1

Introduction To Networking

The Alcatel-Lucent NRS I exam topics covered in this chapter include the following:

- The significance of the ARPANET
- The problems with having different protocols, and the solutions
- How the Internet evolved from a military-based network to a research-based network and then into a commercial network
- An overview of the modern Internet
- Differences between an Internet provider and a content provider
- Differences between traditional and modern ISP services
- The advantages of protocol layering
- The characteristics of the TCP/IP protocol layers, and how the layers work together
- The definition and development of the OSI Reference Model
- The similarities between the TCI/IP and OSI models of protocol

This chapter provides an introduction to the history and principles that underlie the Internet, the biggest network in the world. It is important that you have a foundational understanding of the hardware and software components that constitue the Internet in order to fully appreciate the remaining topics in the chapters that follow. The Internet has gone through large evolutionary changes in its lifetime, and these changes provide key insights into modern networking principles and design philosophies. We also discuss the development of the TCP/IP protocol, protocol layering, and the relationship of the OSI model to modern networking.

Pre-Assessment

The following assessment questions will help you understand what areas of the chapter you should review in more detail to prepare for the exam.

1. The original network that ultimately became the Internet was called
   - **A.** NSFNET
   - **B.** ARPANET
   - **C.** DoDnet
   - **D.** DARPA
2. The primary organization behind the development of the original Internet was
   - **A.** IBM
   - **B.** Digital Equipment Corporation (DEC)
   - **C.** Stanford University
   - **D.** the U.S. Department of Defense
3. Which of the following was *not* a primary design concern during the development of the original Internet?
   - **A.** Reliability
   - **B.** Bandwidth
   - **C.** Interoperability
   - **D.** Support for diverse network mediums

4. Which of the following was *not* a reason TCP was a superior transport protocol to NCP?

   A. Support for global addressing

   B. Support for end-to-end checksums

   C. Support for applications such as email

   D. Support for fragmentation and reassembly

5. Which of the following OSI layers is *not* paired with the correct implementation?

   A. Layer 7—Email

   B. Layer 3—TCP

   C. Layer 4—UDP

   D. Layer 2—PPP

You will find the answers to each of these questions in Appendix A. You can also download all of the CD materials for this book at `http://booksupport.wiley.com` to take all the assessment tests and review the answers.

1.1: Before the Internet

In the earliest days of computing circa the late 1960s, the majority of companies purchased only a single large system to handle all of their data processing needs. The systems were proprietary and closed, using hardware and software architectures that were compatible only with the same manufacturers's equipment. The basic components were large central mainframes that connected to intelligent communications "controllers," into which were plugged "dumb" terminals and printers. Network communication consisted entirely of the data sent between a terminal or printer and the mainframe. The terminals were incapable of local storage or configuration, and all intelligence in the entire system resided on the mainframe.

If companies wanted to expand their operations, they were locked into single vendors such as IBM or Digital Equipment Corporation. This led to serious compatibility issues when different organizations within a company or different companies needed to communicate with each other because cross-platform communication did not exist. There was no easy solution to this problem, and the dominant vendors had no incentive to ensure that their systems were compatible with those of other manufacturers.

The U.S. military found itself in an untenable situation when it realized that its different computer systems around the country could not communicate with each other because of proprietary systems and protocols. This meant that different sites could not share data or resources, and in the event of a disaster or systems failure, large amounts of information would be unavailable. *It was this realization of the need for systems that could share information and back each other up that drove the creation of the original "Internet."* In those early days, it was known as the *ARPANET.*

ARPANET: Genesis of the Internet

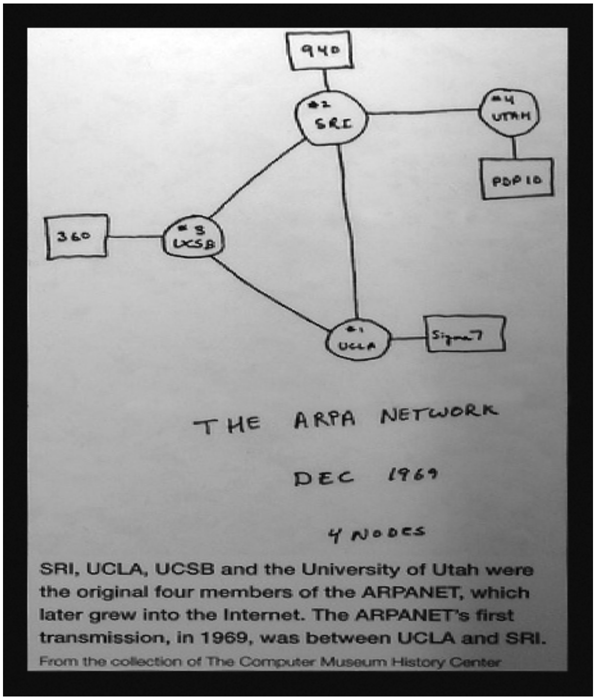
The ARPANET was conceived by the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense (DoD) to be the first cross-organizational communications network. It became the world's first packet-switched network, eventually leading to today's modern Internet. Its beginnings were humble, consisting of only four sites at Stanford, UC Santa Barbara, UCLA, and the University of Utah.

Owing to its military origins, in addition to information-sharing capabilities, the ARPANET was also designed with redundancy in mind. This was the Cold War era, and any communications system had to be able to survive a Soviet nuclear strike on any single or even multiple locations without complete failure. With this in mind, the system was designed with redundant packet switches, links, and a protocol to move data that could dynamically route around failed links and locations. Figure 1.1 shows an early drawing of the proposed network.

Connecting physical components and physical links was merely the first step in the development of the ARPANET, however. In order to make the system truly useful, it would have to support the ability for disparate devices to communicate with each other in a reliable fashion. These systems might be from a variety of manufacturers, and they might be connected to networks in various ways such as by radio or satellite.

As an example, in 1969 the Advanced Research Projects Agency (ARPA) had funded an experimental packet radio network under the direction of Professor Norman Abramson at the University of Hawaii called, appropriately enough, *ALOHANET*. The network connected sites spread around the Hawaiian Islands to a central time-sharing computer on the University of Hawaii campus. ALOHANET users could connect to the ARPANET, but the ALOHANET was not part of the ARPANET core, so from ARPANET's perspective, it was just a terminal connection.

Figure 1.1 The original ARPA network had only four nodes.



SRI, UCLA, UCSB and the University of Utah were the original four members of the ARPANET, which later grew into the Internet. The ARPANET's first transmission, in 1969, was between UCLA and SRI.

From the collection of The Computer Museum History Center

Other developments began to transpire to drive the ARPANET's need for heterogeneous communications. Robert Kahn, a Bolt, Baranek, and Newman researcher who had been instrumental in designing the ARPANET and improving its reliability, had been organizing an event to demonstrate the ARPANET. During this event in the spring of 1973, a new working group called the International Network Working Group (INWG) was organized.

One of the tasks that the INWG decided to undertake was to connect ARPANET and ALOHANET to some of the new packet-switching European networks to create a giant global network. Robert Kahn began a lengthy series of discussions with Vint Cerf, the INWG chairman, to find a solution to their mutual challenges.

Their model was an internetworking of ARPANET with ALOHANET and a satellite network (SATNET)—each of which used different communication protocols and different physical interfaces, optimized for that particular network's needs. Although the model was still in its infancy, the ARPANET designers were beginning to encounter various types of networks that needed to connect to their systems, and they faced a variety of challenges. The challenges faced by Kahn and Cerf would sow the intellectual seeds for the development of a protocol that could provide intercommunication across a wide variety of systems and physical infrastructures. These seeds would later bear fruit with the development of TCP/IP.

ARPANET Challenges and the Origin of TCP/IP

One of the biggest initial challenges to the ARPANET was to guarantee a high degree of reliability across a variety of communication media. Recall that the ARPANET was designed principally to support the U.S. DoD's military requirements, which meant that the network had to be very reliable under failure situations. The ARPANET had originally been designed to use the Network Control Protocol (NCP) for communication between end systems.

NCP provided connections and flow control between different processes running on different computers on the ARPANET. Applications such as email and file transfer were built to use NCP to send the required information and receive responses from other systems. While NCP provided many necessary features and was a good first step, it was not resilient enough to handle unreliable links such as packet radio and satellite links. (Think of the static that is sometimes encountered when listening to your favorite radio station, and imagine data packets encountering similar interference.) This posed a serious problem for interconnecting systems that relied on those types of technologies to the ARPANET.

NCP addressing proved to be problematic as well, since it only addressed next-hop nodes. This would be equivalent to being able to telephone only people in your own area code, or only address letters to people who lived in your same city. While still useful, such limitations would obviously prevent you from communicating outside of your immediate part of the world, and thus NCP addressing was hardly sufficient for the sort of global interconnections that ARPANET designers were contemplating.

If this weren't enough, each network that connected to the ARPANET had its own maximum packet size. In order to facilitate network communication, information is sent over the physical medium in discrete units called *packets*. A *packet* is equivalent to an envelope that holds a certain amount of information and no more. If you need to send more information than will fit in one envelope, you use multiple envelopes. On the ARPANET, various networks supported various maximum-sized packets (*envelopes*), so when a system needed to transfer information from one system to another, it often required unpacking one large envelope to fit into many smaller envelopes.

In order to alleviate these problems, Kahn undertook the development of a new host-to-host protocol. The new protocol would support global addressing, the ability to recover from lost packets, fragmentation and reassembly (the big-envelope-to-small-envelope problem), end-to-end checksums to verify that packet contents have not been altered in transit, and host-to-host flow control. He asked Cerf, who was by this time a professor at Stanford University, to help with the protocol development because he had experience with the design of NCP. To solicit the widest possible input for the project, Cerf ran a series of seminars at Stanford for students and visitors to discuss and challenge ideas as they were formed.

The outcome of this effort was a protocol whose success exceeded anything that its designers could possibly have envisioned. Cerf and Kahn presented their first version of the new protocol at a meeting of the INWG at Sussex University in the United Kingdom in September 1973. They called it the *Transmission Control Protocol* (TCP). And the rest, as the saying goes, is history.

A point on terminology is worth mentioning here. The original TCP included the addressing and other functions of the IP protocol; hence, the original protocol was known simply as *TCP* (the IP protocol is an important part of the TCP/IP protocol stack and will be discussed in detail in Chapter 5). After more work and discussions on the protocol, Kahn and Cerf decided in 1978 to split TCP into two discrete protocols, one called *TCP* and one called the *Internet Protocol* (IP). Each protocol would have separate functions. This new family of protocols became known as *TCP/IP*, and this is how we refer to it for the remainder of this chapter.

From War Room to Boardroom: The Internet Comes of Age

In 1980, the U.S. military adopted TCP/IP as a networking standard, and a "flag day" transition from NCP to TCP/IP was scheduled for ARPANET on January 1, 1983. The transition went reasonably smoothly, and this event marked the beginning of the *Internet* and the beginning of the end for the ARPANET.

Over the years, the ARPANET had become heavily utilized and burdened with congestion, and by 1985, it was reaching the end of its usefulness. In response, the National Science Foundation (NSF) initiated phase 1 development of the NSFNET. The NSFNET was created from a series of regional networks and peer networks. For example, the NASA Science Network was part of the original NSFNET. All of these networks were connected to a major backbone network to form the core NSFNET.

The NSFNET in its inception created a hierarchical network architecture and was more distributed than the ARPANET. The bottom tier consisted of university campuses and research institutions. These were connected to the middle tier (the regional networks). The regional networks were then connected to the main backbone network (the highest tier), consisting of six nationally funded supercomputers.

For many years, the NSFNET was reserved for research and educational purposes. Government agency networks were reserved for government-oriented missions exclusively. In fact, this policy continued into the early 1990s. However, as more peer networks began to be connected and new and different types of communications evolved, additional pressures mounted on the NSFNET administrators to provide additional connectivity and features.

As the NSFNET grew, there began to be a lot of commercial and general purpose interest in obtaining network access and interconnectivity. This, in turn, gave rise to an entire industry of network service providers (also known as *Internet service providers*, or ISPs) willing to fulfill this need for network connectivity. This growth in network connections began to occur on an international scale as networks outside the United States developed their own internetwork connections. These new and existing entities began to interconnect their networks in various ways, increasing the complexity of the infrastructure. Although the NSFNET clearly did not have the size and scope of the modern Internet, even at the early stages, the foundation was being laid for the evolution to the Internet as we know it today.

This growth was, in fact, anticipated by the founders of the INWG. The INWG actively encouraged the development of Internet and TCP/IP-related protocols with an eye toward the growth of internetworking. From the very beginning, anyone was allowed to participate in the development process merely by generating ideas for protocols to use on these emerging networks. These original documents were and still are known as *Requests For Comments* (RFCs). While today's RFCs are more formal and build on a rich and storied tradition of previous RFCs, they are still one of the major driving forces for innovation of new protocols and features.

The INWG evolved over the years into the Internet Engineering Task Force (IETF), which is now the standards body for TCP/IP and related protocols. Despite its importance, the IETF has never had an official charter. It still operates as an open organization where anyone representing research/commercial interests can contribute and improve the existing Internet protocols. IETF working groups enable individual contributors to meet and present and review their work with everyone else via the RFC process.

1.2: Service Providers and Content Providers

Anyone who can offer Internet connectivity could claim to be a service or Internet provider. The term *service provider* covers everything from a provider with a multimillion-dollar backbone and infrastructure providing Internet access to Fortune 100 companies, to a provider with a single router and access server in his garage providing dial-up Internet service to family and friends. The primary function of a service provider is to provide a simple connection to the Internet and possibly some very basic services such as email. Traditionally, a service provider did not go beyond this to give the customer additional application content.

In contrast, a *content provider* provides only information that is requested by the home user or small corporation. This information is typically resident on data servers. The access to these data servers occurs via application protocols (which will be discussed later). The most common example of an application protocol is the HTTP (hypertext transfer) protocol. (The group of servers that provide data via the HTTP protocol is often referred to collectively as the *World Wide Web*, or WWW). By using the HTTP protocol, users can access information from any server that "hosts" the particular information (the *website*) that is being sought by the user. For instance, using the HTTP protocol, the user can simply type **www.google.com** into a web browser such as Internet Explorer and obtain information from the website/data server that hosts `www.google.com`.

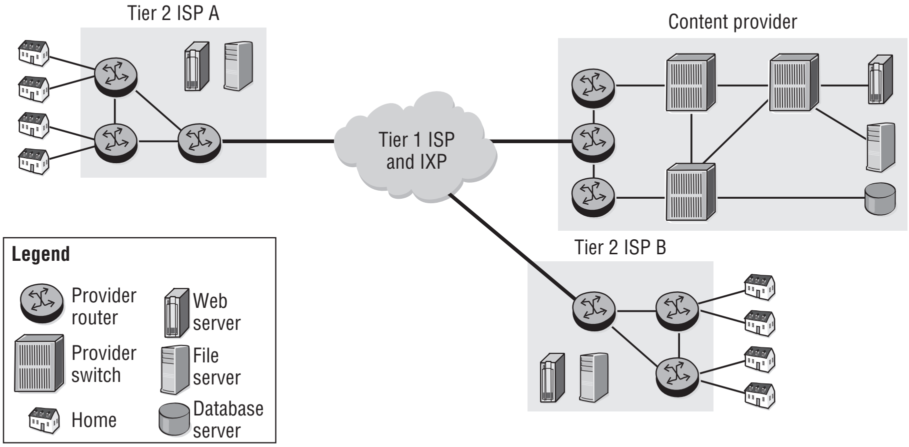
If the user happens to be in Ottawa, Canada and obtains services from ISP A, and the data server hosting the Google website happens to be in California, USA and is locally accessed by ISP B, then ISP A and ISP B will *peer* ("connect") with each other. *Peering* refers to a mutual agreement between two Internet service providers (ISPs) or, more generally, autonomous systems, to enable the exchange of information between each other's customers by direct or indirect interconnections. The indirect interconnection is via the *Internet Exchange Point* (IXP). Apart from Web access, which is the predominant Internet service, ISPs can also provide email access with multiple email accounts, data storage and, very recently, broadcast TV services.

Service providers can be broadly classified into three types or tiers, based on their size and functions:

- **Tier 1**—Tier 1 service providers serve mostly as transit providers. Because of their superior capital and resources, they are able to connect directly to any other major network and do not need to connect to a transit network to obtain service. By definition, a Tier 1 network does not purchase information transit from any other network to reach any other portion of the Internet. Therefore, in order to be a Tier 1 provider, a network must peer with every other Tier 1 network. A new network cannot become a Tier 1 without the implicit approval of every other Tier 1 network, since any one network's refusal to peer with it will prevent the new network from being considered a Tier 1 network. Examples of Tier 1 providers include AT&T, Global Crossing, and NTT Communications.
- **Tier 2**—Tier 2 service providers provide transit for some networks and also request transit service from Tier 1 providers to connect to other parts of the Internet. Examples of these types of providers are Bell Canada and British Telecom.
- **Tier 3**—Tier 3 service providers are smaller still than Tier 2 providers and require Tier 2 or Tier 1 providers for transiting to parts of the Internet. The Tier 3 service providers can provide reselling services for various Tier 2 providers to their customers. Examples of these types of providers would be most small providers that service only a single city or small regional network.

IXPs allow various Tier 1, 2, and 3 providers to exchange Internet data. The IXPs enable information exchange at local points, which avoids having to traverse or backhaul traffic through major points in order to reach the Internet. You can think of an IXP as serving the same purpose as a centralized train station or airline hub. Packets arrive from various providers and through the magic of Internet routing protocols are shunted off to the next provider's network for delivery to their ultimate destination. For example, in Figure 1.2, Tier 2 ISP A and Tier 2 ISP B must connect to each other through an IXP or through a Tier 1 ISP that has agreed to forward their traffic. An IXP would also connect multiple Tier 1 providers to each other in a similar fashion.

Figure 1.2 ISPs exchange data through an IXP.



Enterprises can connect among their regional offices via Tier 2 or Tier 1 ISPs. An office in one region can connect to a Tier 2 ISP, while an office in another region may connect to a different Tier 2 ISP, but they will require a Tier 1 provider to link the two, as shown in Figure 1.3. If possible, connecting to a single service provider will usually provide increased service levels because all of the network transit points are under the control of a single provider. Some providers will even offer service guarantees for customer traffic as long as it does not leave their network, which can be very important for customers deploying services that are sensitive to delay, such as Voice over IP (VoIP).

For example, Figure 1.4 shows how a company may have its offices split into multiple regions around a country. It might be the case that Region 1 and Region 2 have a lot of interoffice application sharing, and thus connecting them to the same Tier 2 would provide adequate performance for less cost than connecting them to a Tier 1 provider. Similarly, Region 3 and Region 4 might have the same type of interoffice application-sharing requirement. In the event that Regions 1 and 2 and Regions 3 and 4 all need to communicate with each other, their respective providers could fulfill this need by using a Tier 1 provider network or IXP as a transit network.

In the next section, we take a closer look at service providers and their functions.

Figure 1.3 An IXP and a variety of tier providers connect regional offices.

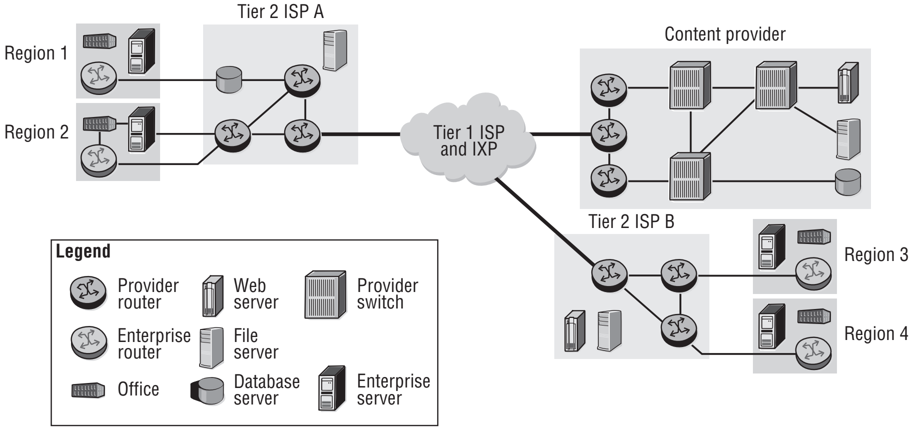
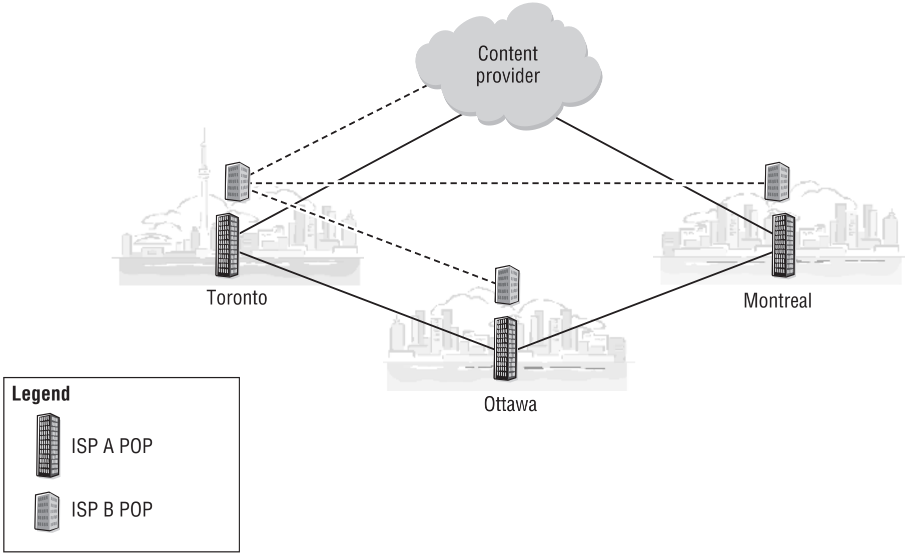


Figure 1.4 A single content provider serves content to multiple locations.



1.3: Modern Internet Service Providers

When the Internet was in its early commercial days, traditional ISPs provided access to basic services such as email and Web surfing via a modem dial-up. The connections were low speed with a theoretical limit of 56 Kbps at the peak of modem technology. Achieving the peak rate was rare, with actual speeds in the 28-Kbps to 36-Kbps range. The nature of the low-speed connections severely limited the range of services that ISPs could offer. Typical services consisted entirely of simple text-based web pages, email, and chatrooms. Additionally, the modem line often had to be shared with the home voice phone, so connectivity was limited to only those times when a connection was required. This paradigm changed dramatically as consumers began to have access to high-speed services such as DSL, and ISPs evolved to provide many additional services to customers.

Modern ISPs can be content providers or peer with several other content providers to provide their users with a myriad of services mainly categorized by voice, video, and data applications. The newer ISPs now compete with the traditional cable, satellite, and telecom providers. The bundling of these three major services that were offered as individual services in the past is referred to as *Triple Play*. In contrast, some of the cable and satellite providers are offering Internet services such as voice and data and also are able to peer with other ISPs and content providers and, in turn, compete with the telecom providers and other ISPs. Deregulation has also allowed companies who traditionally were able to offer only single services such as voice or cable, to offer new services in a wide-open, competitive services marketplace.

A major motivation behind the bundling of traditionally offered individual services is cost reduction. Another motivation is to offer customized services with varying price points. For example, ISP A may offer its end-users three packages: a basic service, a premium service, and an elite service. Each package is incrementally priced and consists of higher service utilization. The basic package could offer a 10 Megabits per second (Mbps) combined voice, Internet, and basic video service, whereas the premium and elite packages could be a 20 or 40 Mbps voice, very-high-speed Internet, and high-definition video services.

Apart from residential customer traffic, ISPs typically provide the business traffic needs for an enterprise whose traffic requirements with respect to bandwidth and timely delivery of an enterprise are well beyond that of the typical home user. A medium to large company may require the ISP's geographical presence to connect to its offices or other enterprise organizations. Additionally, the enterprises may require various types of services from an ISP such as web hosting or Layer 2/Layer 3 VPN (L2/L3 VPN) services for intersite connectivity.

This enterprise traffic via the ISP network is critical to the daily operations of the enterprise, and the delivery of this type of traffic is usually guaranteed by the ISP networks using *service level agreements* (SLAs). *Service level agreements* are contractual agreements between an ISP and its customers that define traffic guarantees and penalties resulting in payouts to the customers if the stated service level cannot be met for any reason. In order to provide these stated levels of service, the providers will limit the scope of the devices they will support. Normally, the provider will guarantee service levels up to a demarcation point.

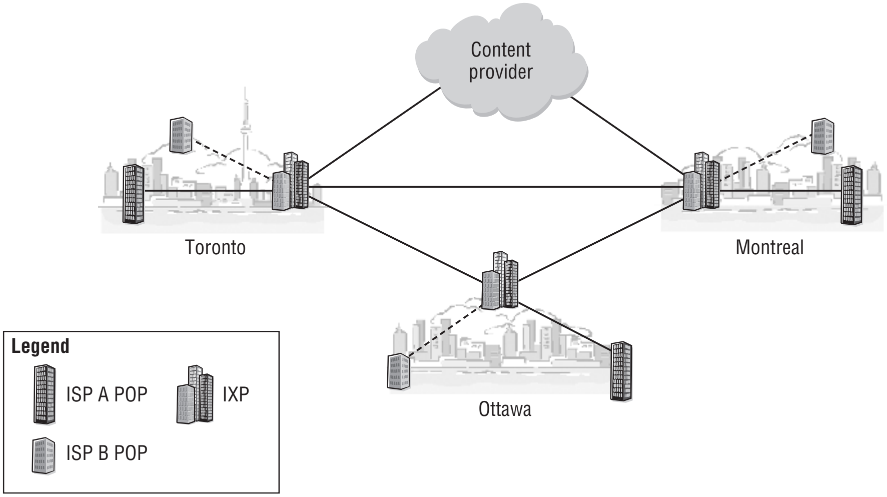
Demarcation points serve as a means for the ISP to support the customer or other ISPs at a particular point characterized by equipment managed by the customer or the ISP. *Demarcation points* are essentially points at which delivery of packets becomes the responsibility of the customer or the provider, depending on the direction of packet delivery. A typical demarcation point for a home user would be the customer's DSL modem. The provider will test their signal up to the DSL modem and ensure that their signal has good quality up to that point. Beyond the DSL modem, it is the home user's responsibility to ensure that they have the correct connections from their computer to the modem.

All of these additions in terms of services and competing providers have led to exceptionally large growth of the Internet. Today, the Internet backbone has grown staggeringly complex compared to its humble beginning as the ARPANET. It is a collection of service providers that have connection points called *Points of Presence* (POPs) over multiple regions. The collection of POPs and the interconnections between them form the *provider networks*. Customers who require Internet service from a service provider are connected via access or hosting facilities in that provider's POP. The service providers have direct or indirect access to the data servers. The customers can be the "end-hosts" who receive the Internet service from their respective service providers.

One of the factors that make the backbone complex is that the customer of a service provider may also be a service provider for its own customers, and may connect to other service providers for certain data servers. What this means in practical terms is that a provider may or may not house the content that its customers use. The content can just as easily be stored on a different provider's network or on multiple provider networks as it can be on the customer's provider network. The peering and network connections of the providers make the actual location and delivery of the requested information transparent and seamless. Figure 1.4 illustrates how a single content provider can serve its content to multiple ISP POP locations.

As previously discussed, interconnections between providers are facilitated through exchange points known as IXPs. Because IXPs serve as the switching points for a large amount of traffic, it is critical that they switch packets from one provider network to another as quickly and reliably as possible. Having an IXP at the city level helps all the traffic between various ISPs and content providers to travel within the same city. In Figure 1.5, for instance, the ISP A POP and ISP B POP in Toronto can communicate via the Toronto IXP. If a content provider wants to peer with the IXP in a city, all the traffic between the ISP POPs in that city and the content provider is now localized. Without the presence of an IXP, the intracity inter-ISP traffic might have to be carried to an IXP in another city. This means that a user could be sitting at her apartment in Toronto and accessing a server at her office in downtown Montreal, but her traffic might be routed via Ottawa because of an out-of-town IXP.

Figure 1.5 Content goes through an IXP and is forwarded to various ISP POPs.

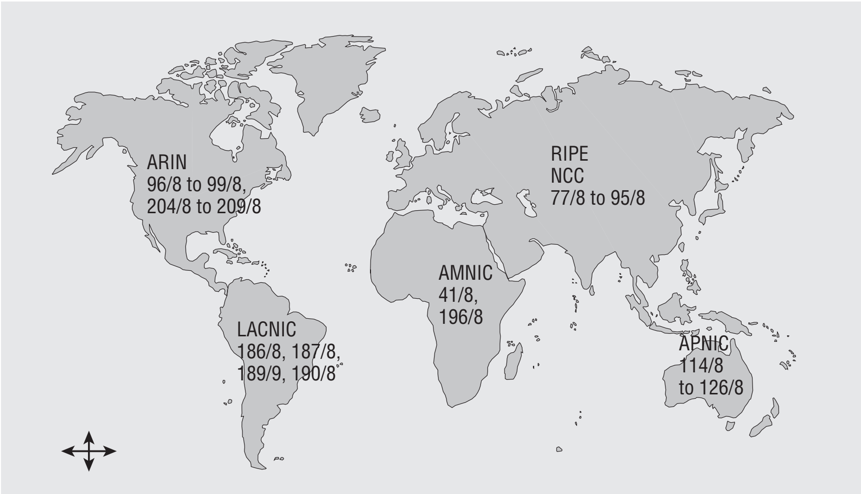


1.4: Overview of TCP/IP

For the Internet to work properly, the underlying components need a common way of communicating. This is achieved by providing common addressing to all the physical components. You can think of a physical device address the same way you think of a home address. Each address uniquely identifies a particular house somewhere in the world, and no two addresses are exactly alike. Similarly, the addressing is hierarchical, having a street address, a city, and a country (and a state, if you are in the United States). However, since these addresses are meant to be used solely by machines, they use only numeric addresses. In order to get data from one addressed host to another, you need a protocol that understands the addressing and knows how to get from point A to point B. On the Internet, this protocol is known appropriately enough as the *Internet Protocol* (IP), and the addresses are known as *IP addresses*.

An example of an IP address is *138.120.105.45*, and this address must be unique to a single computer. Delivering packets to a given IP address is again very similar to the methods that one would use to deliver mail to a person's home. The distribution of IP addresses is supervised by a centralized authority known as the *Internet Assigned Numbers Authority* (IANA), the way home addresses are issued by local government agencies. Indeed, the actual issuance of IP addresses is handled by different delegated Regional Internet Registry (RIR) agencies in different parts of the world, as shown in Figure 1.6.

Figure 1.6 Regional Internet Registry agents allocate IP addresses.



Some of the IP address allocations managed by RIR (*Regional Internet Registry*)

Just as with the postal system, the Internet provides a method for sending information from one place to another. On the Internet, information is sent in discrete units called *packets*. These *packets* actually consist of multiple pieces of information that are layered one on top of the other. Each piece of information is relevant to a particular process used by end-user computers or intermediate network devices. To continue the analogy, the layering of information can be compared to regular postal service, where there are several distinct functions:

- Creating the letter
- Enclosing the letter in an envelope, writing the sender's and recipient's addresses
- Choosing the type of delivery for the envelope (same day service, same week, etc.)
- Placing the appropriate stamp on the letter reflecting the service
- Physically sending the letter via carriers through air, water, or land

All these functions are relevant with transporting the letter to the proper destination. At the destination, the letter is received, and depending on the transport service, an acknowledgment of receipt may be sent back to the sender. The letter is simply then removed from the envelope, and its contents are then read.

Layering of TCP/IP information is treated in a similar fashion. The central difference is that TCP/IP is intended to transfer information to individual listening processes at a given IP address. A computer with a single IP address might provide numerous services such as email, web hosting, and data storage. TCP/IP allows for all of the IP packets to be delivered to the same computer system, and then unpacked and delivered to the individual process that requires that particular piece of information. TCP/IP also provides a reliable service, meaning it will re-send packets if it does not receive an acknowledgment of receipt. This layering approach, wherein service functions are distributed, is common of network protocols in general and TCP/IP in particular.

Each layer of the protocol layering stack adds the pertinent information (destination, error checks, etc.) at the beginning of the data, thereby adding more information to the data. The data en route to the receiver passes through several other systems that look only at the relevant header information for the layers that they are interested in and pass the data to another device. This would be similar in function to having envelopes inside other envelopes, with each envelope having different information.

For example, there might be an envelope that has just address information for an office building. Once the envelope is delivered to the building, the mail room would open the outer envelope, and inside there would be another envelope with a specific office location. None of the intermediate mail delivery systems would need to know about the exact office, so that information would be shielded from them.

The purpose of a network protocol suite is to define the protocols and technologies that support the interconnection of a diverse array of hardware and systems for deployment of a wide range of applications over the network. Anyone who has used an Internet application such as a web browser or email can appreciate the complexity of the systems required to support these applications. It is only due to our familiarity with these tools that we do not marvel that a user in Toronto, Canada can send an email quickly and effortlessly to a colleague in Pune, India whom he has never met face-to-face.

The layering of protocols provides a way to simplify this complex problem by segregating it into a number of smaller functions. Each layer performs a specific function that contributes to the overall functioning of the network. Systems participating in the protocols may not need to participate at every level, making it easier to move data from one point on the Internet to another with minimal effort. For example, network devices on the Internet may know only the destination address of information and need not know that the information in the packets is email or Web traffic, just as the intermediate mail sites need not know the exact office destination of the inside letter in our earlier example.

Understanding the TCP/IP Layers

The *TCP/IP protocol suite* (or "Internet protocol suite") is constructed around four layers of technology, as illustrated in Figure 1.7. The *application layer* provides all the services (e.g., web browsing and email) available to users of the Internet. The *network interfaces layer* includes all the hardware that comprises the physical infrastructure of the Internet. The two intermediate layers, the *transport layer* and *Internet Protocol layer*, provide a common set of services that are available to all Internet applications and that operate on all the hardware infrastructure of the Internet.

The Application Services Layer

The *application services layer* is the layer for the user. This layer only describes network applications. (Applications such as word processors and database programs are not considered network applications as they do not require network connectivity and are therefore not part of this layer.) Figure 1.8 lists some examples of network applications in the "Application services" box. Without network connectivity, these applications would be essentially useless.

Figure 1.7 The Internet Protocol suite is constructed around four layers.

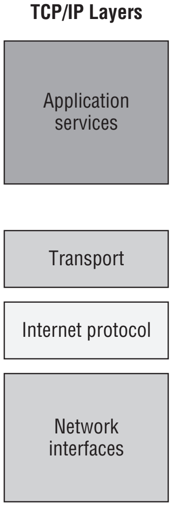
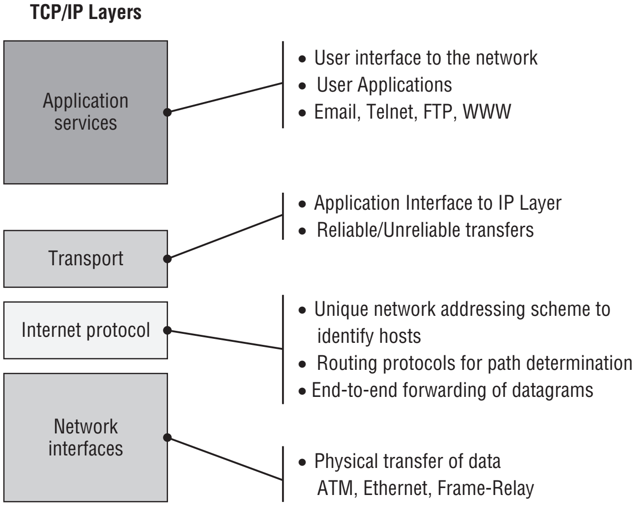


Figure 1.8 Example applications for the TCP/IP layers.


TCP/IP Layers
Application services
Transport
Internet protocol
Network interfaces
• User interface to the network
• User Applications
• Email, Telnet, FTP, WWW
• Application Interface to IP Layer
• Reliable/Unreliable transfers
• Unique network addressing scheme to identify hosts
• Routing protocols for path determination
• End-to-end forwarding of datagrams
• Physical transfer of data ATM, Ethernet, Frame-Relay

The Transport Layer

The *transport layer* is the application's interface to the network. The transport protocol provides a mechanism for an application to communicate with an application residing on another device in the network. In the TCP/IP protocol suite, there are two transport protocols: the *Transmission Control Protocol* (TCP), and the *User Datagram Protocol* (UDP). *TCP* is a connection-oriented protocol that provides an ordered and reliable transfer of data over the network. *UDP* is a connectionless protocol that supports the transfer of a single datagram across the network with no delivery guarantee. UDP is simpler and operates with less overhead than TCP. However, most Internet applications use TCP for data transfer because it provides a reliable transfer service. This includes HTTP (web browsing), email, Telnet, and FTP. Some applications, such as the Dynamic Host Configuration Protocol (DHCP) and the Trivial File Transfer Protocol (TFTP), use UDP because they only require a simple datagram transfer.

The Internet Protocol Layer

The *Internet Protocol layer* provides a common addressing plan for all hosts on the Internet as well as a simple, unreliable datagram transfer service between these hosts. IP is the common glue that connects the Internet. IP also defines the way a datagram (or packet) is routed to its final destination. In an IP network, the forwarding of packets across the network is handled by routers. IP routers examine the destination address of a datagram and determine which router would be the next hop to provide the best route to the destination. The router forwards the packet to the next hop router, where the process is repeated until the datagram reaches its destination. This is known as *hop-by-hop routing.*

Routers communicate with each other using dynamic routing protocols to exchange information about the networks they are connected to. This allows routers to make forwarding decisions for the datagrams they receive. In later chapters, you will learn about some key IP routing protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). These topics are covered in much greater detail in later chapters, so for now, all that you need to understand is that routing protocols allow for the delivery of packets from one "hop" or router to the next in a predictable manner.

The Network Interfaces Layer

The *network interfaces layer* comprises the hardware that supports the physical interconnection of all network devices. The technologies of this layer are often designed in multiple layers themselves. The common attribute of all technologies of this layer is that they are able to forward IP datagrams or packets. There are many different technologies that operate at this layer, some of which are very complex. Some of the protocols commonly used at this layer include Asynchronous Transfer Mode (ATM), Frame-Relay, Point to Point Protocol (PPP), and Ethernet. However, there are many other protocols used; some are open standards, and some are proprietary.

The diversity of the network interfaces layer demonstrates one of the benefits of protocol layering. As new transmission technologies are developed, it is not necessary to make changes to the upper layers to incorporate these technologies into the network. The only requirement is that the new technology be able to support the forwarding of IP datagrams. For example, the original Ethernet standard only supported speeds up to 10 Mbps. Later came FastEthernet, which supported speeds up to 100 Mbps, and later still came GigabitEthernet, with speeds up to 1 Gbps. It is now not uncommon to encounter 10 Gbps Ethernet. Owing to the layering nature of TCP/IP, none of these changes at the lower layers required any changes to the upper protocols.

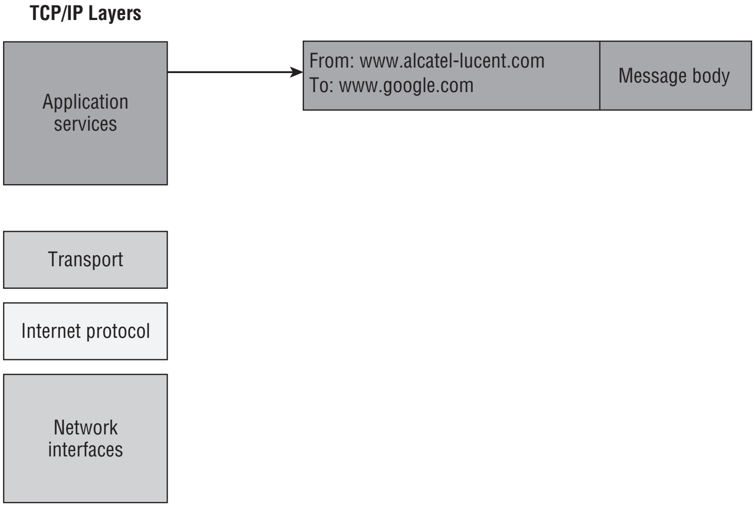
Forwarding Data

When a network application wants to communicate with another application across the network, it must first prepare the data in the specific format defined by the protocol to be used by the receiving application. A specific protocol is used so that the receiving application will know how to interpret the data it receives.

For example, in the case of a world wide web (WWW) request, the message consists of two parts, the message header and the body, as shown in Figure 1.9. The message header contains the sender's and receiver's addresses, as well as other information such as the urgency of the message and the nature of the message body. The format of the header and the nature of the addressing are defined by the application protocol. In the case of a WWW request the protocol is the hypertext transfer protocol (HTTP).

In addition to defining the format of the message, the protocol also specifies how the applications are expected to interact with each other, including the exchange of commands and the expected responses.

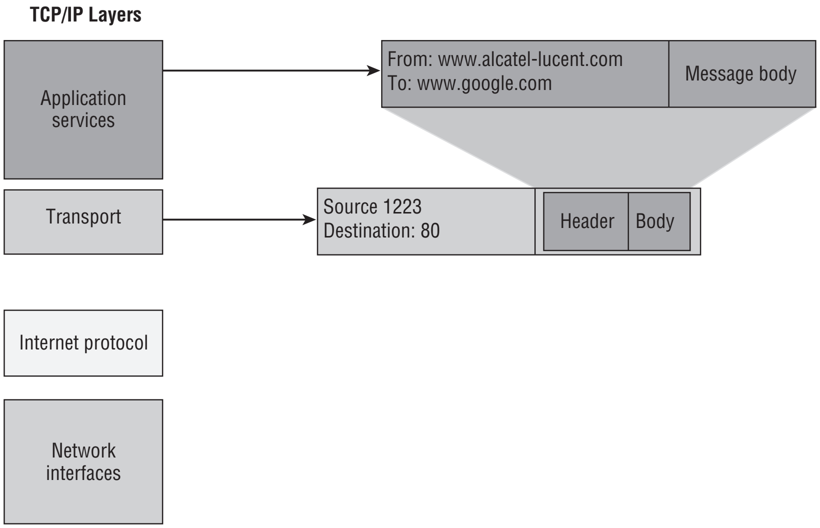
Figure 1.9 A WWW application creates a message that includes the sender and recipient information in the message header and the contents of the message in the message body.



TCP treats all application data as a simple byte stream, including both the message header and the message body. TCP accepts the application's data and breaks it into segments for transmission across the network as required. To accomplish this reliable transfer, TCP packages the application data with a TCP header. On the receiving end of the connection, the TCP protocol removes the TCP header and reconstructs the application data stream exactly as it was received from the application on the sender's side of the network. In other words, TCP simply takes the data given to it by the upper layer application, and passes it to the upper layer application at the other end without trying to interpret the contents. Any protocol-specific formatting of the data is up to the upper layers.

As shown in Figure 1.10, the TCP and UDP headers carry source and destination addresses that identify the sending and recipient applications because a single host system may support multiple applications, as mentioned previously. These addresses are known as *port numbers*. Some port numbers are considered "well known" and should become familiar to you, such as port 80 for HTTP, port 21 for FTP, port 25 for Simple Mail Transfer Protocol (SMTP) and port 23 for Telnet. To transmit its segments of data across the network, TCP uses the services of the IP layer.

Figure 1.10 The transport layer specifies the TCP source and destination port numbers that will be used by the upper layer application.

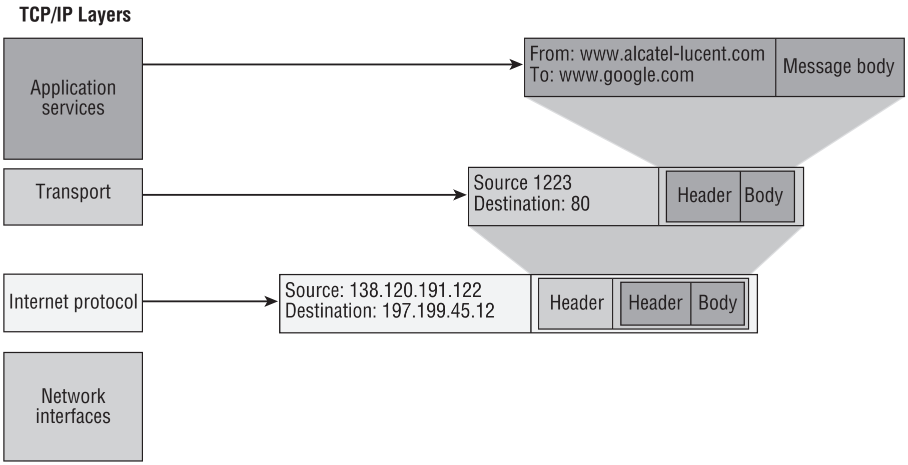


The IP layer provides a common addressing scheme across the network as well as a simple, unreliable datagram forwarding service between nodes in the network. Data from the transport layer is packaged in IP datagrams for transfer over the network. Each datagram travels independently across the network. The intermediate routers forward the datagram on a hop-by-hop basis based on the destination address. This allows for the network to dynamically route around any problems or failures in the network and deliver the packets as efficiently as possible. The packets may arrive out of order, so it is up to the upper layer TCP protocol to reassemble the message correctly.

As indicated in Figure 1.11, each datagram contains source and destination addresses that identify the end nodes in the network, and as you have seen, every node in an IP network is expected to have a unique IP address. IP uses the services of the underlying network interfaces such as Ethernet or ATM to accomplish the physical transfer of data.

The *data link layer* is the term used to describe the network interfaces used by IP for physically transmitting the data across the network. The units of data transmitted at the data link layer are usually known as *frames*. IP datagrams must always be encapsulated in some type of Data Link frame for transmission.

Figure 1.11 The network layer adds source and destination IP addresses so that the packet can be forwarded through the network.



A typical Data Link frame contains a header, usually containing some type of address. The frame also often carries a trailer that contains some type of checksum to verify the integrity of the transmitted data. There are many types of technology used as network interfaces by IP, and they each have their own specific format and rules of operation. As noted earlier, the common characteristic is that the technologies are all capable of carrying IP datagrams.

The addressing at this layer identifies the two endpoints of a data exchange to the data link protocol. For example, Figure 1.12 shows the addressing of an Ethernet frame. Some Point-to-Point Protocols such as PPP may not use addresses if there is only one possible destination for the data. After all, you don't need addressing when two network devices can only send information to each other.

If it was not obvious from the preceding discussion, routers provide the critical traffic control of the IP datagrams across the Internet. Once the end-user's computer creates the packet and places it on the wire, its job is done until it receives a response. From the user's perspective, the information simply goes into a *black box* and gets sent to its destination, and a response is received. Behind the scenes, numerous, sometimes dozens, of routers reliably perform their duties accepting packets, examining the destination, looking up the next hop router for delivery, and forwarding the packet on its merry way. Along the way, the packet may cross Ethernet, ATM, Frame-Relay, PPP, and carrier pigeon to reach its intended destination (OK, maybe not really carrier pigeon, but you get the idea). Figure 1.13 illustrates this concept.

Figure 1.12 The data link layer adds source and destination MAC addresses for forwarding on the local network segments.

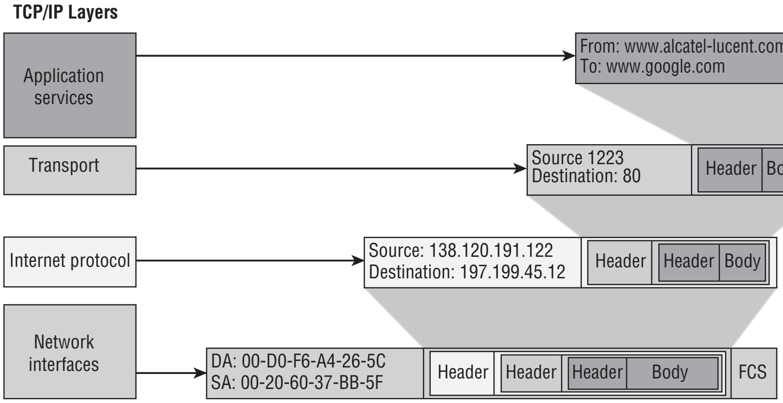
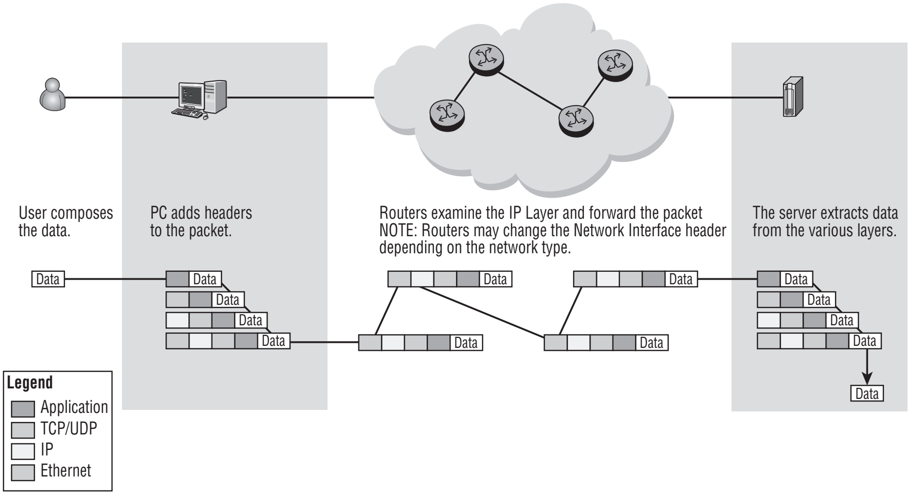


Figure 1.13 Data from applications is sent to the TCP/IP protocol stack where all the appropriate headers are added and the packet is sent on to the network for forwarding to its destination. As the packet travels through the network, the Layer 2 information is changed at each router, but the network, transport, and application information remain unchanged.



User composes the data.
PC adds headers to the packet.
Routers examine the IP Layer and forward the packet
NOTE: Routers may change the Network Interface header depending on the network type.
The server extracts data from the various layers.
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Legend
Application
TCP/UDP
IP
Ethernet

The process begins when the user composes data and hands it to a process on his or her computer for network processing. Assume that the user has composed an email and pressed the Send key in her email application. Behind the scenes, the email application will take the data the user has written and place an SMTP protocol header in front of it (the SMTP header is marked in darkest gray in Figure 1.13). The mail application will then make a request to a TCP process running on the user's computer to send the SMTP data. TCP will accept the SMTP information, possibly breaking it into discrete units of information, and place a TCP header in front of the SMTP header.

This process continues in like manner with TCP handing its information to the IP process that places an IP header in front of the TCP header (the IP header is marked in lightest gray in Figure 1.13), after which the IP process passes its information to the Ethernet process so that the Ethernet header can be placed in front of the IP header. Now that all of the protocol headers have been inserted one in front of the other, the packet is ready for transmission across the network. In the figure, each router in the path would remove the Layer 2/Ethernet header, read the Layer 3/IP header information in order to know what next hop to forward the packet to, place a new Layer 2 header onto the packet, and forward the packet out of the appropriate interface.

The reason for this is that Layer 2 headers are only used on local network connections such as Ethernet or a PPP link. Once the packet reaches the next router, the Layer 2 information is no longer relevant, and therefore a new Layer 2 header must be created and added to the packet in front of the IP header. The exact process used to determine what information to place in the Layer 2 header is described in later chapters, so all that is necessary for you to understand at this point is that a new Layer 2 header is created by each router along the path as it forwards packets. Another critical point to understand is that the packets are forwarded strictly based on the information in the IP header. None of the routers in the path need to read either the TCP or SMTP headers in order to properly forward the packets.

We should note that in some circumstances a router might examine the TCP information in order to forward packets based on the upper layer protocol. For example, a provider might wish to give HTTP traffic priority over SMTP traffic in the event of network congestion. However, this is a deviation from the standard forwarding process. Typically, a router will forward packets based only on the information in the IP header as described. It is only when the packet reaches the end-host that the TCP and SMTP headers are examined and removed.

At the destination host, the process previously described to create the packet is performed in the reverse direction, with each process removing the appropriate header and forwarding the remaining information, complete with upper layer headers, on to the next process. In our example, an Ethernet process would remove the Ethernet header, and then it would be processed by the IP, TCP, and SMTP processes, respectively.

This is a simplified example for mail. In reality, the mail would be delivered to a mail server, and then the destination user would retrieve his or her email using a different protocol. However, the process performed at the mail server to receive the packets is as described.

That this process works so well and so reliably is a testament to the layered design of TCP/IP that provides for the segregation of responsibility required to allow for such diverse networks to intercommunicate seamlessly.

The OSI Reference Model

Up to this point, we have been discussing the TCP/IP layering model exclusively. However, the TCP/IP model is not the only game in town. The *Open Systems Interconnection* or *OSI Reference Model* represents a logical way of organizing how networks talk to each other so that all hardware and software vendors have an agreed-upon framework to develop networking technologies—similar to TCP/IP. The OSI model was created by the International Organization for Standardization (ISO) with the following goals:

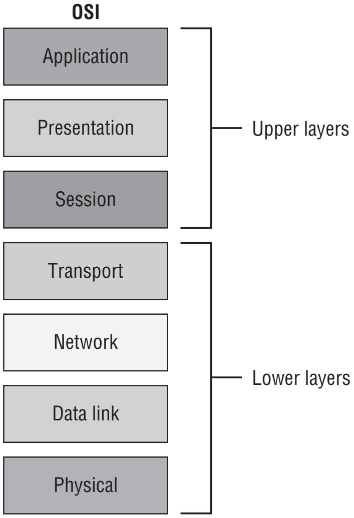
- Simplify complex procedures into an easy-to-understand structure.
- Allow vendors to interoperate.
- Provide the ability to isolate problems from one layer that may be passed to other areas.
- Allow a modular plug-and-play functionality.
- Provide an independent layer design.

The ISO is a network of national standards institutes based in Geneva, Switzerland. Its goal is to help promulgate standards that have been developed by consensus in particular fields of industry. The OSI model is simply one of many computer industry standards developed by the ISO, which, in turn, is simply one field of the many that have ISO-developed standards.

The OSI model is represented by the seven layers depicted in Figure 1.14. These layers may be grouped into two main areas, defined simply as the upper and lower layers.

Although a single device (e.g., a UNIX workstation) can execute all seven layers, this is not practical in real networks. The amount of traffic that needs to be moved through modern networks requires purpose-built devices that handle various layer functions. Two such examples are bridges, which are purpose-built for Layer 2 operation; and routers, which are purpose-built for Layer 3 operations.

Figure 1.14 The OSI reference model defines seven distinct layers.



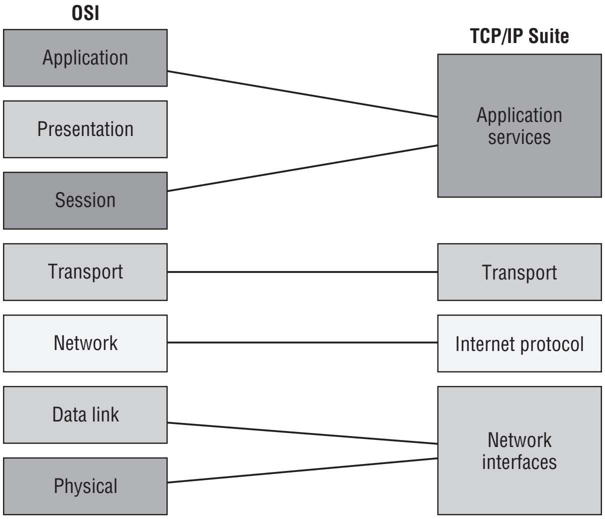
It is important to understand at the outset that the OSI model is simply that—a model. There are various protocols that implement services at each of the described layers of the model, but the model itself is simply a reference point for protocol designers. The OSI reference model was developed at the end of the 1970s, but the development of actual protocols to support the reference model was slow. (Recall that TCP had been developed in 1973 and thus predates the OSI model.) By the early 1990s, several OSI protocols (TP0-4, CLNS, CONS, X.400, and X.500) had been specified and commercial implementations attempted, but the success of TCP/IP and the weaknesses of OSI led to the complete adoption of TCP/IP for internetworking. OSI protocols were never in wide use, and today they are more of interest for historical ideas that didn't really catch on, like the Edsel or New Coke.

However, despite the failure of the actual OSI protocols, the OSI Reference Model terminology lives on and is widely used to describe the layering of network protocols. Indeed, much networking terminology derives from the OSI protocol suite. A very few remnants of OSI are still in use; for example, Lightweight Directory Access Protocol (LDAP), which is a derivation and simplification of X.500, and Intermediate System to Intermediate System (IS-IS), which was designed as an OSI routing protocol. IS-IS was later adapted to TCP/IP networks and is still a key routing protocol in many provider networks, while LDAP is used in some networks to provide user authentication services.

The following list maps the TCP/IP suite layers to the OSI model to see how they fit and where they differ. The TCP/IP suite differs from the OSI model in that the TCP/IP suite uses four protocol layers and the OSI model uses seven layers. Figure 1.15 shows the rough protocol layer relationship between the two models. It is "rough" because some of the functions of the TCP/IP layers bleed into the functions of the other OSI layers. This is not the fault of the TCP designers as after all, they were first to the party and their primary concerns were to design a practical protocol and not a protocol that fits neatly into the OSI model.

- **Network Interfaces**—The network interfaces layer defines the interface between hosts and network devices and contains the functionality of both the physical and data link layers of the OSI model. Protocols such as Ethernet describe both the framing of data (Layer 2) and the physical transmission of the frame over the media (Layer 1). This layer is often referred to as *Layer 2* or *L2* because it provides OSI Layer 2–type services to the IP layer.
- **Internet Protocol**—The IP layer provides a universal and consistent forwarding service across a TCP/IP network. IP provides services comparable to the OSI network layer and is sometimes referred to as a *Layer 3* (or *L3*) protocol. The OSI protocol CLNP corresponds most closely to IP.
- **Transport**—The transport layer comprises two main protocols, TCP and UDP. These transport protocols provide similar services to the OSI transport protocols. TCP is very similar to the OSI transport protocol, TP4. TCP and UDP may be referred to as *Layer 4 protocols*.

Figure 1.15 The TCP/IP layers do not map exactly to the OSI layers; multiple OSI layers are performed by a single TCP/IP layer.



- **Application Services**—The application services provide end-user access to the Internet. Any of the services of the upper three OSI protocols that are required are incorporated into the application protocols. There are several Internet protocols that provide services similar to these OSI layers, although they do not follow the layering or service definitions of OSI. For example, MultIprotocol Mail Extensions (MIME) provides presentation-like services similar to SMTP. Application layer protocols are sometimes referred to as *Layer 7 protocols*.

The most important aspect of the OSI model is the terminology itself and not the particular implementation of protocols, which are, as previously stated, for the most part dead. You will often encounter vendor technologies that perform certain new functions such as "content switching" that purport to perform forwarding operations on Layer 4 or Layer 7. It is a key part of evaluating these claims and technologies that you understand what these layers entail and why having devices that understand these upper layers might be important for particular services.

Chapter Review

Now that you have completed this chapter, you should have a good understanding of the following topics. If you are not familiar with these topics, please go back and review the appropriate sections.

- The development of the ARPANET and its evolution to the modern Internet
- The function of the IETF and its relationship to the Internet
- The problems that the creation of TCP/IP was designed to solve
- The distinction between an Internet provider and a content provider
- The basic components of the Internet needed for it to function
- Protocol layering and why it is used
- The layers of the TCP/IP protocol
- The similarities and differences between the TCP/IP protocol and the OSI model

Post-Assessment

The following questions will test your knowledge and prepare you for the Alcatel-Lucent NRS I Certification Exam. Please review each question carefully and choose the most correct answer. You can compare your response with the answers listed in Appendix A. You can also download all of the CD content at `http://booksupport.wiley.com` to take all the assessment tests and review the answers. Good luck!

1. The original network that ultimately became the Internet was called

   A. NSFNET.

   B. ARPANET.

   C. DoDnet.

   D. DARPA.

2. The primary organization behind the development of the original Internet was

   A. IBM

   B. Digital Equipment Corporation (DEC)

   C. Stanford University

   D. the U.S. Department of Defense

3. Which of the following was *not* a primary design concern during the development of the original Internet?

   A. Reliability

   B. Bandwidth

   C. Interoperability

   D. Support for diverse network media

4. Which of the following was *not* a reason TCP was a superior transport protocol to NCP?

   A. Support for global addressing

   B. Support for end-to-end checksums

   C. Support for applications such as email

   D. Support for fragmentation and reassembly

5. Which of the following OSI layers is *not* paired with the correct implementation?

   A. Layer 7—Email

   B. Layer 3—TCP

   C. Layer 4—UDP

   D. Layer 2—PPP

6. Part of the growth of the ARPANET was driven by the ability of anyone to create and disseminate information about potential protocols and applications in a particular kind of document. These documents are known as

   A. Requests For Information.

   B. Protocol Revisions.

   C. Requests For Comments.

   D. Requests For Configurations.

7. ISPs connect to each other at well-defined network locations to exchange information. These connection points are known as

   A. ISPs.

   B. IXPs.

   C. BGPs.

   D. POPs.

8. A company that has locations throughout the country can obtain service at each location from a Tier 1, Tier 2, or Tier 3 provider. What is one reason a company might choose to connect all locations to a Tier 1 provider despite the higher costs involved?

   A. Sites at different tiers cannot communicate.

   B. Tier 3 providers don't use TCP/IP.

   C. Only Tier 1 providers provide content.

   D. A single provider could offer SLAs to each location.

9. Which of the following services would likely be offered by a content provider but *not* a service provider?

   A. Standard dial-up service

   B. Live video streaming from sports events

   C. Email service

   D. Basic Web Services

10. Which of the following accurately describes the TCP protocol?

    A. Connectionless with no guarantee of delivery

    B. Connectionless with guarantee of delivery

    C. Connection-oriented with guarantee of delivery

    D. None of the above

11. Originally, the IP protocol functions were performed by

    A. Ethernet.

    B. TCP.

    C. NCP.

    D. ALOHANET.

12. When an HTTP packet needs to be forwarded over the Internet, which of the following accurately describes the order of the headers as they would be placed in front of each other in the packet (assume the orginating device is on an Ethernet network)?

    A. HTTP, IP, TCP, Ethernet

    B. HTTP, TCP, IP, Ethernet

    C. HTTP, UDP, IP, Ethernet

    D. HTTP, IP, Ethernet

13. A router processing the packet described in Question 12 would need to examine and/or manipulate the headers for

    A. Ethernet only.

    B. IP only.

    C. TCP and IP only.

    D. IP and Ethernet only.

14. What would a router processing the packet described in Question 12 do with the Layer 2 header of the incoming packet?

- **A.** Remove the source Layer 2 address, add its own, and forward the packet.
- **B.** Remove the Layer 2 addresses and replace them with new addresses.
- **C.** Remove the entire Layer 2 header and create a new one based on the next hop interface.
- **D.** Leave the original Layer 2 header but forward the packet based on the destination address.

15. Most of the OSI-created protocols are no longer in use, although a few still provide some critically important functions. Which of the following describes an OSI protocol that is still in use?

- **A.** OSPF
- **B.** LDP
- **C.** TP0
- **D.** IS-IS