

OSI Model

Objective 1.1: Compare and contrast the Open Systems Interconnection (OSI) model layers and encapsulation concepts.

The OSI reference model can serve as a roadmap to many of the other concepts covered on the Network+ exam, providing a common lexicon that enables people working with these processes to communicate more easily.

In this chapter, you'll learn everything you need to know about Network+ Objective 1.1, including the following topics:

- ▶ **OSI model**
- ▶ **Data encapsulation and decapsulation within the OSI model context**

OSI MODEL

The *Open Systems Interconnection (OSI) reference model* is an architectural diagram of network communications. The model defines a highly complex process by dividing it into its component elements, in this case seven discrete layers, as shown in Figure 1.1.

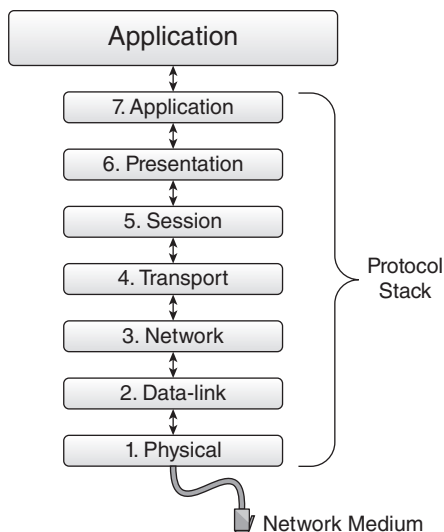


FIGURE 1.1 The seven layers of the OSI reference model

The seven layers of the model correspond to various network communications processes (called protocols) running on a computer. Table 1.1 lists the model layers, their functions, and the protocols that implement them. These protocols are covered in more detail throughout this book.

EXAM TIP

Memorizing this table is a good idea, as many Network+ exam questions reference the various layers of the OSI model, their functions, and their protocols. An easy way to remember the order of the layers is with the mnemonic “Please Do Not Throw Sausage Pizza Away.”

TABLE 1.1 OSI reference model layers

Number	Layer	Main functions	Protocols
7	Application	File, print, messaging, and other services	DNS, FTP, HTTP, POP3, SMTP, SNMP, DHCP
6	Presentation	Syntax translation	
5	Session	Dialog control, dialog separation	

Number	Layer	Main functions	Protocols
4	Transport	Data segmentation, packet acknowledgment, error detection and correction, flow control, port identification	TCP, UDP
3	Network	Addressing, routing, error detection, protocol identification	IP, IPv4, IPv6, ICMP
2	Data Link	Framing, media access control, protocol identification	IEEE 802.3 (Ethernet), IEEE 802.11 (Wi-Fi)
1	Physical	Network interface hardware, binary signaling	Twisted pair, fiber-optic, Wi-Fi

EXAM TIP

The layers of the OSI model are numbered from the bottom of the stack to the top. The physical layer at the bottom is designated as layer 1 and the application layer at the top as layer 7. In some reference works and product documentation, the model layers are referenced by number instead of name.

A *protocol* is a language that computers use to communicate with each other. Collectively, the processes running on the seven layers of the model are called the *protocol stack*. The network medium (whether a copper cable, fiber-optic cable, or wireless) connects to the physical layer at the bottom of the stack, and the signals entering at this layer travel up through the layers of the stack to the top. In the same way, data being sent over the network by software running at the top of the stack travel down through the layers and out to the network using the connected medium.

This vertical communication between the layers of the protocol stack takes the form of services that each layer provides for the layer above and requests from the layer below. For example, when an email client application sends a message, it typically generates a request for the services of the Simple Mail Transfer Protocol (SMTP), running at the application layer. SMTP then passes the request down to the next lower layer, which hands it off to the next layer, and so forth, until it reaches the bottom layer and is transmitted over the network.

The server receiving the transmission over the network then begins the same process in reverse, passing the message up through the layers of its own protocol stack until it reaches a mail server application running on the computer. The protocols running at specific layers of the OSI model have complementary functions that make network communication possible. The protocols at the individual layers of the model can therefore be said to communicate virtually with their counterparts on other systems.

One of the objectives of the OSI model was to define a networking architecture that enables manufacturers to create products that communicate readily with those of other manufacturers. As long as two computers are running compatible protocols at each layer of the OSI model, communication between them is theoretically possible, even if they are using different hardware and software.

While there have been other protocols over the years, LANs today nearly always run Ethernet or Wi-Fi at the physical and data link layers, *Internet Protocol (IP)* at the network layer, *Transmission Control Protocol (TCP)* and *User Datagram Protocol (UDP)* at the transport layer, and a collection of *Transmission Control Protocol/Internet Protocol (TCP/IP)* protocols and services at the application layer.

This TCP/IP protocol suite has a layered architecture similar to that of the OSI model, but it has only four layers, as shown in Table 1.2, and the layers do not correspond exactly to those of the OSI model. Some documents, such as the Requests for Comments (RFCs) that define the various TCP/IP protocols, reference the layers of the TCP/IP model instead of the OSI model layers.

TABLE 1.2 Corresponding OSI model and TCP/IP layers

OSI model layers	TCP/IP layers
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Link
Physical	

The seven layers of the OSI reference model are described in more detail in the following sections.

Layer 1: The Physical Layer

The physical layer, as the name implies, provides the actual physical connection between the computer and the network. The hardware implementing the physical layer in a computer—called a *network interface adapter*—typically includes either a connector for a network cable or a transceiver that provides wireless connectivity. In the network

communications architecture, the physical layer specification is responsible for providing the following elements:

Network Medium The technology that carries signals from one computer to another, such as copper-based cable, fiber-optic cable, or radio transmissions

Network Interface The connection between the computer and the network medium, such as a cable connector or a radio transceiver

Network Topology The arrangement of the network medium in the work site, such as the star topology typically used by cabled networks today

Network Installation The guidelines for the installation of the network medium, such as radio frequencies, maximum cable lengths, number of devices permitted, and proximity to other equipment

Network Signaling Specifies the nature of the signals: electrical, optical, or radio, which the devices use to transmit binary data over the network medium.

Computers communicate at the physical layer by generating signals and transmitting them over a network medium. For outgoing communications, the physical layer receives data from a protocol running on the layer above it (the data link layer), converts the data into a binary code appropriate for the medium, and transmits it over the network. In the same way, the physical layer receives incoming signals from the network and converts them into an appropriate form for the data link layer protocol.

These elements—the network medium, interface, topology, installation, and signaling scheme—are all typically defined by a single protocol specification. For example, a typical Ethernet physical layer specification might call for unshielded twisted pair (UTP) cable, installed in a star topology, with a maximum segment length of 100 meters, using RJ-45 connectors and an encoding scheme called Differential Manchester. Other Ethernet specifications call for fiber-optic or other network media, each with its own interface, topology, installation, and signaling characteristics. For more information on physical layer hardware, see Chapter 2, “Network Topologies and Types,” and Chapter 3, “Cables and Connectors.”

EXAM TIP

When preparing for the Network+ exam, it can be easy to get lost in details that are unlikely to appear in exam questions. It's only important to know the basic functions associated with each layer of the model and how those functions fit into the network communications process.

Layer 2: The Data Link Layer

The data link layer, layer 2 of the OSI model, is closely associated with the physical layer. The two together implement the computer's connection to and communication with the local area network (LAN) to which the device is attached. Selecting a protocol for the data

link layer also dictates the network medium options that are available at the physical layer. For more information on Ethernet variants and their supported cable types, see Chapter 3.

The functions that data link layer protocols typically perform include the following:

Frame Format The data link layer protocol packages the data passed down to it from the network layer in a protocol data unit called a frame. A *frame* consists of a header and a footer generated by the data link layer protocol, with the network layer data as the payload in between.

Addressing Every network interface adapter has a unique 6-byte identifier assigned to it by the manufacturer called a *media access control (MAC) address*. Data link layer frames use these addresses in their headers to identify the sender and recipient of each packet.

Protocol Identification Data link layer headers include a code that indicates which network layer protocol generated the data in each packet. This enables the data link layer protocol on the receiving system to pass incoming traffic up to the correct network layer process.

Error Detection The frame footer contains the result of a cyclical redundancy check (CRC) calculation performed on the frame's data payload by the sending system. The receiving system performs the same calculation and, if the results do not match, discards the packet.

Media Access Control (MAC) When a LAN has devices connected to a shared network medium, the data link layer protocol uses a media access control mechanism to prevent two devices from transmitting simultaneously and causing a data collision.

Physical Layer Specifications The data link layer protocol also includes specifications for the physical layer options associated with the protocol.

Most cabled LANs run the Ethernet protocol at the data link layer, the actual name for which is IEEE 802.3, after the standards published by the Institute of Electrical and Electronics Engineers. The functions just listed are all implemented as part of the Ethernet frame.

Layer 3: The Network Layer

The protocols running at the physical and data link layers of the OSI model are dedicated to local network communications only. For example, Ethernet is capable of transmitting frames to another computer or router on the LAN; it is not concerned with the delivery of the data to its final destination. That is the responsibility of the network layer protocol.

Network layer devices called routers connect individual networks together, forming an *internetwork*, or network of networks. A *router* is an intermediate system that only processes incoming packets as high as the network layer, as shown in Figure 1.2.

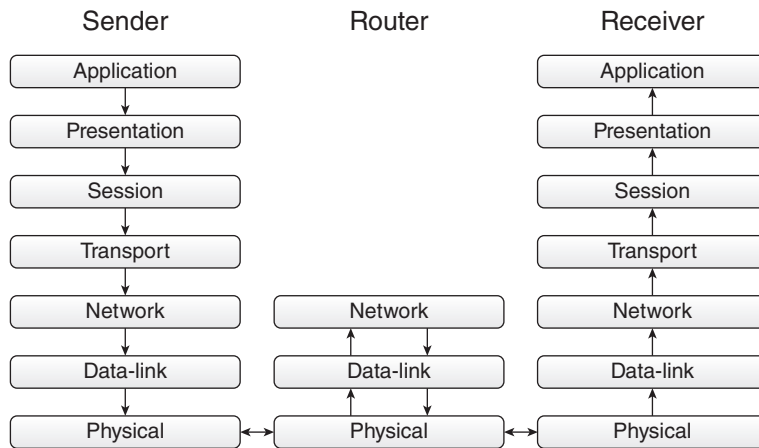


FIGURE 1.2 Network traffic processed by an intermediate router

The functions of a network layer protocol are as follows:

End-to-End Addressing The network layer protocol header contains the addresses of both the sending system and its final destination. Data link layer MAC addresses change for each leg of an internetwork journey, but network layer addresses always reflect a packet's starting point and its final destination.

Routing Routers are intermediate network layer devices that forward packets to other networks. Network layer protocols use routing tables to look up the addresses of other networks, so the router can forward packets to the correct destinations.

Fragmentation Networks can have different frame size limits, based on their *maximum transmission unit (MTU)* values, so if a single frame is too large to transmit over another network, the network layer protocol splits it into smaller fragments and transmits each fragment in a separate frame. The network layer protocol reassembles the fragments when they reach their final destination.

Protocol Identification Just as a data link layer protocol header contains a code to identify the network layer protocol that generated the data in the packet, the network layer protocol contains a code identifying the transport layer protocol that generated the data.

On TCP/IP networks, the primary protocol at the network layer is the *Internet Protocol (IP)*. IP exists in two versions: *IPv4*, with a 32-bit address space, which is the standard Internet protocol at the network layer, and *IPv6*, which is a newer version with a 128-bit address space. IPv6 has not yet been widely adopted for public Internet communications. For more information on the Internet Protocol, see Chapter 4, "IP Addressing."

IP functions are implemented in the header, which the protocol applies to the data it receives from the transport layer. This forms a unit called a *datagram*.

Layer 4: The Transport Layer

The term *TCP/IP* refers to the two protocols operating at the transport and network layers. The protocols at these two layers function together to provide a unified quality of service. IP, as noted earlier, provides addressing and routing services. The protocols at the transport layer of the OSI model provide additional end-to-end communication services, such as guaranteed delivery and flow control, to the adjacent layers, as needed.

Connection-Oriented and Connectionless Protocols

The OSI model standard recognizes two basic types of end-to-end protocol at the transport layer:

Connection-Oriented A connection-oriented protocol always establishes a connection between the sending and receiving systems before it transmits any application data. The systems do this by exchanging messages (called a *handshake*) containing *TCP flags* that confirm that both systems are available to send and receive data. After the transmission, another handshake terminates the connection. These additional messages (and a connection-oriented protocol's larger header) cause a significant increase in bandwidth overhead for this type of protocol. On most networks, the primary connection-oriented protocol at the transport layer is the *Transmission Control Protocol (TCP)*.

Connectionless A connectionless protocol provides no additional services other than the basic transmission of data, so there is no need for handshake messages and the header can be much smaller. This reduces the protocol's bandwidth overhead significantly. On most networks the primary connectionless protocol at the transport layer is the *User Datagram Protocol (UDP)*. The Internet Protocol (IP) running at the network layer is also connectionless.

Transport Layer Functions

The transport layer provides a variety of services for data transmissions. Most of the functions in the following list are provided by connection-oriented protocols such as TCP only. Only the protocol identification function is required for all protocols at the transport layer.

Data Segmentation A connection-oriented protocol accepts data from upper-layer application processes and splits it into segments of appropriate size for the network. Each segment is numbered so that they can be reassembled by the receiving system, even if they arrive out of order.

Packet Acknowledgment Connection-oriented protocols provide a *guaranteed delivery* service. The receiving system generates messages that acknowledge the successful receipt of packets from the sender.

Flow Control A connection-oriented mechanism to regulate the speed at which a sending system transmits data. The responses returned to the sender by the receiving system specify the size of a *sliding window*—that is, a buffer that holds incoming data. When the sliding window gets smaller, the sending system reduces its transmission speed to avoid overwhelming the receiver.

Signaled Error Correction A *signaled error* is a packet that has been discarded by the network layer protocol due to a CRC failure. The network layer protocol cannot correct errors, so it signals the transport layer protocol to retransmit the lost packets.

Unsigned Error Detection and Correction The transport layer provides the only end-to-end error detection and correction mechanism for the entire packet. The transport layer protocol on the sending system performs a CRC calculation and includes the results in its header. The receiving system then performs the same calculation and compares the results. For any lost packets, the receiver alters the packet acknowledgment value in its header, causing the sender to retransmit the unacknowledged packets.

Port Identification Transport layer protocol headers always contain a code (called a *port*) identifying the application layer protocol that generated or will receive the packet's data.

The transport layer functions are implemented in headers applied by each protocol. Because it is a connection-oriented protocol with many functions, TCP has a large 20-byte header. UDP is connectionless and therefore needs only a comparatively small 8-byte header.

Layer 5: The Session Layer

The boundary between the transport layer and the session layer marks an important change in the functioning of the protocol stack. All of the communication functions, both end-to-end and local, that are needed to transmit data from one system to another over a TCP/IP network are handled by the bottom four layers of the OSI model. The session, presentation, and application layers assume that the communication services are functional and that any messages they need to send or receive will be handled correctly.

The top three layers of the OSI model do not have separate protocols to implement their functions in the TCP/IP suite. Instead, the session layer functions as something of a toolbox; the OSI model standard lists 22 functions performed at the session layer, but on a TCP/IP network, those functions are usually integrated into an application layer protocol. It is for this reason that the session, presentation, and application layers are often referenced collectively as the *upper layers*.

The functions associated with the session layer are mostly concerned with the establishment, maintenance, and termination of communications during a connection between two end systems, called a *session*. The maintenance of a session can be a complicated business requiring various session layer functions to initiate the dialog, maintain orderly communication, and then terminate it.

Two of the important functions used to manage a session include the following:

- ▶ **Dialog control:** Specifies the mode in which the systems communicate. *Simplex* mode is when only one of the computers is transmitting. In *half duplex*, also called *two-way alternate (TWA)* mode, the systems exchange a token, and only the computer with the token can transmit. In *full duplex*, also called *two-way simultaneous (TWS)* mode, there is no token, and the computers can transmit any time.
- ▶ **Dialog separation:** Calls for the creation of checkpoints in the data stream to separate functions.

Layer 6: The Presentation Layer

As noted earlier, each OSI model layer provides services to the layer above and requests services from the layer below. In the case of the presentation layer, its primary function is to provide pass-through services that enable application layer protocols to request session layer services. For each of the services at the session layer, there is a corresponding pass-through service at the presentation layer. This enables the upper layers to function as a single entity.

The presentation layer also includes a syntax translation service that enables two end systems to communicate, despite their use of data compression, encryption, or different bit-encoding algorithms.

Layer 7: The Application Layer

The application layer is the entrance point to the protocol stack for applications running on the computer. In some cases, the application is separate from the application layer protocol, but in others, the application and the application layer protocol are the same.

There are hundreds of application layer protocols, many more than at any of the other layers. This is because there are so many applications that require a variety of highly specific networking services.

DATA ENCAPSULATION

As outgoing data travels down through the protocol stack to the network medium, it undergoes a process called *data encapsulation*, which is the functional equivalent of putting a letter into an envelope for mailing. When an application produces data that needs to be transmitted over the network, it generates a request for one of the application layer protocols. This protocol generates a message, properly formatted for the application, and passes it down through the layers of the stack. The protocols at the lower layers then add their own data, such as addresses and codes, to implement their functions.

For example, when the application layer data reaches the transport layer, the protocol there generates a *Transmission Control Protocol (TCP)* or *User Datagram Protocol (UDP)* header and adds it to the data. The TCP header includes *TCP flags* that the protocol uses to establish a connection between the sending and receiving systems and then break down the connection after the data transmission. The application layer request then becomes the *payload* in the transport layer segment or datagram, as shown in Figure 1.3.

The transport layer protocol then passes the data down to the network layer, which adds its own *Internet Protocol (IP)* header. The *maximum transmission unit (MTU)* value for the interface specifies the size of the largest IP datagram that can be transmitted over the network. Datagrams that are larger than the MTU are split into fragments for transmission. Finally, at the data link layer, all of the data from the layers above becomes the payload in a data link layer frame. The data link layer protocol typically adds both an *Ethernet header* and a footer to the payload, resulting in a packet that is ready for transmission.

For packets arriving at the destination system, the process occurs in reverse, with the data passed up through the OSI model layers, a process called *decapsulation*. Each layer processes the incoming packet by reading and stripping off the header and then passing the payload up to the correct protocol at the next higher layer.

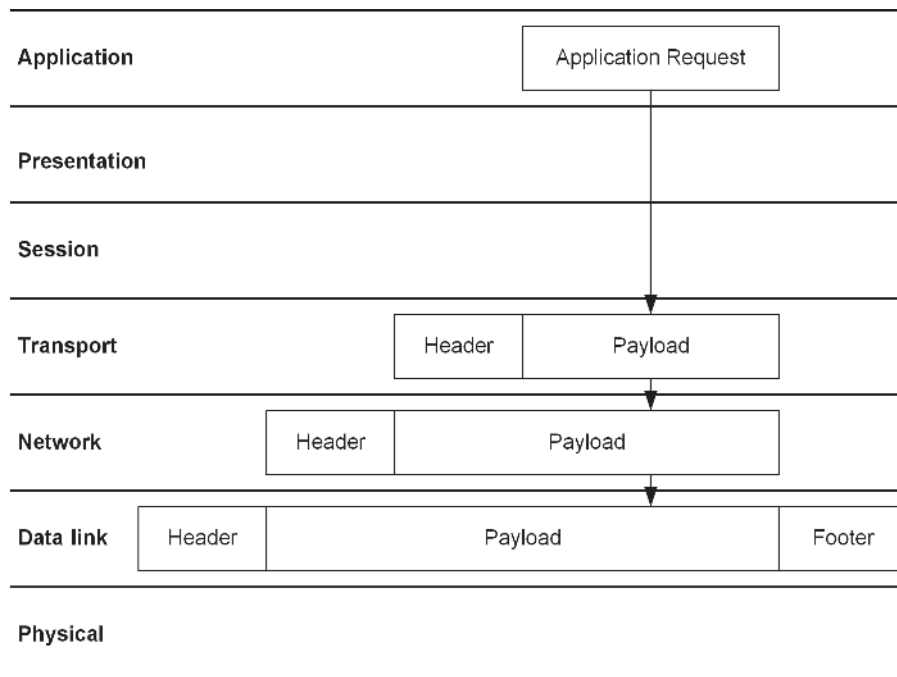


FIGURE 1.3 The data encapsulation process

CERTMIKE EXAM ESSENTIALS

- ▶ The OSI reference model divides the network communications process into seven layers. In order from the first layer to the seventh layer, these are the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer, and the application layer. You can remember them in order using the mnemonic “Please Do Not Throw Sausage Pizza Away.”
- ▶ Data encapsulation is the process by which the data generated by an application is packaged for transmission over the network through the application of headers by the protocols at the transport, network, and data link layers. When packets arrive at their destination, a decapsulation process occurs in which the protocols at each layer strip off their respective headers and pass the payload data up to the next layer.

Practice Question 1

There are several OSI model layers that provide error detection services, but only one that provides error correction as well. At which layer of the OSI model is there a protocol that provides both end-to-end error detection and correction?

- A. Physical
 - B. Data link
 - C. Network
 - D. Transport
-

Practice Question 2

Which of the following OSI model layers specifies the address of each packet's sender and its ultimate recipient?

- A. Data link
 - B. Network
 - C. Transport
 - D. Session
-

Practice Question 1 Explanation

- A. is incorrect because the physical layer provides the connection to the network and implements the signaling algorithm that converts the packet data into the electrical, optical, or radio signals necessary for transmission. This layer does not manipulate or evaluate the data in any way, so it performs no error correction.
- B. is incorrect because the data link layer is concerned only with local network communication. The data link layer performs a CRC calculation on each frame and includes the results in a frame check sequence field, but it cannot perform end-to-end error detection and has no error correction capability.
- C. is incorrect because as a connectionless protocol, it has no packet acknowledgment capability and therefore no means of error correction. The network layer performs error detection by including a CRC code in its header. If a packet fails the CRC test, the network layer discards it and signals the loss to the transport layer.
- D. is correct because TCP, a connection-oriented transport layer protocol, performs its own error detection, but it also includes a packet acknowledgment service. The segments transmitted during a TCP connection are numbered, so the acknowledgment messages generated by the receiver can specify the number of the last packet received correctly. The sender processes these messages and automatically retransmits any unacknowledged packets.

Correct Answer: D, Transport

Practice Question 2 Explanation

- A. is incorrect because the data link layer is only concerned with local network communications. The frame's destination address does not identify the ultimate recipient of the packet, only the next recipient on the LAN.
- B. is correct because IP, at the network layer, is the protocol responsible for end-to-end transmissions between the packet's sender and its ultimate recipient. The IP header therefore contains IP addresses that identify the two end systems.
- C. is incorrect because the transport layer headers do not contain addresses for the end systems. The transport layer protocols, TCP and UDP, provide services that support the end-to-end network communication process, but they rely on the network layer for addressing.
- D. is incorrect because the session layer provides functions that manage connections between end systems once they are established, but it does not have a dedicated protocol and has no addressing functionality.

Correct Answer: B, Network
