Chapter 1: Introducing TCP/IP

Exam Objectives

- **Understanding the purpose of TCP/IP**
- **Revealing the components of TCP/IP**
- **Exposing the OSI reference model**
- **Understanding the function of each layer in the OSI model**
- **Examining data encapsulation**
- **Comparing differences in the DoD and OSI models**

In almost every aspect of our daily computing lives, we are communicating, researching, and relying on the Internet, a culmination of some of the finest inventions ever conceived. From the humble beginnings of the telephone and telegraph to radio broadcasts and computers, the Internet has allowed us to "get connected." From online banking to online dating, more than 1.5 billion users use this great tool for education and entertainment, conducting business, and connecting globally with others. Just as we use speech and more than 6,800 different languages to transmit thoughts and ideas, this interconnected web of networks also needed its own language for communication. This language is TCP/IP.

Transmission Control Protocol (TCP) and Internet Protocol (IP) are a set of protocols — a standard set of rules that control and enable communication among computers — developed to allow data exchange and sharing of resources across a network. All hosts that speak this same language on the network can understand one another and communicate together. This language, or protocol, defines how messages are formatted and how errors are handled. A networking world without these rules and protocols in place would probably resemble the ancient Tower of Babel.

You must have a strong understanding of TCP/IP protocols, the OSI reference model and its seven layers, and TCP/IP encapsulation methods to do well on the exam. You will see many questions on the test regarding TCP/IP and its suite of protocols. I briefly examine the Open Systems Interconnection (OSI) and DoD (Department of Defense) conceptual models that provide a framework for TCP/IP and other protocols. I also cover the different components of TCP/IP and data encapsulation. For an in-depth review of the TCP/IP layers and protocols, see Book II, Chapter 2.

TCP/IP communication

TCP/IP allows many different operating systems and computer platforms to interoperate with one another seamlessly over a computer network. A computer network is a collection of computers or devices that communicate together over a shared transmission medium. This transmission medium is the physical cabling and network interfaces that direct and support network traffic. And whether you are running Microsoft Windows, Sun Solaris, Ubuntu Linux, Novell Netware, or MAC OS X, TCP/IP will function regardless of operating system or hardware manufacturer type. This allows all of these different entities to interact and understand one another using one language, a great advantage of TCP/IP.

TCP/IP is designed with one major goal in mind; to *decentralize* network communications. If a single node on the network fails, other devices continue to function independently and are uninfluenced by the failed system. Unlike centralized management, this gives nodes an equal share and priority on the network. This is a key feature of TCP/IP. A decentralized network using TCP/IP is enhanced further by using decentralized features of dynamic routing, which you find out more about in Chapters 2 and 3 of Book II.

TCP/IP is based on a defined set of rules for how data communication protocols operate; these rules are defined in the Open Systems Interconnection (OSI) model. This model examines each function of the protocol stack within TCP/IP and computer networking. Actually, TCP/IP may be better defined using the four-layer Department of Defense (DoD) model approach I discuss in the section "TCP/IP in the DoD model," later in this chapter.

We pioneered this

In the late 1960s, the Advanced Research Projects Agency (ARPA) was tasked with finding an efficient means of connecting various computer sites for the purpose of sharing research data. It has been rumored that ARPA began investigating ways of creating this reliable interconnection of remote networks with the sole purpose that these networks would continue to function during, and even withstand, a nuclear attack in wartime scenarios. This is, in fact, a myth. Although network survivability during major losses of the network was highly desired, especially by the U.S. government, ARPA's need for a robust network was more due to unreliable equipment and its connecting links than, say, a nuclear threat.

Combined with valuable experimental research on data transfer technology from laboratories in the United States, France, and the United Kingdom, ARPA successfully created a topology-independent, *packet-switching* network that allowed communication among all types of operating systems and hardware platforms. This open-architectural design of ARPANET would grow over the years to eventually form what we now know as the Internet, an internetworking of networks.

The goal of this new packet-switching network was to optimize traffic load and ensure delivery of data, increasing robustness of communication. The sending computer would assign each packet with a destination address and sequence number, and each block of data could then be routed independently yet still be reassembled in proper order by the receiving node, allowing multiple packets to travel different paths to the desired destination. Reassembly of these packets in the correct sequence would then be processed by the receiving computer.

Splitting data into packets would prove to be much more desirable than *circuit-switching* technology, which transmitted data from source to destination using a dedicated, reserved connection. The inflexible nature of circuit switching would make the line unavailable for other connections until the call was terminated. Messages could be transferred all at once using the entire available bandwidth of the circuit, but the main disadvantage was that no other connection could use the bandwidth until the connection was terminated and a new one established. ARPA would later adopt packet-switching technology as the foundation for ARPANET, with packet-switching proving to be the core method of internetwork communications.

Components of TCP/IP

The following list describes the components that make up the TCP/IP suite:

- **Internet Protocol (IP):** The most important, underlying protocol of the TCP/IP suite, *IP* handles data exchange between computers using small data packets called *datagrams,* determining the best route for delivery. Any available path chosen by the datagram may be taken to reach the destination. IP does not handle delivery verification like TCP does and is only concerned with sending and receiving data. Check out Figure 1-1 for an example of IP and TCP packet headers.
- **Transmission Control Protocol (TCP):** *TCP* is a connection-oriented protocol responsible for reliable delivery of data between applications, ensuring that data sent by the source machine is received properly by the destination machine. TCP enables the sending of messages from both source and destination machines to communicate connection status and inserts header information into each packet, allowing error-free delivery and checking for unauthorized modification of packet data.
- **User Datagram Protocol (UDP):** A *connectionless,* best-effort protocol useful for sending datagrams in any order and without delivery guarantee, assuming that other applications or protocols can handle the required error checking and handshaking. Having less data to transmit, UDP datagrams are smaller and delivered faster than TCP packets, but lack the reliable nature that TCP provides. As you can see in Figure 1-2, UDP does not provide error checking and handshaking as does the more reliable TCP.

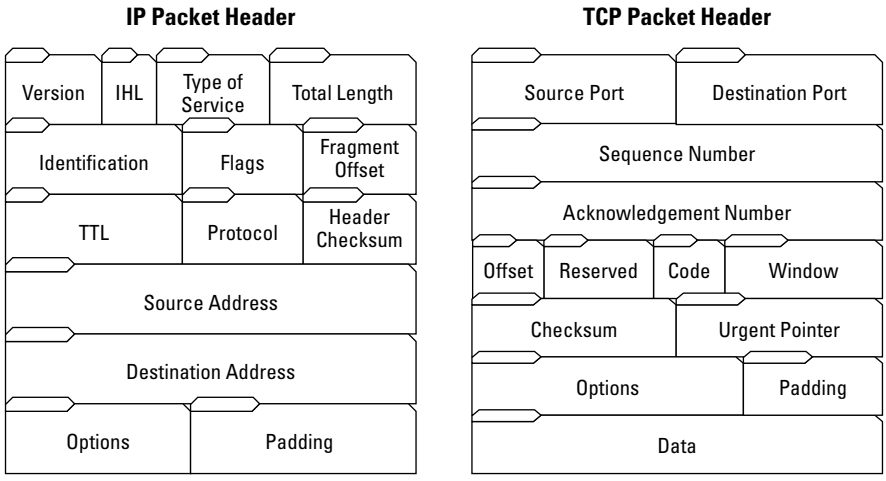
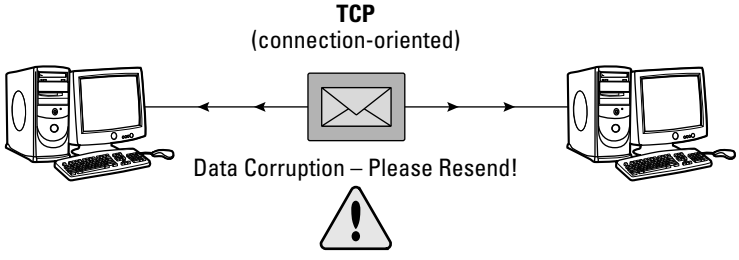


Figure 1-1: IP and TCP packet headers.



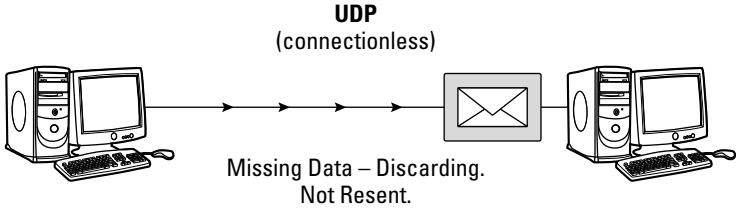


Figure 1-2: TCP versus UDP communications.

- **Hypertext Transfer Protocol (HTTP):** The request and response protocol used for Web browsers to transfer files, text, and graphics over the Internet between clients and servers. The client, possibly a home or business user using a Web browser, would initiate a communication request and receive a response from a remote server machine hosting the Internet Web site, establishing a connection through TCP port 80. An HTTP or Web site address starts with the prefix `http://`.

- **File Transfer Protocol (FTP):** An application protocol in the TCP/IP stack for transmitting files between network nodes using port 21 for control and port 20 for data. The most common use of FTP is downloading files from the Internet. Thousands of public and private FTP sites exist, allowing restricted or anonymous access to unlimited amounts of data. FTP address syntax is similar to HTTP, using `ftp://` as its convention.
- **Telnet:** A client/server protocol using the reliable connection-oriented transport mechanisms of TCP. Telnet sets up nonencrypted data communication between client and server on TCP port 23 and allows local client machines to access remote hosts as if the user were physically sitting at the remote node. In Figure 1-3, you can see an example of Telnet communications. This method of access is vulnerable to data interception by malicious users and lacks secure authentication. *Secure Shell (SSH)* protocol has replaced Telnet in most remote-access environments and provides stronger security and authentication features.
- **Domain Name System (DNS):** An Internet naming protocol used for addressing and naming remote computer systems. The DNS lookup process translates commonly used, simple names into long, difficult-to-remember numeric IP addresses (which are stored on DNS servers). This feature prevents a Web user from having to remember a Web site's IP address, such as *192.168.176.56.* Instead, DNS allows a user to enter a DNS Web site address (such as *www.sitename.com)* and retrieves the IP address transparently. Check out Figure 1-4 for an explanation of the DNS lookup process.
- **Dynamic Host Configuration Protocol (DHCP):** A protocol used for dynamic IP addressing and network configuration of clients issued from a central server. This dramatically reduces administration overhead and allows automatic network configuration to DHCP clients by sending a broadcast message to the DHCP server, requesting issue of a network IP address, subnet mask, DNS server address, default gateway, and other needed data. In Figure 1-5, you can see an example of a DHCP client requesting an IP lease from a DHCP server.

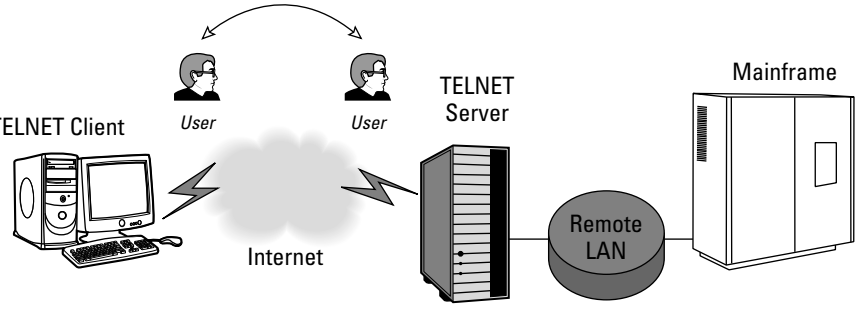


Figure 1-3: Telnet communications.

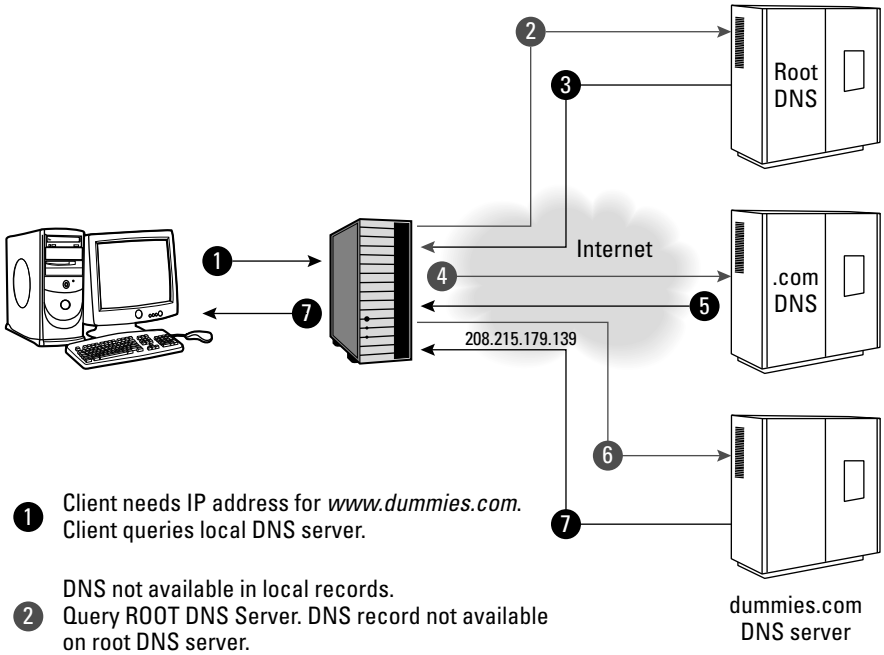


Figure 1-4: Domain Name System lookup.

Figure 1-5: Dynamic Host Configuration Protocol.

- **Internet Protocol Security (IPsec):** A security method that uses packet encryption and authentication techniques to secure IP communications. Internet Key Exchange (IKE), Authentication Headers (AHs), and Encapsulating Security Payload (ESP) handle negotiation, authentication, protection, and security duties and are some of the protocols used in this suite.
- **Address Resolution Protocol (ARP):** ARP is used to map a host machine's hardware Ethernet MAC address from the host's known IP address. The client sends a request to a remote host asking for resolution of a certain address, and the remote host identifies the required address and returns the query to the client. This is useful for identifying and communicating with Ethernet hosts on a local-area network (LAN).
- **Reverse Address Resolution Protocol (RARP):** A protocol that provides the reverse method of finding the IP address from a host's known hardware address.
- **Network Time Protocol (NTP):** A protocol that's used by computer systems over IP networks to time-synchronize with each other using a reference time source.
- **Open Shortest Path First (OSPF):** A dynamic routing protocol used by routers and network devices to exchange routing information. OSPF exchanges routing information between network devices, with the goal of building a routing map of the entire network. OSPF detects changes in data routes automatically, recognizes link failures, and reroutes traffic to build a loop-free environment.
- **Network File System (NFS):** A file-sharing system that allows access to remote data as if it were located on the local host.
- **Internet Control Message Protocol (ICMP):** A protocol that provides error and statistic handling for connectivity verification, sending out messages when a datagram is unable to reach its destination. When a gateway or router is unable to forward a datagram, or when a gateway can deliver a datagram on a shorter route, an ICMP message is delivered to the client. This protocol is often used by system administrators who want to verify that routers are sending packets correctly and to the proper destination address. The ICMP `ping` command is used to test whether another host is available over an IP network. This is accomplished by sending an ICMP "echo request" packet to a target interface and waiting for a reply.
- **Post Office Protocol 3 (POP3):** A standard protocol for e-mail retrieval from a remote server working on TCP port 110. An e-mail server stores electronic mail messages that can be retrieved all at once by remote systems using e-mail client software. A typical connection lasts long enough to download a user's e-mail and then disconnects while the retrieved e-mail is deleted from the server.

- **Internet Message Access Protocol (IMAP):** Another protocol used for e-mail retrieval on port 143 by remote clients that support both offline and online modes of operation. Unlike POP3, IMAP provides both connected and disconnect modes of operation and allows simultaneous, multiple user access to the same mailbox.
- **Simple Mail Transfer Protocol (SMTP):** A protocol for sending messages reliably and efficiently. (POP3 and IMAP are usually considered e-mail *receiving* protocols, while SMTP provides e-mail services typically used for *sending* messages.) Small text commands are used for negotiation and transmission control over a TCP data stream connection.

Introducing the major TCP/IP layers and protocols

Computers and networks also follow a structured model that describes how data flows from one source to another. The Open Systems Interconnection (OSI) reference model describes a set of rules for digital communication between hardware and software.

The OSI reference model

The OSI reference model, shown in Figure 1-6, was designed by the International Organization for Standardization (ISO) and defines how data is moved between a source and destination computer's software applications over a network. Delivering the data using seven conceptual layers defined by the ISO, these layers divide network communications architecture in a top-to-bottom approach. Moving up the OSI model from the bottommost layer to the top, services are *provided* to the next-uppermost layer (by the layer just below it), while services are *received* from the topmost layer to each next-lower layer. Each layer is responsible for a specific, exclusive set of functions not handled at any other layer.

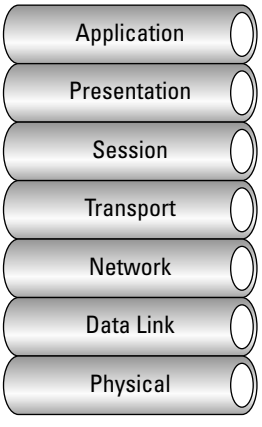


Figure 1-6: The OSI reference model.

The main purposes of this model are to allow compatibility among various computer manufacturers' network hardware and to provide a structured method for designing network protocols. Equipment from different vendors on the same network would communicate seamlessly together using the OSI model as a reference. Network protocols designed to follow this layered architecture offer a real advantage to networking equipment manufacturers. Each manufacturer can design its equipment using this model as a blueprint, guaranteeing compatibility and functionality.

Digital information sent from one machine to another using this model flows from the top of the OSI model (where the user interacts with the computer using software), starting at the application layer of the sending computer and traveling down this theoretical stack into the presentation, session, transport, network, data link, and physical layers.

Data arrives at the physical layer and is transported by a physical means (such as the LAN cabling and network adapter) to the receiver's computer. Then the data travels back up the invisible stack to arrive at the destination computer's application program (possibly a Web browser making an HTTP request). See Figure 1-7.

You may be wondering how the OSI layers communicate with each other. Each layer has a defined set of responsibilities and is independent of the layers above and below it. Communication is possible with layers above and below a given layer on the same system and its peer layer on the other side of the connection. The network layer may prepare and hand data off to either the transport or data link layer, depending on the direction of network traffic. If data is being received, it flows up the stack. Data that is being sent travels down the stack. The network layer on the sending computer also communicates with the network layer on the receiving computer, its peer layer. See Figure 1-8.

These layers are not actually moving data through the network and should not be confused with protocols used for communication. Network protocols are tasked with moving the data through a network and follow the rules put in place by the OSI model. Some protocols operate at different layers in the stack than other protocols, depending on function, and implement the layers' functionality.

The seven layers consist of upper and lower layers and are shown from the highest layer to the lowest (refer back to Figure 1-6):

- Layer 7: Application layer
- Layer 6: Presentation layer
- Layer 5: Session layer

- Layer 4: Transport layer
- Layer 3: Network layer
- Layer 2: Data link layer
- Layer 1: Physical layer

Commit the OSI model to memory. A good way to remember the names of each layer is to use the mnemonic device "**A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing" (from the top down), or in reverse order, "**P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way."

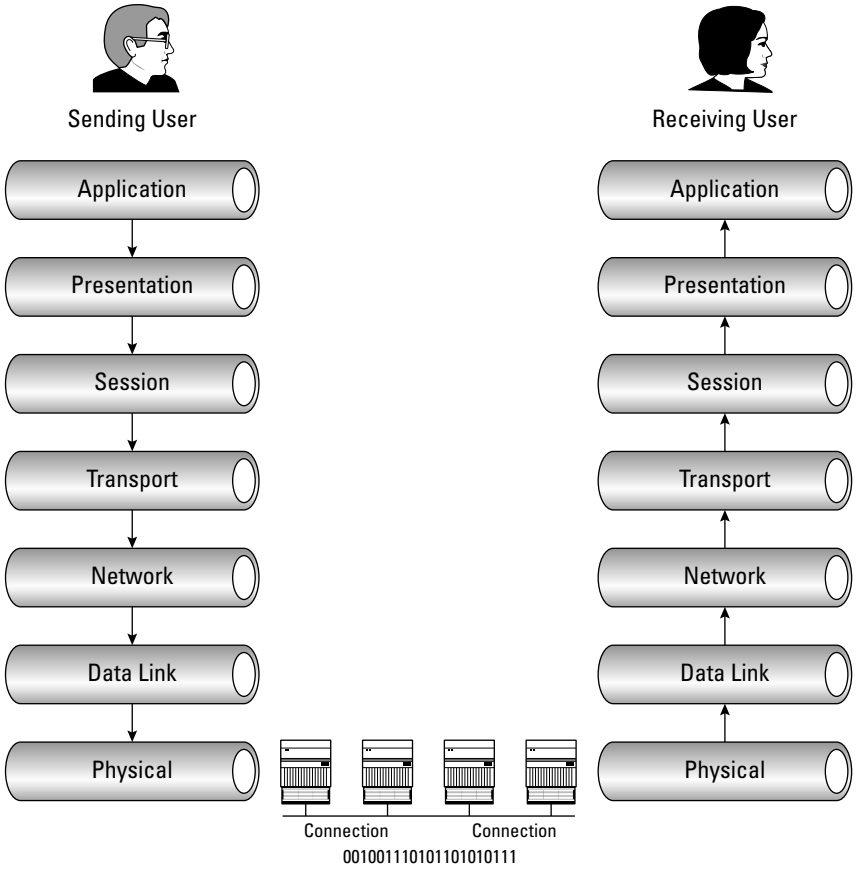


Figure 1-7: Data flow in the OSI reference model.

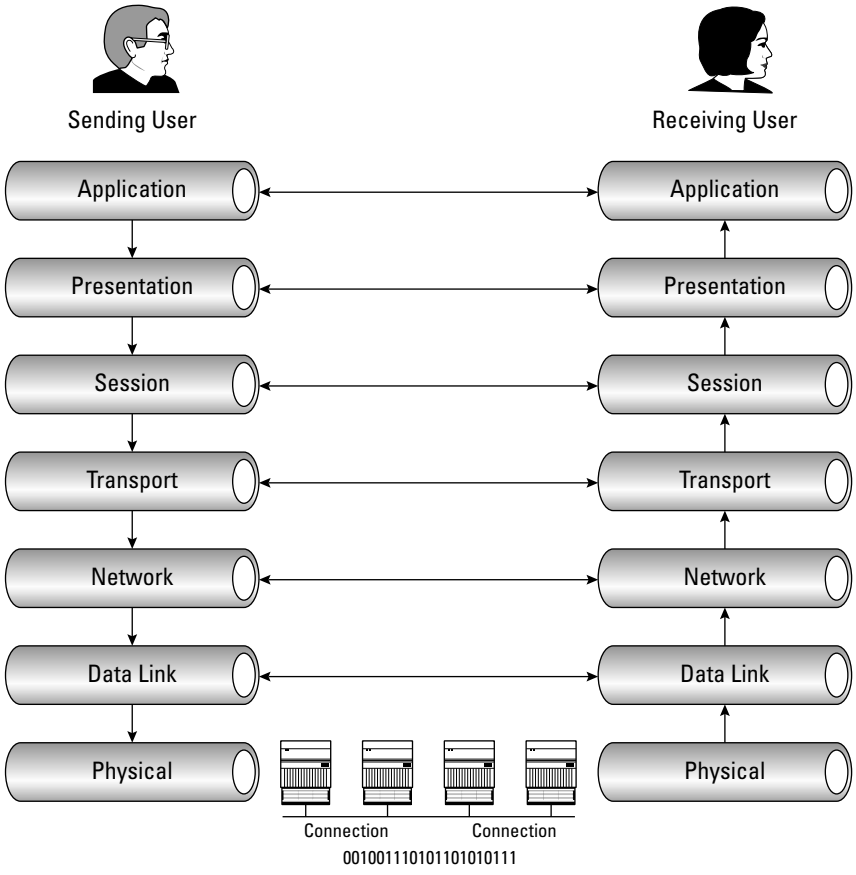


Figure 1-8: Data flow to OSI peer layers.

The seven OSI layers

The upper and lower layers have different responsibilities and are split accordingly. The *upper* layers, shown in Figure 1-9, include the *application, presentation,* and *session* layers and are tasked with software application communications:

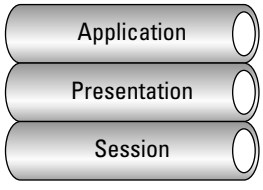


Figure 1-9: The OSI model upper layers.

- **Application layer:** The application layer is responsible for application-specific end-user processes and is the closest layer to the actual end user. Typical functions of the application layer include communication synchronization, resource availability determination, and identification of partners for communication. Some examples of application layer protocols include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Telnet.
- **Presentation layer:** The presentation layer, also referred to as the *syntax layer,* formats and encrypts data over the network. It also provides data to the application layer in a format it can understand. Translation is performed between application and network formats, ensuring that no compatibility issues exist.
- **Session layer:** The last of the upper layers is the session layer. It is responsible for controlling computer connections between local and remote applications. It initiates, controls, and terminates sessions using full-duplex, half-duplex, or simplex operations.

The *lower* layers, shown in Figure 1-10, provide data transportation services using both hardware and software methods for flow control, addressing, and routing over the network and make up the *transport, network, data link,* and *physical* layers:

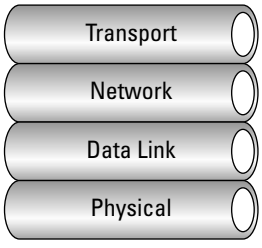


Figure 1-10: The OSI model lower layers.

- **Transport layer:** The transport layer provides seamless data transfer between computer systems using flow control and end-to-end error recovery, keeping track of datagram delivery and retransmitting lost packets. TCP and the connectionless UDP operate at this level.
- **Network layer:** The network layer is responsible for transporting data between networks and is also called the *routing* layer. All routed networks connected to the Internet function at this level using the Internet Protocol, and data is transferred in a series of hops. Internetworking, switching, routing, addressing, error handling, congestion control, and packet sequencing are handled here.

- **Data link layer:** The data link layer resides above and manages errors from the physical layer and is divided into two sublayers:
  - The *Media Access Control (MAC) layer,* useful for controlling network access
  - The *Logical Link Control (LLC) layer,* responsible for frame synchronization, flow control, and error checking

  The data link layer also encodes and decodes data packets into bits and manages errors from the physical layer, promoting reliable delivery on the physical medium.
- **Physical layer:** The physical layer resides at the bottom of the stack and defines the physical hardware used on the network, such as network interface cards, host bus adapters, repeaters, hubs, and cabling.

For the test: WAN technology uses telephone companies (and other carriers) to transmit data over long geographic distances and functions at the two lowest OSI layers. Although also considered to be a network layer function, WAN operation is generally considered to function at the physical and data link layers.

TCP/IP in the DoD model

Although the protocol architecture of TCP/IP follows the guidelines of the seven-layer OSI approach, TCP/IP more closely resembles a four-layer approach of the *DoD model* architecture. The DoD model conceptually referred to here is shown using fewer layers, but basic theory remains the same. Instead of data traveling through the seven-layer OSI model, data passes down through *four basic layers,* with each layer responsible for handing off control to the next adjoining layer. In this and the following sections, I give you a closer look at each layer and describe how this process works.

The four layers are listed here from the highest layer to the lowest:

- Layer 4: Application layer
- Layer 3: Transport layer
- Layer 2: Internet layer
- Layer 1: Network access layer

Data in the DoD model is sent from the host machine and travels down the stack, starting at the application layer and passing through the transport, Internet, and network access layers. The data is then transmitted over the network medium and returned up the four layers to the receiving host.

The four DoD layers

As shown in Figure 1-11, the four major components of the DoD model are the application, transport, Internet, and network access layers. Each layer has a specific responsibility and is listed here from the highest layer to the lowest:

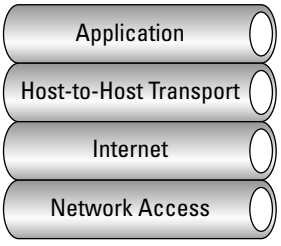


Figure 1-11: The DoD model.

- **Application layer:** The top layer in the DoD model roughly combines the functionality of the three upper layers described in the OSI model, namely, the application, presentation, and session layers. This is where user information is handled and packaged for delivery to the transport layer.
- **Transport layer:** The Transmission Control Protocol and User Datagram Protocol reside at the transport, or host-to-host, layer and provide valuable services in data delivery and error correction. TCP is the traffic cop and makes sure that the data gets to its intended destination, while UDP is unconcerned with verifying reliable delivery of packets.
- **Internet layer:** The Internet layer resides between the transport and network access layers and is the foundation of IP networking. The Internet Protocol is associated at this layer and is responsible for delivering packets across interconnected networks.
- **Network access layer:** Like the physical layer in the OSI model, the network access layer is responsible for delivering data across the physical network hardware.

Demystifying data encapsulation

Encapsulation in telecommunications is defined as the inclusion of one data structure inside another so that the first data structure is temporarily hidden from view. Data is encapsulated and decapsulated in this way as it travels through the different layers of the OSI and DoD models.

Starting from the application layer and moving downward, user information is formed into data and handed to the presentation layer for encapsulation. The presentation layer encapsulates the data provided by the application layer and passes it on to the session layer. The session layer synchronizes

with the corresponding session layer on the destination host and passes the data to the transport layer, which converts the data into *segments* and delivers these segments from source to destination. The network layer encapsulates the segments from the transport layer into packets, or datagrams, and gives a network *header* defining the source and destination IP addresses. These packets of data are given to the data link layer and converted into *frames.* Frames are then converted into *binary data,* ready for network transfer. This process is shown in Figure 1-12.

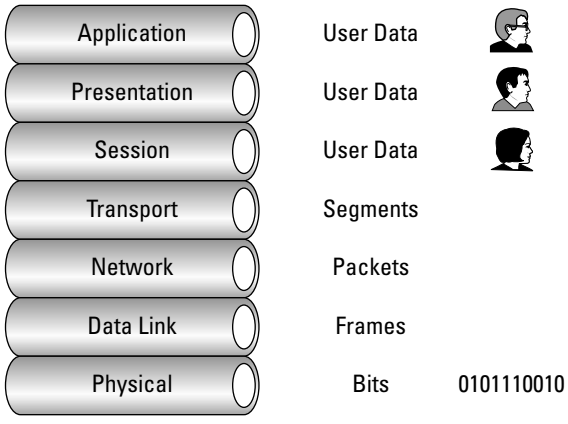


Figure 1-12: Data encapsulation.

Data encapsulation by OSI layer is described in the following table:

Layer	*Encapsulation*
Application	User data
Presentation	User data
Session	User data
Transport	Segments
Network	Packets or datagrams
Data link	Frames
Physical	Bits

User information goes through a five-step process during encapsulation to arrive at the physical wire:

1. User information is processed by the application, presentation, and session layers and prepares the data for transmission.

   For example, Robert opens his Web browser application on his laptop and types in the URL `http://www.cisco.com`.

2. The upper layers present the data to the transport layer, which converts the user data into segments.

   Continuing with the example, Robert's data request passes down from the upper layers to the transport layer and a header is added, acknowledging the HTTP request.

3. The network layer receives the segments and converts them into packets.

   The transport layer passes the data down to the network layer, where source and destination information is added, providing the address to the destination.

4. The data link layer converts the packets into frames.

   The data link layer frames the packets and adds the Ethernet hardware address of the source computer and the MAC address of the nearest connected device on the remote network.

5. The physical layer receives the data frames and converts them into binary format.

   Data frames are converted into bits and transmitted over the network, returning Robert's requested Web page.

Prep Test

1 **Which layer of the OSI model is responsible for reliable delivery of data across the physical network?**

A ❍ Network layer
B ❍ Data link layer
C ❍ Transport layer
D ❍ Physical layer

2 **Transmission Control Protocol operates at which OSI layer?**

A ❍ Transport layer
B ❍ Network layer
C ❍ Session layer
D ❍ Data link layer

3 **Which layer of the OSI model is responsible for managing sessions between applications?**

A ❍ Presentation layer
B ❍ Application layer
C ❍ Transport layer
D ❍ Session layer

4 **Which of the following are *not* steps in the data-encapsulation process? (Choose two.)**

A ❍ Segments are converted into frames
B ❍ User information is converted into data
C ❍ Frames are converted into bits
D ❍ Packets are converted into frames
E ❍ Data is converted into packets

5 **What does SMTP stand for?**

A ❍ Sending Mail Transfer Protocol
B ❍ Simple Mail Transfer Protocol
C ❍ Simple Method Timing Protocol
D ❍ Simple Management Transfer Protocol

6 **The User Datagram Protocol operates at which OSI layer?**

A ❍ Transport layer
B ❍ Network layer
C ❍ Session layer
D ❍ Data link layer

7 **TCP/IP is based on which type of technology?**

A ❍ Circuit-switching
B ❍ Packet-switching
C ❍ Frame-switching
D ❍ Header-switching

8 **Which layer of the OSI model theoretically resides closest to the end user?**

A ❍ Presentation layer
B ❍ Data link layer
C ❍ Application layer
D ❍ Physical layer

9 **TCP is considered to be what type of protocol?**

A ❍ Connectionless
B ❍ Proprietary
C ❍ Session-oriented
D ❍ Connection-oriented

10 **Which layer in the DoD model is responsible for routing IP packets?**

A ❍ Physical layer
B ❍ Session layer
C ❍ Network layer
D ❍ Internet layer