CHAPTER 1

Network Models and Protocols

Exam Objectives

1.1 Layered Network Models

1.2 The Layers of the TCP/IP 5-Layer Model

1.3 Network Protocols

1.4 Peer-to-Peer Communication

1.5 TCP/IP Protocols by Name and Function

After completing this chapter, you will be able to meet the following Network Administration Exam objectives:

- Identify the purpose of each layer in the TCP/IP 5-layer model.
- Describe the functionality of each of the following Network Protocols: TCP, UDP, IP, and ICMP.
- Describe the relationship between the following Network Protocols: TCP, UDP, IP, and ICMP.
- Describe peer-to-peer communication.

To help you meet these objectives, this chapter covers the following topics:

- layered network models
- the layers of the TCP/IP 5-layer model
- network protocols
- peer-to-peer communications
- TCP/IP protocols by name and function

1.1 Layered Network Models

This chapter first introduces layered network models and then describes the services provided by each layer of the model. We then briefly describe, in the context of a *protocol stack*, the *network protocols* that provide the services to upper layer protocols or applications at each layer. You will learn about the features of the most important network protocols, TCP/UDP/IP and ICMP, and this information will serve as the foundation for later chapters that cover these protocols in greater detail. This gradual or phased introduction of the important network protocols will allow you to understand the basics of each protocol before we explore their more complex aspects.

Network protocols are modular by design and function at specific layers of a hierarchical protocol stack. Each layer in the hierarchy provides services to the layer above it and uses the services of the layer beneath it. There are instances in which nonadjacent layers communicate directly, but these are exceptions to the rule.

Through this hierarchy, each layer provides an *abstraction* to the layer just above it. This abstraction is desirable, as upper layers need not know how their data is routed across the Internet, or over which network their data will travel.

To understand how applications such as **sendmail**, **telnet**, and **ftp** interface with the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols, we must examine how the protocols communicate with each other and how they offer a service to applications. Each protocol was designed to offer a service to another protocol or application and will be explored in that context.

We can best view the protocols as an ordered stack of modules based on a set of hierarchical relationships. The hierarchy is of fundamental importance because it explains and exposes not only the relationships among the interacting protocols but also the properties of each protocol, revealing why a particular protocol is able meet the requirements of a particular application.

There are many protocol families and models. This book explores two models, which are covered by Sun course SA-389, Solaris Operating Environment TCP/IP Network Administration:

- the OSI/ISO 7-layer reference model
- the TCP/IP Sun/DoD 5-layer model

The two models are different in several respects, although both perform the same function, which essentially is to reveal the hierarchical, modular nature of network protocol design and operation. The network models also provide guidance for network protocol designers. Throughout this book, we will refer to the Open Systems Interconnection OSI/ISO 7-layer reference model simply as the OSI model and to the DoD TCP/IP Sun/DoD 5-layer model as the TCP/IP model.

The OSI/ISO 7-Layer Reference Model

The OSI/ISO 7-layer reference model was created in the early 1980s. Table 1.1 shows the seven layers of the model.

Table 1.1 *Layers of the OSI Model*

LAYER (NUMBER)	DESCRIBES/DEFINES
Application (7)	Applications and network services
Presentation (6)	The way data is presented
Session (5)	Manages connection terms of a session
Transport (4)	End-to-end messaging between applications
Network (3)	Data addressing and delivery between networks
Datalink (2)	Error detection and packet framing across a physical network
Physical (1)	Network hardware, electrical voltage and current

Note that:

- The OSI model was developed by the International Standards Organization (ISO).
- The layers of the OSI model are numbered from the base upward.
- The Physical layer (1) is at the base and the Application layer (7) is at the top.
- The OSI model is a generic networking model.
- The OSI model was designed in the early 1980s and intended for multiple manufacturers and standards.
- The OSI model was originally focused on open systems and interfacing multiple stacks.

- Chronologically, the OSI model was created long after the TCP/IP family of protocols.

The model is to some degree an ideal, as it does not pertain to any specific protocol family, but rather provides a framework within which network protocol designers and hardware manufacturers may work as they strive to produce modular products.

We next outline the TCP/IP model and compare and contrast it with the OSI model.

The TCP/IP 5-Layer Model

The Department of Defense (DoD) TCP/IP 5-layer model was created in 1969. Table 1.2 shows the layers of this model and the service provided by each layer.

Table 1.2 *Layers of the TCP/IP Model and Purpose of Each Layer*

Layer (Number)	Purpose
Application (5)	Reserved for applications and protocols
Transport (4)	Provides end-to-end delivery service for layer 5 applications and protocols
Internet (3)	Provides a network routing service to upper layers
Network (2)	Provides a framing service to the Internet layer
Physical (1)	Provides an electrical signal bit transmission service to the network

With this model, aimed specifically at TCP/IP conventions, we can identify the protocols at each layer, as shown in Table 1.3.

The most important points to note about the TCP/IP model are:

- The TCP/IP protocols were developed and funded by the USA DoD for purposes of research and experimentation.
- The TCP/IP model was conceived in 1969.
- The TCP/IP model accommodates only the TCP/IP protocols.
- The TCP/IP model has only five layers.

Table 1.3 *Layers of the TCP/IP Model and Entities That Function at Each Layer*

LAYER (NUMBER)	NETWORK COMPONENT THAT OPERATES AT THIS LAYER
Application (5)	HTTP, FTP, telnet, SMTP, NTP, POP, IMAP, and others
Transport (4)	TCP/UDP
Internet (3)	IP, ICMP, ARP, RARP
Network (2)	Data Link: Ethernet, Token Ring, FDDI, ATM, and others
Physical (1)	Coaxial, fiberoptic, twisted pair

Benefits of Using Network Models: A Summary

It is beneficial to consider the organization of any network model because the network model

- reveals the hierarchical, modular nature of network protocol design and implementation.
- enables us to think in terms of each protocol performing a given function or service at a specific layer.
- visually reveals a host's protocol stack as implemented in the kernel.
- reveals the order of the protocol stack.

The striking differences between the models are shown in Table 1.4.

Table 1.4 *Differences Between the OSI 7-Layer and TCP/IP 5-Layer Models*

OSI MODEL	TCP/IP MODEL
Devised 1983	Devised 1969
Created by ISO	Created by USA DoD
Multiple vendors/multiple protocols/ ISO protocols	TCP/IP protocol family
Seven layers	Five layers
Generic networking model	TCP/IP-specific model

EXAM NOTES

KEY LEARNING POINTS

- **The OSI model is used as a frame of reference when describing protocol architectures and functional characteristics.**
- **The TCP/IP model is specifically intended for the TCP/IP family of protocols.**
- **The TCP/IP model and protocols chronologically predate the OSI model and protocols.**
- **Traditional TCP/IP applications talk directly to the Transport layer and have no distinct session and presentation layer protocols.**

Next, we examine the TCP/IP 5-layer model in more detail to illustrate which protocols operate at each of the layers. The TCP/IP model and the protocols that function within each of its layers constitute the basis for the rest of this book. A brief description of each protocol appears in this chapter; later chapters examine the individual protocols in greater detail.

1.2 The Layers of the TCP/IP 5-Layer Model

A closer examination of the layers of the TCP/IP 5-layer model follows. The layers of the protocol stack are numbered, as you saw in Table 1.3. This examination starts at the Application layer (layer 5) of the protocol stack and works downward to the level of the Physical layer (layer 1) at the bottom of the stack.

Application Layer (5)

There are literally hundreds of applications that function at the Application layer, and more are being developed as the Internet evolves. Well-known TCP/IP applications include:

- Web browsers/servers using the HyperText Transfer Protocol (HTTP)
- mail applications and related protocols
  - Post Office Protocol (POP)

  - Internet Message Access Protocol (IMAP)
  - Simple Mail Transport Protocol (SMTP)
- File Transfer Protocol (FTP)
- **telnet** and **rlogin**
- Domain Name Service (DNS)
- Network File System (NFS)
- Network Information Service (NIS) and Network Information Service Plus (NIS+)

Application layer protocols and applications are unable to deliver their own data across a network or Internet unassisted and so need to be *encapsulated* in a Transport protocol such as TCP or User Datagram Protocol (UDP). The Transport layer provides this *transport service* (also known as a *delivery service)* to the Application layer. The Transport layer protocol in turn relies on the Internet Protocol (IP) to provide an end-to-end routing service.

You will visit these concepts again when IP (Chapter 5), routing (Chapter 6), TCP/UDP (Chapter 7), and the Client-Server model (Chapter 8) are investigated in greater detail.

Focus on the following important points that summarize features of the Application layer. Chapter 8 examines how these applications interact with Transport layer protocols.

EXAM NOTES

KEY LEARNING POINTS

- **Many protocols function at the Application layer.**
- **HTTP, SMTP, POP, FTP, and DNS are the most heavily used Application layer protocols on the Internet.**
- **Web browsers, mail clients, news readers, ftp, telnet, sendmail are just a few of the thousands of layer 5 applications.**
- **The Application layer uses the Transport layer for delivery.**
- **Users often interact directly with Application layer programs.**
- **The Application layer (5) is at the top of the stack.**

Transport Layer (4)

TCP and UDP are the only two protocols that function at the Transport layer (4). They encapsulate or carry the layer 5 protocols and offer an end-to-end transport service. They accept data from a client network application on a client host and deliver it to the server application on the server host that is providing the client with the service. The client and the servers are usually on different systems and therefore need a network to connect them. Data travels between the client and server across one or more networks.

For example, a **telnet** client on the client host needs to reach the **in.telnetd** server daemon running on the server host. The **telnet** client process uses the TCP Transport layer protocol to connect to the **in.telnetd** server process (which usually exists on a different system). Other Application layer protocols use the UDP transport protocol, which offers a nonguaranteed transport service, trading guaranteed delivery for speed and minimized overhead. DNS queries, for example, use UDP, as speed is of the essence, and failure considerations are not so critical.

The choice of using either TCP or UDP at the Transport layer is made by the network programmer and is based on the type of service required. Some well-known network protocols that function at the Application layer use both TCP and UDP for different functions—DNS, for example, which is fully explored in Chapter 11. Some applications that were originally designed to use UDP, such as Sun's NFS that allows file sharing between systems, have switched to using TCP. Version 2 of NFS used UDP, but version 3 uses TCP.

Focus on the following important points before proceeding to the next section on the Internet layer.

EXAM NOTES

KEY LEARNING POINTS

- **Most Solaris applications use TCP and UDP for end-to-end delivery.**
- **TCP and UDP are transport protocols.**
- **TCP and UDP are encapsulated in IP.**
- **TCP and UDP are not able to route their own data and so use IP to perform this task.**

Chapter 7 explores the Transport layer protocols (TCP and UDP) in detail, examining why a given application may be better suited to using TCP than UDP, or the converse. In Chapter 8 we examine the client-server model and learn how the applications that use TCP/UDP are configured.

We next examine the Internet layer, on which the Transport layer relies for routing the data it is transporting. The Transport layer provides client and server applications with an end-to-end delivery service (which is not the same as routing), which you will see in Chapter 7, whereas IP (described in Chapter 6) provides an end-to-end IP datagram routing service.

Internet Layer (3)

At every other layer of the TCP/IP network model, multiple protocols exist to carry out similar tasks. The Internet layer is an exception, with only the IP protocol capable of performing the critical routing function. Although other protocols function at the Internet layer, only the IP protocol offers a datagram-based routing service to the upper layers of the stack. IP is the only protocol able to encapsulate and route packets to a destination IP address (identified in the IP datagram header), understand the IP address scheme, and route data for upper-layer protocols such as TCP, UDP, and ICMP. These, in turn, carry encapsulated application layer data.

The Internet layer protocol, IP, makes the *routing decision* and handles the routing of IP datagrams. IP is also responsible for the fragmentation of IP datagrams, which might be necessary if the underlying Network Interface layer (2) demands it. Chapter 5 covers IP fragmentation in detail.

Note

IP does not build the routing tables; it merely uses the kernel routing table information in an attempt to route the IP datagram(s), as you will learn in Chapter 6.

IP was designed to run over any LAN or WAN technology. An IP's address is essentially a logical or software address that is *mapped* onto a Network layer (2) address so that IP datagrams can be packaged up *(framed)* and carried across a specific type of Network Interface layer technology, such as Token Ring or Ethernet. ARP manages the mapping between the Internet layer (3) where IP resides and the Network Interface layer where Ethernet resides. Chapter 4 examines ARP further. Chapter 5 covers IPv4 in detail and Chapter 13 explores IPv6.

EXAM NOTES

KEY LEARNING POINTS

- **IP makes the routing decision and routes IP datagrams.**
- **IP routes datagram(s) to the destination IP address in the IP header.**
- **IP datagrams typically carry (encapsulate) TCP/UDP application data.**
- **IP functions at the Internet layer.**
- **IP's header is a minimum of 20 bytes and can be larger.**
- **IP's unit of data is an IP datagram.**

Network Interface Layer (2)

Ethernet functions at this layer. There are alternatives to Ethernet at this layer, but Ethernet is dominant in terms of UNIX LANS and is the only LAN examined in detail in this book. Ethernet frames or encapsulates the data it is carrying, prior to transmitting its packet onto the Ethernet bus.

Network Interface layer technologies have Maximum Transmission Units (MTUs) of various sizes. (The MTU determines the maximum number of bytes that can be transmitted in a single frame, not including the headers.) As Table 1.5 shows, MTU size varies according to the Network Interface layer technology.

Table 1.5 *Network Layer MTUs*

NETWORK TYPE	MTU (BYTES)
Hyperchannel	65,535
16 MB/second Token Ring (IBM)	17,914
4 Mbits/second Token Ring (IEEE 802.5)	4,464
Fiber Distributed Data Interface (FDDI)	4,352
Ethernet	1,500
IEEE 802.3/802.2	1,492
Point to Point (low delay)	296

Ethernet's MTU is especially significant because it determines whether an IP datagram carried by Ethernet needs fragmenting. See Chapter 5 for further information.

EXAM NOTES

KEY LEARNING POINTS

- **The Network Interface layer MTU determines the maximum number of bytes carried in a single frame.**
- **Ethernet is a Network Interface layer technology.**
- **Network Interface layer technologies have MTUs of various sizes.**

Physical Layer (1)

This layer at the base of the stack identifies the LAN transmission media, typically fiberoptic or wire-based media. This layer concerns the transmission of bits across a physical media with very little interpretation of the data.

Chapter 3 examines the copper wire and fiberoptic-based media used by Ethernet, FDDI, and ATM in greater detail.

So far, we have examined the network protocols in the context of a network model. The ensuing discussion of the most important network protocols reveals their features and provides a foundation for chapters that follow.

1.3 Network Protocols

This section is a brief introduction to TCP, UDP, IP, and ICMP. First, we briefly examine the two transport protocols, TCP and UDP, and then take a cursory look at IP and ICMP.

Transport Layer Protocols: TCP and UDP

We briefly mentioned the TCP and UDP protocols in the context of the TCP/IP model described earlier in this chapter. We now examine the Transport layer in greater detail.

The Transport layer offers either a high-overhead, guaranteed, connection-oriented transport service such as that offered by TCP, or a low-overhead, connectionless, nonguaranteed service such as the one offered by UDP. See Chapter 7 for more information about the Transport layer protocols and Chapters 5 and 13 for details about the IP protocols (IPv4, IPv6) that route the Transport protocols data.

Transmission Control Protocol (TCP)

TCP provides a connection-oriented, guaranteed, reliable transport service. TCP has significant overhead because every byte of transmitted data is both acknowledged and sequenced by both ends of the connection. The data itself is delivered in one or more *segments*, which are TCP's basic units of data. In addition, TCP establishes a connection at the Transport layer, which must be formed before transmission of client data may occur over the Transport layer.

Although TCP guarantees data delivery, it must be remembered that TCP is routed over IP and therefore uses the same underlying packet-based routing service as UDP—namely, IP, which is not guaranteed. Both TCP and UDP use IP to route their data, but they impose different levels of reliability and overhead as determined by their own specifications as Transport layer protocols. TCP extensively monitors the quality of data sent and received, whereas UDP has only a simple checksum error checking method, which is often turned off and therefore ignored.

The cost of TCP reliability, as just stated, is significant overhead. The TCP protocol sends a sequenced (numbered) *byte stream* of segmented data. The sequence numbering of bytes ensures that the destination TCP protocol at the receiving end orders the data stream correctly. Unfragmented IP datagrams that arrive in the wrong order (which have encapsulated TCP segments) are simply passed up the stack to TCP, which ensures that the data is delivered to the application in the correct order. Fragmented IP datagrams, on the other hand, are reassembled by IP itself before being passed to TCP based on fragment offset information stored in the IP header. Some example TCP client applications are

telnet, ftp, and **sendmail**, all of which require a guaranteed delivery service and therefore use TCP rather than UDP.

EXAM NOTES

KEY LEARNING POINTS

- **TCP is a connection-oriented protocol.**
- **The segment is the TCP unit of data.**
- **TCP is a Transport layer protocol.**
- **TCP is encapsulated and routed by IP.**
- **TCP is said to offer a guaranteed, reliable, connection-oriented service that creates significant overhead.**

Essentially, if data security must be guaranteed, TCP is a better choice than UDP, although more expensive in terms of both the amount of control data transmitted and the control overhead incurred. Chapter 7 examines the TCP protocol in greater detail.

User Datagram Protocol (UDP)

UDP is a connectionless transport protocol, which means that no connection is established at the Transport layer prior to data being sent between client and server applications.

Some applications—for example, a router propagating routing information every 30 seconds—can afford to lose an occasional data packet. Routing clients that miss the occasional routing table update do not usually suffer adverse effects. The Routing Information Protocol (RIP) uses UDP. DNS queries also use UDP, as speed is more important than reliability for this application. UDP applications tend to send small packets that can be transported in a single UDP datagram. UDP has an optional checksum error check, which introduces minimal overhead and is usually turned off.

To check the UDP checksum variable (**udp_do_checksum**) under Solaris 8 use **ndd**:

```
# ndd -get  /dev/udp udp_do_checksum
0
```

A value of 0 means false; that is, disable the UDP checksum feature. A value of 1, which means true, indicates that the UDP protocol checksum feature is enabled.

To enable the UDP checksum feature, issue the following command:

```
# ndd -set /dev/udp udp_do_checksum 1
```

To check the current value of **udp_do_checksum**:

```
# ndd -get  /dev/udp udp_do_checksum
1
```

EXAM NOTES

KEY LEARNING POINTS

- **UDP is a connectionless protocol.**
- **UDP is a transport protocol.**
- **UDP is encapsulated and routed in IP.**
- **UDP is not guaranteed.**
- **UDP does minimal error checking.**
- **UDP has a simple checksum for error checking.**
- **UDP's header is only eight bytes.**

Chapter 7 looks at the UDP protocol in greater detail.

Internet Protocol (IPv4)

IPv4 is a connectionless protocol like UDP. Unlike UDP, however, IP is not a transport protocol but instead offers a datagram routing service to the Transport layer (a datagram is the unit of data for the IP layer). The Transport layer protocols, TCP and UDP, use IP to route their client data between application client and server hosts. It is worth stressing, therefore, that IP routes data between hosts, and in effect, between the Transport layers on end-to-end hosts, but not between clients and server processes. The Transport layer protocols, TCP and UDP, transport the data between client and server processes, using IP to form a bridge between hosts. IP as a protocol is responsible for the following:

- fragmenting of IP datagrams if the Network Interface layer MTU demands it
- reassembly of IP fragments
- making the IP datagram routing decision

EXAM NOTES

KEY LEARNING POINTS

- **IP is a connectionless protocol.**
- **IP is an Internet layer protocol.**
- **IP is encapsulated in many different datalink technologies such as Ethernet, Token Ring, and Point-to-Point Protocol (PPP).**
- **IP is not guaranteed.**

Chapter 5 looks in detail at IPv4, and Chapter 13 explores IPv6. Next we look at ICMP, which also functions at the Internet layer (3).

Internet Control Message Protocol (ICMP)

ICMP is considered almost part of the IP protocol, even though it is a separate protocol in its own right. ICMP generates messages, which receive a response either from the IP layer itself or a higher layer protocol such as TCP or UDP. Although ICMP is at the same layer as IP, it is not an alternative to IP; rather, it assists IP with IP error detection and correction.

ICMP data is encapsulated in IP datagram(s), as is TCP and UDP data, but ICMP is not a Transport layer protocol. Error messages such as **network unreachable**, **host unreachable**, and **port unreachable** are examples of ICMP error messages.

See Chapter 5 for further details about ICMP.

EXAM NOTES

KEY LEARNING POINTS

- **ICMP is a connectionless protocol.**
- **ICMP is an Internet layer protocol.**
- **ICMP is unusual in that it is encapsulated in IP but is not a transport protocol.**
- **ICMP assists IP with error detection and correction.**
- **ICMP generates two types of messages: error and query.**

1.4 Peer-to-Peer Communication

The term *peer-to-peer* identifies two communicating entities, functioning or operating at the same layer of the stack, as *peers*. The peers are usually on different systems that are connected by one or more networks, as shown in Figure 1-1.

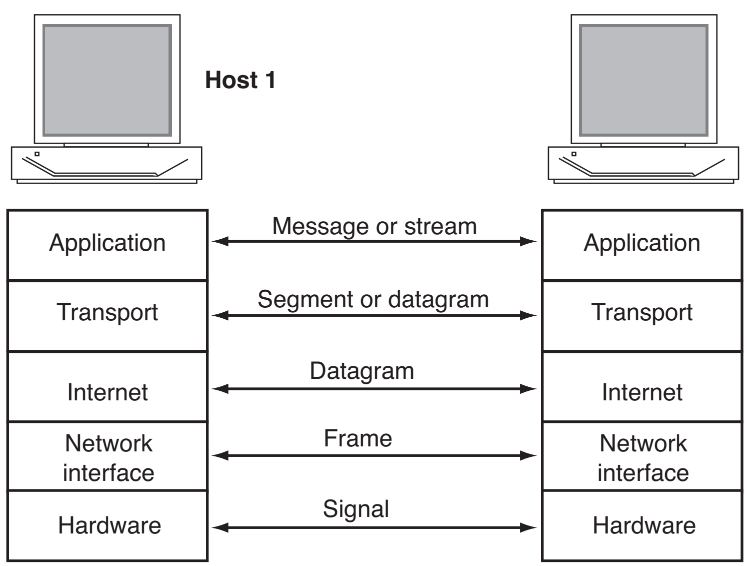

Figure 1–1 *Peer relationships*

EXAM NOTES

KEY LEARNING POINTS

- **The two ends of a communication are at the same layer of the stack.**
- **The peer entities may be application processes or protocols.**

The following section identifies protocols by the Requests for Comments (RFCs) that formally describe them.

1.5 TCP/IP Protocols by Name and Function

In this chapter we already looked briefly at some TCP/IP protocols. It is, however, worth summarizing each protocol formally in terms of its name and the RFC that describes it. Some of these protocols will be explored in later chapters as indicated in Table 1.6.

The following list of RFCs and protocols is not definitive. Listed are the most useful RFCs, but not all RFCs that relate to the subject, as the list is extensive for the more complex protocols. The RFC column of the table lists the chapter of this book that covers the protocol where applicable. We work down the stack from the Application layer (5), again, taking the journey that application data takes.

The Application layer has literally hundreds of applications and many protocols that the applications at this layer use. They are listed in Table 1.6 by relevant RFC, protocol name, and a brief description.

Table 1.7 lists the important Transport layer protocols. Chapter 7 explores these in detail.

Table 1.8 describes the protocols of the Internet layer. Chapter 4 covers ARP/RARP, and Chapters 5 and 13 cover IPv4 and IPv6, respectively.

Table 1.9 shows the Network layer protocols, with many technologies to choose from. Ethernet is by far the most important LAN technology to understand. Alternatives such as Token Ring and Token Bus exist, but they are far less important than Ethernet.

Table 1.6 *Application Layer Protocol Descriptions*

RFC	Protocol	Description
1034,1035 *Chapter 11*	DNS	Domain Name Service is a text-based, distributed host-to-IP address (and IP address-to-host) mapping solution based on Berkeley Internet Name Domain (BIND) library routines. It also provides, through mail exchanger (MX) information, the IP addresses and names of servers that accept inbound e-mail for given domain names.
959	**ftp**	File Transfer Protocol transfers a file by copying a complete file from one system to another system.
854,855	**telnet**	This service provides a remote connection and login facility, which runs over any TCP/IP network.
1258,1280	**rlogin**	This service, offered by UNIX systems, enables users of one machine to connect to other UNIX systems across the Internet and to interact as though their terminals were connected directly to the machines.
2131 *Chapter 9*	DHCP	Dynamic Host Configuration Protocol automates the assignment of IP addresses in an organization's network.
821	SMTP	Simple Mail Transfer Protocol transfers electronic mail messages from one machine to another.
1157	SNMP	Simple Network Management Protocol is the language that allows for the monitoring and control of network entities.
1939	POP3	Post Office Protocol lets users collect e-mail from a POP server over any TCP/IP network. The user's host runs a POP client application, and the server runs the POP server software, which gives the user authenticated access to his or her mailbox.
2060	IMAP4	Internet Message Access Protocol, like POP, lets users collect e-mail from an IMAP server over a TCP/IP network. The user's host runs an IMAP client application, and the server runs the IMAP server software, which gives the user authenticated access to his or her mailbox. IMAP4 is more powerful and offers more features than POP3.
1945,2068	HTTP	HyperText Transfer Protocol is used by the World Wide Web to display text, pictures, sounds, and other multimedia information with a Web browser.

Table 1.7 *Transport Layer (4) Protocol Descriptions*

RFC	Protocol	Description
793 *Chapter 7*	TCP	Transmission Control Protocol is a connection-oriented protocol that provides the full-duplex, stream service on which many application protocols depend. It is encapsulated in IP.
768 *Chapter 7*	UDP	User Datagram Protocol provides a datagram delivery to application and is encapsulated in IP.

Table 1.8 *Internet Layer (3) Protocol Descriptions*

RFC	Protocol	Description
826 *Chapter 4*	ARP	Address Resolution Protocol defines the method used to map a 32-bit IP address to a 48-bit Ethernet address.
903 *Chapter 4*	RARP	Reverse Address Resolution Protocol is the reverse of ARP. It maps a 48-bit Ethernet address to a 32-bit IP address.
791,950 919,922 *Chapters 5 and 13*	IP	Internet Protocol determines the path a datagram must take, based on the destination host's IP address.
792 *Chapter 5*	ICMP	Internet Control Message Protocol communicates error messages and other controls within IP datagrams.

Table 1.9 *Network Layer (2) Protocol Descriptions*

RFC	Protocol	Description
1055	SLIP	Serial Line IP encapsulates IP datagrams on serial lines.
1661	PPP	Point-to-Point Protocol transmits datagrams over serial point-to-point links.
826, 894 *Chapter 3*	Ethernet	Ethernet is a broadcast-based, contention-bus LAN technology.

EXAM NOTES

KEY LEARNING POINTS

- **The OSI/ISO 7-layer reference model is a generic network model.**
- **The DoD 5-layer model is specifically for the TCP/IP protocol family.**
- **TCP and UDP are transport protocols that provide a delivery service to the Application layer.**
- **IP is an Internet layer protocol that offers a datagram-based routing service to the Transport and Application layers.**

SUMMARY

This chapter explored network models, specifically the OSI 7-layer model and the TCP/IP 5-layer model. Having contrasted the models, it should be apparent that OSI is a generic model in contrast to the TCP/IP model, which is specifically aimed at the TCP/IP family of protocols. An examination of the protocol stack revealed a hierarchical approach to network management and a modular approach to protocol implementation and design. The five layers of the TCP/IP model were examined and the main network protocols at each layer introduced. A discussion of peer-to-peer communications described how the communicating protocols or applications are considered peers when they are both functioning at the same layer in the stack.

Finally, we provided a tabular summary of relevant protocols and applications that function at various layers of the protocol stack, listed by RFC, protocol, description, and the chapter of this book that discusses them.

TEST YOURSELF

MULTIPLE CHOICE

1. *Which of the following are layers of the OSI/ISO 7-layer reference model? Choose three.*

   A. Session layer

B. Application layer
C. Semblance layer
D. Service layer
E. Presentation layer

2. *Which of the following are layers of the TCP/IP 5-layer model? Choose three.*
A. Transmit layer
B. Physical layer
C. Transient layer
D. Transport layer
E. Internet layer

3. *At which layer of the TCP/IP 5-layer model does Ethernet function?*
A. Application layer
B. Transport layer
C. Internet layer
D. Network Interface layer
E. Physical layer

4. *At which layer of the TCP/IP 5-layer model does the UDP protocol function?*
A. Application layer
B. Transport layer
C. Internet layer
D. Network layer
E. Physical layer

5. *What does the acronym UDP mean?*
A. User Distance Protocol
B. Uniform Data Protocol
C. United Datagram Protocol
D. User Datagram Protocol
E. Ultimate Datagram Protocol

6. *What does the acronym TCP mean?*
A. Transport Control Protocol
B. Transmission Counter Protocol
C. Transmission Control Protocol
D. Transport Concert Protocol

E. Transmission Count Protocol

7. *Which word describes communicating layers across hosts that are at the same layer within the stack?*
   A. partners
   B. peerers
   C. primers
   D. peers
   E. players

8. *Which protocol is ICMP encapsulated in?*
   A. TCP
   B. IP
   C. UDP
   D. ARP
   E. RARP

9. *What is the size of the UDP header in bytes?*
   A. 10
   B. 20
   C. 12
   D. 4
   E. 8

10. *What is the minimum size of the IPv4 header in bytes?*
   A. 20
   B. 18
   C. 12
   D. 40
   E. 25

FREE RESPONSE

1. State in your own words three advantages of modularizing the design of network protocols.
2. List the similarities and differences between IP and UDP as protocols.