} bytes, respectively, and the symbols kbps, Mbps, Gbps, and Tbps for 10^3, 10^6, 10^9, and 10^{12} bits/sec, respectively.

1.10 OUTLINE OF THE REST OF THE BOOK

This book discusses both the principles and practice of computer networking. Most chapters start with a discussion of the relevant principles, followed by a number of examples that illustrate these principles. These examples are usually taken from the Internet and wireless networks such as the mobile phone network since these are both important and very different. Other examples will be given where relevant.

The book is structured according to the hybrid model of Fig. 1-36. Starting with Chapyer 2, we begin working our way up the protocol hierarchy beginning at the bottom. We provide some background in the field of data communication that covers both wired and wireless transmission systems. This material is concerned with how to deliver information over physical channels, although we cover only the architectural rather than the hardware aspects. Several examples of the physical layer, such as the public switched telephone network, the mobile telephone network, and the cable television network are also discussed.

Chapters 3 and 4 discuss the data link layer in two parts. Chapter 3 looks at the problem of how to send packets across a link, including error detection and correction. We look at DSL (used for broadband Internet access over phone lines) as a real-world example of a data link protocol.

In Chapter 4, we examine the medium access sublayer. This is the part of the data link layer that deals with how to share a channel between multiple computers. The examples we look at include wireless, such as 802.11 and wired LANs such as Ethernet. Link layer switches that connect LANs, such as switched Ethernet, are also discussed here.

Chapter 5 deals with the network layer, especially routing. Many routing algorithms, both static and dynamic, are covered. Even with good routing algorithms, though, if more traffic is offered than the network can handle, some packets will be delayed or discarded. We discuss this issue from how to prevent congestion to how to guarantee a certain quality of service. Connecting heterogeneous networks to form internetworks also leads to numerous problems that are discussed here. The network layer in the Internet is given extensive coverage.

Chapter 6 deals with the transport layer. Much of the emphasis is on connection-oriented protocols and reliability, since many applications need these. Both Internet transport protocols, UDP and TCP, are covered in detail, as are their performance issues, especially that of TCP, one of the Internet's key protocols.

Chapter 7 deals with the application layer, its protocols, and its applications. The first topic is DNS, which is the Internet's telephone book. Next comes email, including a discussion of its protocols. Then we move on to the Web, with detailed discussions of static and dynamic content, and what happens on the client and server sides. We follow this with a look at networked multimedia, including streaming audio and video. Finally, we discuss content-delivery networks, including peer-to-peer technology.

Chapter 8 is about network security. This topic has aspects that relate to all layers, so it is easiest to treat it after all the layers have been thoroughly explained. The chapter starts with an introduction to cryptography. Later, it shows how cryptography can be used to secure communication, email, and the Web. The chapter ends with a discussion of some areas in which security collides with privacy, freedom of speech, censorship, and other social issues.

Chapter 9 contains an annotated list of suggested readings arranged by chapter. It is intended to help those readers who would like to pursue their study of networking further. The chapter also has an alphabetical bibliography of all the references cited in this book.

The authors' Web sites:

https://www.pearsonhighered.com/tanenbaum (*https://www.pearsonhighered.com/tanenbaum*)
https://computernetworksbook.com

have additional information that may be of interest.

1.11 SUMMARY

Computer networks have many uses, both for companies and for individuals, in the home and while on the move. Companies use networks of computers to share corporate information, typically using the client-server model with employee desktops acting as clients accessing powerful servers in the machine room. For individuals, networks offer access to a variety of information and entertainment resources, as well as a way to buy and sell products and services. Individuals often access the Internet via their phone or cable providers at home, though increasingly wireless access is used for laptops and phones. Technology advances are enabling new kinds of mobile applications and networks with computers embedded in appliances and other consumer devices. The same advances raise social issues such as privacy concerns.

Roughly speaking, networks can be divided into LANs, MANs, WANs, and internetworks. LANs typically cover a building and operate at high speeds. MANs usually cover a city. An example is the cable television system, which is now used by many people to access the Internet. WANs may cover a country or a continent. Some of the technologies used to build these networks are point-to-point (e.g., a cable) while others are broadcast (e.g., wireless). Networks can be interconnected with routers to form internetworks, of which the Internet is the largest and most

important example. Wireless networks, for example, 802.11 LANs and 4G mobile telephony, are also becoming extremely popular.

Network software is built around protocols, which are rules by which processes communicate. Most networks support protocol hierarchies, with each layer providing services to the layer above it and insulating them from the details of the protocols used in the lower layers. Protocol stacks are typically based either on the OSI model or on the TCP/IP model. Both have link, network, transport, and application layers, but they differ on the other layers. Design issues include reliability, resource allocation, growth, security, and more. Much of this book deals with protocols and their design.

Networks provide various services to their users. These services can range from connectionless best-efforts packet delivery to connection-oriented guaranteed delivery. In some networks, connectionless service is provided in one layer and connection-oriented service is provided in the layer above it.

Well-known networks include the Internet, the mobile telephone network, and 802.11 LANs. The Internet evolved from the ARPANET, to which other networks were added to form an internetwork. The present-day Internet is actually a collection of many thousands of networks that use the TCP/IP protocol stack. The mobile telephone network provides wireless and mobile access to the Internet at speeds of multiple Mbps, and, of course, carries voice calls as well. Wireless LANs based on the IEEE 802.11 standard are deployed in many homes, hotels, airports, and restaurants, and can provide connectivity at rates of 1 Gbps or more. Wireless networks are also seeing an element of convergence, as evident in proposals such as LTE-U, which would allow cellular network protocols to operate in the unlicensed spectrum alongside 802.11.

Enabling multiple computers to talk to each other requires a large amount of standardization, both in the hardware and software. Organizations such as ITU-T, ISO, IEEE, and IAB manage different parts of the standardization process.

PROBLEMS

1. Imagine that you have trained your St. Bernard, Bernie, to carry a box of three 8-mm tapes instead of a flask of brandy. (When your disk fills up, you consider that an emergency.) These tapes each contain 10 gigabytes. The dog can travel to your side, wherever you may be, at 18 km/hour. For what range of distances does Bernie have a higher data rate than a transmission line whose data rate (excluding overhead) is 150 Mbps? How does your answer change if (i) Bernie's speed is doubled; (ii) each tape capacity is doubled; (iii) the data rate of the transmission line is doubled.

2. An alternative to a LAN is simply a big timesharing system with terminals for all users. Give two advantages of a client-server system using a LAN.

3. The performance of a client-server system is strongly influenced by two major network characteristics: the bandwidth of the network (i.e., how many bits/sec it can transport) and the latency (i.e., how many seconds it takes for the first bit to get from the client to

the server). Give an example of a network that exhibits high bandwidth but also high latency. Then give an example of one that has both low bandwidth and low latency.

4. Besides bandwidth and latency, what other parameter is needed to give a good characterization of the quality of service offered by a network used for (i) digitized voice traffic? (ii) video traffic? (iii) financial transaction traffic?

5. A factor in the delay of a store-and-forward packet-switching system is how long it takes to store and forward a packet through a switch. If switching time is 20 μsec, is this likely to be a major factor in the response of a client-server system where the client is in New York and the server is in California? Assume the propagation speed in copper and fiber to be 2/3 the speed of light in vacuum.

6. A client-server system uses a satellite network, with the satellite at a height of 40,000 km. What is the best-case delay in response to a request?

7. Now that almost everyone has a home computer or mobile device connected to a computer network, instant public referendums on important pending legislation will become possible. Ultimately, existing legislatures could be eliminated, to let the will of the people be expressed directly. The positive aspects of such a direct democracy are fairly obvious; discuss some of the negative aspects.

8. Five routers are to be connected in a point-to-point subnet. Between each pair of routers, the designers may put a high-speed line, a medium-speed line, a low-speed line, or no line. If it takes 50 ms of computer time to generate and inspect each topology, how long will it take to inspect all of them?

9. A group of $2^n - 1$ routers are interconnected in a centralized binary tree, with a router at each tree node. Router i communicates with router j by sending a message to the root of the tree. The root then sends the message back down to j. Derive an approximate expression for the mean number of hops per message for large n, assuming that all router pairs are equally likely.

10. A disadvantage of a broadcast subnet is the capacity wasted when multiple hosts attempt to access the channel at the same time. As a simplistic example, suppose that time is divided into discrete slots, with each of the n hosts attempting to use the channel with probability p during each slot. What fraction of the slots will be wasted due to collisions?

11. What are two reasons for using layered protocols? What is one possible disadvantage of using layered protocols?

12. Match the layers—Link, Network, and Transport—with the guarantees that each layer could provide to higher layers.

Guarantee	Layer
Best effort delivery	Network
Reliable Delivery	Transport
In-order Delivery	Transport
Byte-stream abstraction	Transport
Point-to-point link abstraction	Data link

13. Suppose that two network endpoints have a round-trip time of 100 milliseconds, and that the sender transmits five packets every round trip. What will be the sender's transmission rate for this round-trip time, assuming 1500-byte packets? Give your answer in bytes per second

14. The president of the Specialty Paint Corp. gets the idea to work with a local beer brewer to produce an invisible beer can (as an anti-litter measure). The president tells her legal department to look into it, and they in turn ask engineering for help. As a result, the chief engineer calls his counterpart at the brewery to discuss the technical aspects of the project. The engineers then report back to their respective legal departments, which then confer by telephone to arrange the legal aspects. Finally, the two corporate presidents discuss the financial side of the deal. What principle of a multilayer protocol in the sense of the OSI model does this communication mechanism violate?

15. What is the principal difference between connectionless communication and connection-oriented communication? Give one example of a protocol that uses (i) connectionless communication; (ii) connection-oriented communication.

16. Two networks each provide reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are these identical? If so, why is the distinction made? If not, give an example of how they differ.

17. What does ''negotiation'' mean when discussing network protocols? Give an example.

18. In Fig. 1-31, a service is shown. Are any other services implicit in this figure? If so, where? If not, why not?

19. In some networks, the data link layer handles transmission errors by requesting that damaged frames be retransmitted. If the probability of a frame's being damaged is p, what is the mean number of transmissions required to send a frame? Assume that acknowledgements are never lost.

20. Which of the OSI layers and TCP/IP layers handles each of the following:
(a) Dividing the transmitted bit stream into frames.
(b) Determining which route through the subnet to use.

21. If the unit exchanged at the data link level is called a frame and the unit exchanged at the network level is called a packet, do frames encapsulate packets or do packets encapsulate frames? Explain your answer.

22. A system has an n-layer protocol hierarchy. Applications generate messages of length M bytes. At each of the layers, an h-byte header is added. What fraction of the network bandwidth is filled with headers?

23. List two ways in which the OSI reference model and the TCP/IP reference model are the same. Now list two ways in which they differ.

24. What is the main difference between TCP and UDP?

25. The subnet of Fig. 1-12(b) was designed to withstand a nuclear war. How many bombs would it take to partition the nodes into two disconnected sets? Assume that any bomb wipes out a node and all of the links connected to it.

26. The Internet is roughly doubling in size every 18 months. Although no one really knows for sure, one estimate put the number of hosts on it a 1 billion in 2018. Use these data to compute the expected number of Internet hosts in the year 2027. Do you believe this? Explain why or why not.

27. When a file is transferred between two computers, two acknowledgement strategies are possible. In the first one, the file is chopped up into packets, which are individually acknowledged by the receiver, but the file transfer as a whole is not acknowledged. In the second one, the packets are not acknowledged individually, but the entire file is acknowledged when it arrives. Discuss these two approaches.

28. Mobile phone network operators need to know where their subscribers' mobile phones (hence their users) are located. Explain why this is bad for users. Now give reasons why this is good for users.

29. How long was a bit in the original 802.3 standard in meters? Use a transmission speed of 10 Mbps and assume the propagation speed of the signal in coax is 2/3 the speed of light in vacuum.

30. An image is 1600 × 1200 pixels with 3 bytes/pixel. Assume the image is uncompressed. How long does it take to transmit it over a 56-kbps modem channel? Over a 1-Mbps cable modem? Over a 10-Mbps Ethernet? Over 100-Mbps Ethernet? Over gigabit Ethernet?

31. Ethernet and wireless networks have some similarities and some differences. One property of Ethernet is that only one frame at a time can be transmitted on an Ethernet. Does 802.11 share this property with Ethernet? Discuss your answer.

32. Wireless networks are easy to install, which makes them inexpensive since installation costs usually far overshadow equipment costs. Nevertheless, they also have some disadvantages. Name two of them.

33. List two advantages and two disadvantages of having international standards for network protocols.

34. When a system has a permanent part and a removable part (such as a CD-ROM drive and the CD-ROM), it is important that the system be standardized, so that different companies can make both the permanent and removable parts and everything still works together. Give three examples outside the computer industry where such international standards exist. Now give three areas outside the computer industry where they do not exist.

35. Suppose the algorithms used to implement the operations at layer k is changed. How does this impact operations at layers $k - 1$ and $k + 1$?

36. Suppose there is a change in the service (set of operations) provided by layer k. How does this impact services at layers k-1 and k+1?

37. Match each of the protocols visible in Fig. 1-0 with the correct layer in Fig. 1-36. Explain your answers.

38. Provide a list of reasons for why the response time of a client may be larger than the best-case delay.

39. Find out what networks are used at your school or place of work. Describe the network types, topologies, and switching methods used there.

40. The *ping* program allows you to send a test packet to a given location and see how long it takes to get there and back. Try using *ping* to see how long it takes to get from your location to several known locations. From these data, plot the one-way transit time over the Internet as a function of distance. It is best to use universities since the location of their servers is known very accurately. For example, *berkeley.edu* is in Berkeley, California; *mit.edu* is in Cambridge, Massachusetts; *vu.nl* is in Amsterdam; The Netherlands; *www.usyd.edu.au* is in Sydney, Australia; and *www.uct.ac.za* is in Cape Town, South Africa.

41. Go to IETF's Web site, *www.ietf.org*, to see what they are doing. Pick a project you like and write a half-page report on the problem and the proposed solution.

42. Standardization is very important in the network world. ITU and ISO are the main official standardization organizations. Go to their respective Web sites, *www.itu.org* and *www.iso.org*, and learn about their standardization work. Write a short report about the kinds of things they have standardized.

43. The Internet has a large number of networks. Their arrangement determines the topology of the Internet. A considerable amount of information about the Internet topology is available on line. Use a search engine to find out more about the Internet topology and write a short report summarizing your findings.

44. Search the Internet to find out some of the important peering points used for routing packets in the Internet at present.

45. Write a program that implements message flow from the top layer to the bottom layer of the 7-layer protocol model. Your program should include a separate protocol function for each layer. Protocol headers are sequence up to 64 characters. Each protocol function has two parameters: a message passed from the higher layer protocol (a char buffer) and the size of the message. This function attaches its header in front of the message, prints the new message on the standard output, and then invokes the protocol function of the lower-layer protocol. Program input is an application message.

This page is intentionally left blank