1

An Introduction to TCP/IP

Perhaps the greatest advantage of Linux (and UNIX, in general) is its flexibility, especially with respect to networking. It was designed to make access easy; its use of Transmission Control Protocol/Internet Protocol (TCP/IP) networking is in harmony with that design perspective. In this chapter, we will take a look at networking from an overview perspective and introduce the TCP/IP model. We will also make a number of definitions and introduce a number of concepts that we will use throughout this book.

Computer Networking

Networks connect things together. The collection of roads throughout the United States comprises a network of paths by which vehicles can reach various points. The purpose of a network of roads is to permit people to efficiently move from one place to another. Computer networks consist of interconnected computer systems. The purpose of networking these components together is to share information and computing resources. An obvious example of a computer network is the Internet, in which millions of people access information and computer resources throughout the world. Most of the folks utilizing these resources have little notion of how those resources are made available to them or what tasks are required to provide such resources. That, after all, is the job of some computer jockey–or more formally, a system and/or network administrator.

System and network administrators own the task of ensuring that computer resources remain available. While the tasks of these two distinct roles often overlap, it is the job of the network administrator to ensure that computers and other dedicated network devices, such as repeaters, bridges, routers, and application servers, remain interconnected. In order to accomplish this task, the network administrator needs to be familiar with the software and hardware employed to effectively connect the various components. This familiarity must begin with the concept of network types and models.

Network Types

There are a number of terms used to describe different types of networks. Such terms include local area network (LAN), wide area network (WAN), campus area network (CAN), and metropolitan area network (MAN).

LAN LANs, generally speaking, are collections of interconnected *nodes*, all of which are geographically nearby, such as in the same room or in the same building. A node is any device which can be networked, like a computer (often referred to as a *system*), a printer, an appliance (web server, NFS server, etc.), or a black-box device (usually a router or switch—we define these later in this chapter). It should be noted that with the advent of layer 2 routing (see Chapter 16 for a discussion of layer 2 routing or *switching*; layers are discussed in the next section), the notion that the nodes within a LAN must be geographically close becomes untrue.

WAN A WAN is a collection of interconnected and geographically disparate nodes. Often, the major distinction between a LAN and a WAN is the use of some form of high-speed media to interconnect the nodes. Such media includes microwave, satellite, and telecommunications connections. WAN is a very general term that is often applied to collections of LANs and other WANs as well as collections of nodes. One example is the Internet, which is frequently described as a WAN—in fact, the Internet interconnects both LANs and WANs.

CAN The term CAN is usually applied to LANs or WANs that comprise the collection of interconnected nodes belonging to a single company or university/college but whose interconnection extends across many buildings.

MAN This term is usually applied to the collection of nodes within a metropolitan area that fall under the same corporate control such as that of a telecommunications company or independent service provider (ISP).

If you think that these definitions of LAN, WAN, CAN, and MAN are a bit fuzzy, good! That's because they are! Furthermore, it's likely that they will get fuzzier as time goes on and technology presses forward.

As a consequence of the fuzziness of these terms, we define the term *local network* to mean the collection of all nodes connected via the same medium and sharing the same *network number*. We formally define network number in "IPv4 Addressing" on page 108, but for now think of sharing the same network number as meaning that no hardware or software boundaries must be crossed. In other words, nodes that share the same network number can communicate with each other without requiring the services of a *router* (router is defined in "The Internet Layer" on page 9). The definition of local network given here is synonymous with the definition of *link local scope* (defined in Chapter 5) and is quite similar to that of *broadcast domain,* which is defined in "Broadcast Addresses" on page 117.

Network Models

Let's start with a few definitions. A *network model* reflects a design or architecture to accomplish communication between different systems. Network models are also referred to as network *stacks* or *protocol suites*. Examples of network models includes TCP/IP, Sequenced Packet Exchange/Internet Packet Exchange (SPX/IPX) used by Novelle Netware, the Network Basic Input Output System (NetBIOS), which comprises the building blocks for most Microsoft networking and network applications; and AppleTalk, the network model for Apple Macintosh computers.

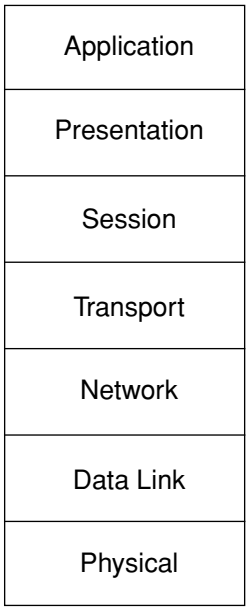
A network model usually consists of *layers*. Each layer of a model represents specific functionality. Within the layers of a model, there are usually *protocols* specified to implement specific tasks. You may think of a protocol as a set of rules or a language. Thus, a layer is normally a collection of protocols.

There are a number of different network models. Some of these models relate to a specific implementation, such as the TCP/IP network model. Others simply describe the process of networking, such as the International Organization

for Standardization/Open System Interconnection Reference Model (ISO/OSI-RM, or more simply, OSI-RM).

OSI-RM

The International Organization for Standardization (ISO) is a worldwide body that promotes standards internationally. In the late 1970s, ISO began work on developing a standard for multivendor computer interconnectivity. The result, published in the late 1980s, was the Open System Interconnection (OSI) model. The OSI model incorporates protocols that can be used to implement a network stack. These protocols are not used extensively largely due to the popularity of the TCP/IP protocol suite. Consequently, the OSI model, with its well-defined layers, is used primarily as a reference model, hence, OSI-RM. Many network models are described by way of OSI-RM and so we provide a description of it here. The OSI-RM is depicted in Figure 1–1.



Layer 7: The Application layer within which all network applications reside.

Layer 6: The Presentation layer provides data representation support.

Layer 5: The Session layer provides for data exchange through dialogs.

Layer 4: The Transport layer provides end-to-end communication.

Layer 3: The Network layer provides internetwork connectivity.

Layer 2: The Data Link layer provides protocols for transmitting and receiving data between directly linked systems.

Layer 1: The Physical layer dictates the physical characteristics of communication.

Figure 1–1 The OSI-RM

As indicated in Figure 1–1, each of the layers are numbered 1 through 7 from physical to application layer.

LAYER 7 All of the capabilities of networking begin in the Application layer. File transfer, messaging, web browsing, and other applications are in this layer. Each such application will appropriately invoke processing of data for transmission through well-defined interfaces to layer(s) below this one.

LAYER 6 The Presentation layer is responsible for data formatting. It takes care of such things as bit and byte ordering and floating point representation. Examples include External Data Representation (XDR) and Abstract Syntax Notation (ASN).

LAYER 5 The Session layer handles the exchange of data through dialog procedures or chat or conversation protocols. This layer is largely designed for mainframe and terminal communications. It has no relevance with respect to TCP/IP networking.

LAYER 4 The Transport layer is responsible for the reliable transfer of data between systems. It manages the communication session including flow control, ordering of information, error detection, and recovery of data.

LAYER 3 The Network layer owns the responsibility of delivering data between different systems in different interconnected networks (*internets*[1]).

LAYER 2 The Data Link layer provides rules for sending and receiving data between two connected nodes over a particular physical medium.

LAYER 1 The Physical layer defines the required hardware, such as cables and interfaces, for a given medium of communication, such as electrical, radio frequency, and light-based. In this way, methods for transmitting and receiving bit-streams of information are defined.

1. The term Internet (upper case I) is used to reference the public internetwork. The term internet (lower case i) is used to generally describe a collection of interconnected networks of which the Internet is one example.

NOTE

There is a great deal more to the OSI model than we have discussed here. For complete details on this standard, visit

```
http://www.iso.ch/cate/3510001.html
```

Also, see "For Further Reading" on page 15 for more resources on this topic.

Next, we discuss the TCP/IP model and begin our journey into the world of TCP/IP networking. We will compare it with the OSI model at the end of the next section.

The TCP/IP Network Model

The TCP/IP network model takes its name from two of its protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Figure 1–2 provides a five-layer[2] representation of the TCP/IP Model. By utilizing a five-layer model, the lower four layers are numbered identically to the lower four layers of the OSI-RM model.

The lower four layers of the model represent functionality performed internally by the Linux kernel. The Application layer includes commands and daemons.

The process of initiating a network communication, like executing `telnet` *`hostname`*, causes the initiator (usually the *client*) to *encapsulate* application data, beginning at the top of the model and moving down, for the network transmission. In other words, each layer wraps the data passed to it by the previous layer with information used to determine where the packet is supposed to go and which service needs to be invoked to handle the application data itself. The information added by each layer is called a *header* when it is prefixed to the data from the previous layer, and a *trailer* when it is suffixed. On the left-hand side of Figure 1–2, you see an increasing number of rectangles as you scan down the layers. The area in gray represents the information added by each layer.

The receiving system, normally the *server*, performs the same steps except in reverse (bottom to top), *deencapsulating* the data. Each layer is responsible for

2. Many texts will use a four-layer representation, combining the Hardware and Network Interface layers.

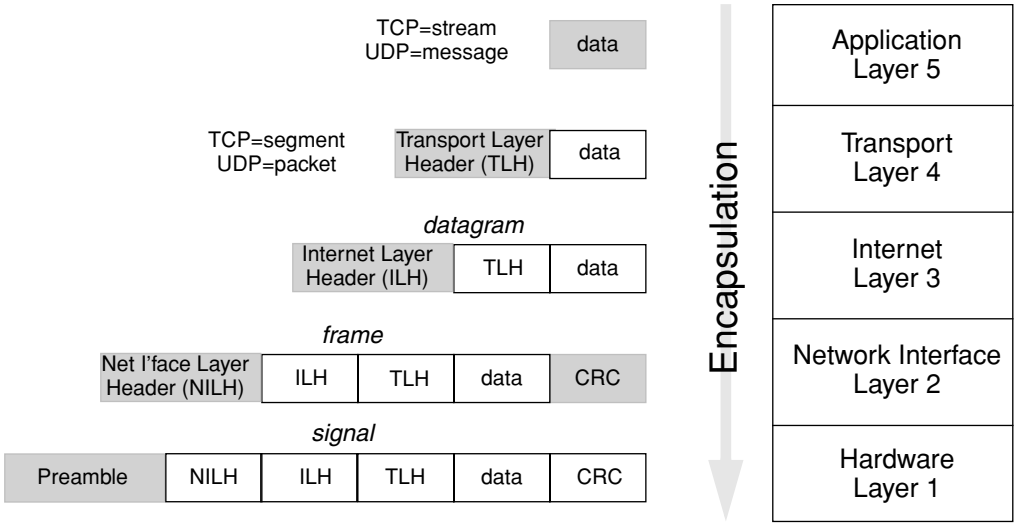


Figure 1–2 The TCP/IP Model

interpreting the header generated by the same layer on the sending system. This process is known as *peer-to-peer* communication.

The information produced during encapsulation, or read during deencapsulation by each layer is ascribed a name as shown in Figure 1–2. For Layer 5, if the underlying Transport layer protocol is the TCP, then the data produced or read by Layer 5 is called a *stream*; if the underlying Transport layer protocol is the User Datagram Protocol (UDP), then it is called a *message*. For Layer 4, if the protocol used is TCP, then the data produced or read by Layer 4 is called a *segment*. Otherwise, if it is UDP, then it is called a *packet*. The data of Layer 3 is called a *datagram*; of Layer 2, a *frame* or *cell*; and of Layer 1, a *signal*.

NOTE

The term *packet* is commonly used instead of signal and/or frame.

In the following sections, we will briefly review the information generated by each layer and its purpose.

The Hardware Layer

The Hardware layer is responsible for exactly that—hardware. This includes cables, interface cards, and repeaters. It accepts the data passed to it by the Network Interface layer and prefixes something called the *Preamble*, which is a well-known sequence of 64 bits used for synchronization purposes. When it finishes its work, it generates a *signal* to be submitted to the media (electrically-based cables in most cases). The Hardware layer also imposes the maximum transfer unit (MTU) used by the Internet layer to ensure that the Hardware layer does not get *frames*[3] that are too large or too small. For Ethernet, the MTU for the signal is 1526 octets, and the minimum signal size is 72 octets.[4]

There are two hardware devices which operate at this layer: *repeaters* and *amplifiers*. A repeater is a device with a number of ports (usually four or more) that is capable of receiving signals, filtering out noise (phenomena not related to the communication at hand), and repeating the signals to every port except the ingress (incoming) port. Amplifiers perform the same task, except that they do not filter noise. Consequently, repeaters are employed in electrical communications environments and amplifiers are employed in light-based communications environments. These devices are often called *hubs* or *concentrators*. We will discuss this layer in greater detail in Chapter 2.

The Network Interface Layer

You may think of the Network Interface layer as a collection of device drivers. Its responsibility is to prepare the data passed to it from the Internet layer for signaling. It does this by prefixing its header (indicated as NILH in Figure 1–2), computing a Cyclic Redundancy Check (CRC—a 32-bit checksum), appending the CRC to the datagram, and passing this information to the device (interface) for signaling in what is called a frame. In particular, this layer understands *physical* addresses (often referred to as Media Access Control [MAC] addresses). When using Ethernet, this is often called an Ethernet address. Physical

3. The data of the Network Interface layer is called a frame when it varies in size. More formally, when using Ethernet, it should be called an *Ethernet frame*. Fixed size data at this layer, such as that used by Asynchronous Transfer Mode (ATM), is called a *cell*.

4. MTU is further discussed throughout the next several chapters, but note that it is more common to quote the Ethernet frame MTU, which is 1518 octets. The minimum Ethernet frame size is 64 octets. Note that frames occur at the Network Access layer. Here we are describing the Hardware layer.

addresses are local and only need to be unique within the local network. For Ethernet interface chipsets, they are 48-bit addresses permanently written into the programmable read-only memory (PROM).

The Network Interface layer writes both the destination and source physical address into its header during encapsulation. Consequently, it is at this layer that, during deencapsulation, initial decisions are made about whether or not to continue processing an incoming frame up the stack. This is discussed in detail in Chapter 2.

There is one device associated with this layer. It is a *switch*. Switches look very much like repeaters, a piece of hardware with at least two network ports, but are more intelligent than repeaters. Since they operate at the Network Interface layer, they are able to make decisions based on physical addresses. Switches are sometimes called hubs or *bridges* or *layer 2 routers*.

NOTE

Switches are sometimes called bridges (or the other way around, if you like). Bridge is an older term that is not commonly used today. Unfortunately, the terms *switching*, *layer 2 switching*, and *layer 3 switching* all confound the issue of what is really being described. Most of the variation in the base term "switch" comes from vendors of switches and routers. An argument could be made that the proper term for the layer 2 device that is capable of making packet-forwarding decisions based on physical addresses is "bridge." A similar argument could be made for the term switch. I could coin a new term, say *swidge*, but I haven't got the courage, and, besides, it would probably make matters worse. After all, companies like Cisco aren't likely to change their terminology just because I say so.

I had to make a decision about which term to use. So I decided to use switch. Throughout this book, I will make no distinction between bridge and switch and will use the term switch to mean a layer 2 device that is capable of making packet-forwarding decisions based on physical addresses.

The Internet Layer

The Internet layer is responsible for a variety of tasks. In order to accomplish these tasks it uses three principal protocols. The Internet Protocol (IP), the Internet Control Messaging Protocol (ICMP), and the Internet Group Man-

agement Protocol (IGMP). The IP is responsible for *routing* and *fragmentation*.[5] The ICMP generates error messages, assists routing through redirection, may implement rudimentary flow control, supports the `ping` command, supports router discovery, and may generate timestamp and netmask queries and responses. The IGMP supports Internet Layer multicasting. Each of these protocols has two available versions: 4 and 6.

The device that operates at this layer is a *router*. Routers are nodes that implement the intelligence of the Internet layer protocols and forward datagrams to the appropriate *networks* or *subnetworks* (discussed in Chapter 4) based on *IP addresses* and the *routing algorithm* (described in Chapter 6). Routers are sometimes called layer 3 switches. Unfortunately, routers are also sometimes called hubs.

The Internet layer produces or reads the Internet layer header. The header contains a lot of information and, in particular, includes the source and destination IP address associated with the packet. There are two versions of the protocols in this layer: version 4 and version 6. IPv4 addresses are 4 octets while IPv6 addresses are 16 octets. These are global addresses, meaning that all nodes throughout a collection of networks that are interconnected (*internet*) must be uniquely identified by this address. Data is passed through such an internet by the process of *routing*. Routing is performed by examining a portion of an IP address in order to determine to which network the data needs to be sent (effectively the purpose of the routing algorithm). Linux systems can act as routers. This header, together with the overall functionality of the Internet layer, is largely discussed in Chapters 4, 5, and 6. Additional details of the Internet layer and Linux router functionality are given in the last six chapters of this book.

The Transport Layer

The Transport Layer is responsible for the end-to-end flow of data. There are two primary protocols used within this layer (TCP and the UDP). An application will use one or the other of these protocols for a given communication. These protocols, the headers they produce or read, and the Transport layer are described in Chapter 7.

5. Fragmentation applies to IP version 4 only.

The Application Layer

This is the layer where all the applications live. These applications are responsible for understanding the data format as well as interpreting the data. Example applications include the Domain Name Service (DNS), the Dynamic Host Configuration Protocol (DHCP), the Network File System (NFS), Samba, electronic mail (e-mail), the file transfer protocol (FTP), and the `telnet` utility. The Application layer is discussed in Chapter 8.

The device that operates at this layer is the *gateway*. Unfortunately, gateway is a term, somewhat like hub, that is used in many ways. We generally define it to mean a link between distinct and/or different computer networks. Often, it is used to refer to a system that is capable of converting from one network protocol stack to another, such as a system that is interconnected into both a TCP/IP network and a Netware network. Gateway is often used to refer to a system that interconnects an internal internetwork and an external network such as the Internet. Other uses of the term gateway are described as they arise.

OSI-RM and TCP/IP Model Comparison

Figure 1–3 depicts the relationship between the OSI-RM and the TCP/IP model.

The relationship between these two models at Layers 1, 2, and 3 is identical. While the diagram in Figure 1–3 shows a one-to-one correspondence at Layer 4, it should be noted that UDP does not perform all of the functions required by the OSI-RM Transport layer; we will explore this fact further in Chapter 7.

The Application layer of the TCP/IP model assumes responsibility for the Session, Presentation, and Application layer of the OSI-RM. This relationship will be detailed further in Chapter 8 and numerous subsequent chapters.

One of the advantages of depicting the TCP/IP model as a five-layer model is that we can refer to Layer 1, 2, and 3 of both models. We will do so throughout this book.

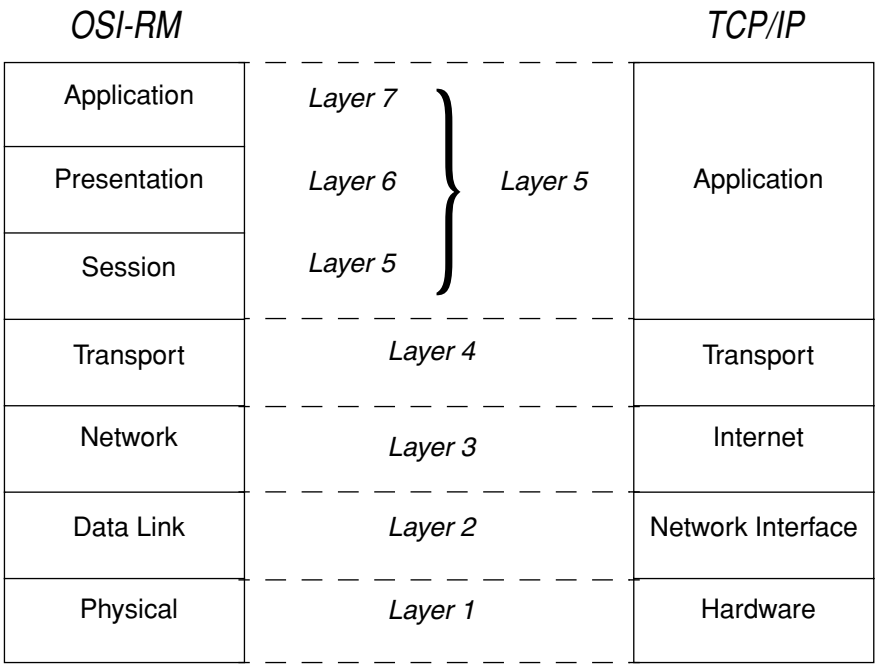


Figure 1–3 OSI-RM and TCP/IP Comparison

The Client-Server Model

It may be rather like stating the obvious, but we define the terms *server* and *client*. A server is a network node that makes compute or data resources available. A client is a network node that utilizes server resources. Of course, for clients and servers to interact, they must be interconnected.

Nodes may act in one, both, or neither capacity of server and client. Generally speaking, from the Linux perspective, the client is the system upon which a command invoking a network-based application is executed. The server is the system on which a *daemon* is running. A daemon is a special program that responds to and handles resource requests. Throughout this book we will deal with, and further refine, these terms.

Request for Comment

The vast majority of the protocols and standards that we discuss throughout this book are embodied in a Request for Comment (RFC), more specifically, an

RFC developed and/or maintained by the Internet Engineering Task Force (IETF).

The overall controlling entity for Internet standards is the Internet Society (ISOC). The Internet Architecture Board (IAB) and the IETF both fall under the ISOC. The IETF does the actual technical work while the IAB handles a lot of the administrative details. The Internet Engineering Steering Group (IESG), also part of the ISOC, provides technical direction for the IAB and the IETF. Our interest is with the work of the IETF and, in particular, with RFCs that provide a standard that is implemented in TCP/IP networking. In addition to *standards track* RFCs (RFCs that have been approved as standards or are becoming standards), the IETF also produces informational and best current practice RFCs that provide details about the implementation and use of some technology without being a standard. You can learn more about the IETF and the associated organizations discussed in this paragraph at

```
http://www.ietf.org/
```

For complete details about IETF RFCs, please see the site

```
http://www.ietf.org/rfc.html
```

A search engine for RFCs can be found at

```
http://www.rfc-editor.org/
```

Where appropriate, we will discuss specific RFCs at various points in this book.

There is another type of RFC maintained by an organization called the Open Group. The Open Group RFCs also define standards and protocols, but are less common. For further information regarding the Open Group RFCs, please visit

```
http://www.opengroup.org/rfc/
```

Throughout this document, all RFCs are IETF RFCs unless specifically identified as an Open Group RFC.

Institute of Electrical and Electronics Engineers (IEEE)

The IEEE is a non-profit technical organization that, among other things, promotes standards for a variety of computing technologies, largely at layers 1 and 2 of the TCP/IP stack. We will be referring to their standards throughout this text. You can learn more about IEEE at their home and standards pages:

```
http://www.ieee.org/
http://standards.ieee.org/
```

Throughout this book we will reference a variety of standards from the IEEE. Table 1–1 lists the high-level IEEE standard number and its general coverage of each standard number.

Table 1–1 IEEE High-Level Standards

STANDARD NUMBER	DESCRIPTION
802.1	High-level architectural overview
802.2	Logical link control (LLC)
802.3	Ethernet with CSMA/CD
802.4	Token bus
802.5	Token ring
802.6	MANs
802.7	Broadband LANs
802.8	Fiber optic LANs
802.9	Data and voice network integration
802.10	Security and privacy
802.11	Wireless networking

For complete coverage of these standards, see the IEEE Web sites referenced above.

There is an excellent write-up of the topics discussed in the last six chapters of this book, including the relationship between IEEE standards and the lower layer ISO standards, at

```
http://www.cs.herts.ac.uk/~simon/networks1/node1.html
```

The Internet, TCP/IP, and Other Stacks

Only a few short years ago you would have been hard-pressed to find 10 people outside of the computer industry who knew about the Internet. Today, it is nearly taken for granted.

The Internet is a global collection of WANs and LANs that are all interconnected. Participation in the Internet ultimately requires TCP/IP capability in the form of a node acting as an Internet gateway or portal. From that perspective, TCP/IP is the network stack of the Internet. There are other network stacks at play within the Internet, and indeed, in a variety of roles throughout the computer industry.

Summary

In this chapter we introduced the OSI-RM and TCP/IP model, briefly described each of their layers, and compared the two models. Additionally, we introduced the client-server model and RFCs. We also defined a number of terms that we will use throughout this text.

For Further Reading

Listed below are books and WWW resources that provide further information about the topics discussed in this chapter.

Books

Comer, Douglas E., *Internetworking with TCP/IP Vol I: Principals, Protocols, Architecture*, 3rd. ed., Englewood Cliffs, New Jersey, Prentice Hall, 1995.
Huitema, Christian, *Routing in the Internet*, Englewood Cliffs, New Jersey, Prentice Hall PTR, 1995.
Perlman, Radia, *Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworking Protocols*, Reading, Massachusetts, Addison-Wesley, 2000.
Stevens, W. Richard, *TCP/IP Illustrated, Volume 1: The Protocols*, Reading, Massachusetts, Addison-Wesley, 1994.

WWW Resources

You will find OSI information at the ISO home page:

```
http://www.iso.ch/
```

IETF RFCs can be found at

```
http://www.ietf.org/rfc.html
```

and Open Group RFCs at

```
http://www.opengroup.org/rfc/
```

For IEEE standards and information, see their home page or standards page:

```
http://www.ieee.org/
http://standards.ieee.org/
```