

Protocol Data Units and Encapsulation

For application data to travel uncorrupted from one host to another, header (or control data), which contains control and addressing information, is added to the data as it moves down the layers. The process of adding control information as it passes through the layered model is called *encapsulation*. *Decapsulation* is the process of removing the extra information and sending only the original application data up to the destination application layer.

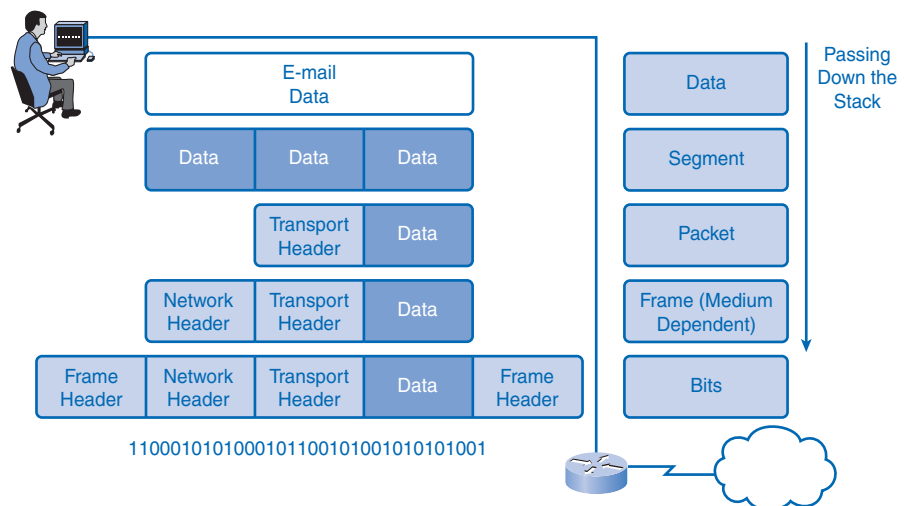
Each layer adds control information at each step. The generic term for data at each level is *protocol data unit (PDU)*, but a PDU is different at each layer. For example, a PDU at the internetwork layer is different from the PDU at the transport layer, because internetwork layer data has been added to the transport layer data. The different names for PDUs at each layer are listed in Table 2-4.

Table 2-4 Protocol Data Unit Naming Conventions

PDU Name	Layer
Data	Application layer PDU
<i>Segment</i>	Transport layer PDU
Packet	Internetwork layer PDU
<i>Frame</i>	Network access layer PDU
Bits	PDU used for the physical transmission of binary data over media

Figure 2-9 depicts the encapsulation process and shows how PDUs are modified.

Figure 2-9 Encapsulation



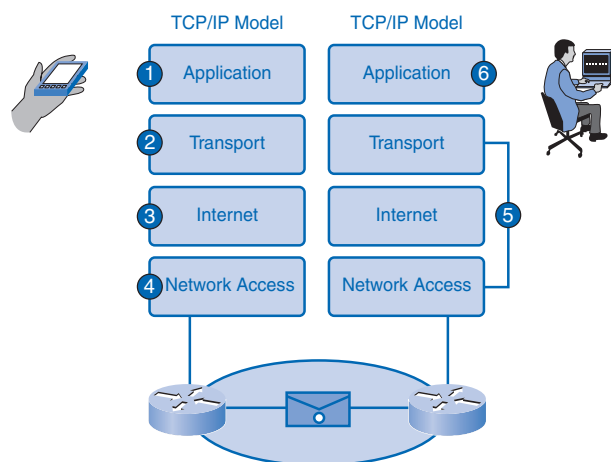
Sending and Receiving Process

The common task of sending an e-mail has many steps in the process. Using the proper terms for PDUs and the TCP/IP model, the process of sending the e-mail is as follows:

1. An end user, using an e-mail application, creates data. The application layer codes the data as e-mail and sends the data to the transport layer.
2. The message is segmented, or broken into pieces, for transport. The transport layer adds control information in a header so that it can be assigned to the correct process and all segments put into proper order at the destination. The segment is sent down to the internetwork layer.
3. The internetwork layer adds IP addressing information in an IP header. The segment is now an addressed packet that can be handled by routers en route to the destination. The internetwork layer sends the packet down to the network access layer.
4. The network access layer creates an Ethernet frame with local network physical address information in the header. This enables the packet to get to the local router and out to the web. The frame also contains a trailer with error-checking information. After the frame is created, it is encoded into bits and sent onto the media to the destination.
5. At the destination host, the process is reversed. The frame is decapsulated to a packet, then to a segment, and then the transport layer puts all segments into the proper order.
6. When all data has arrived and is ready, it is sent to the application layer, and then the original application data goes to the receiver's e-mail application. The message is successful.

Figure 2-10 depicts these steps as an encapsulated message travels down the TCP/IP model on the source and is en route to the destination for decapsulation.

Figure 2-10 Steps in the Communication Process



OSI Model

The *Open Systems Interconnection (OSI)* model, known as the OSI model, provides an abstract description of the network communication process. Developed by the *International Organization for Standardization (ISO)* to provide a road map for nonproprietary protocol development, the OSI model did not evolve as readily as the TCP/IP model. Many of the OSI protocols are no longer in use, but knowledge of the model as a reference is a basic expectation for networking professionals. Many professionals refer to the layers by number rather than name, so it is important to know both.

The OSI model is just a reference model, so manufacturers have been free to create protocols and products that combine functions of one or more layers. New protocols might not exactly match the functions described at each layer but might fit into parts of two different layers.

As designed, the communication process begins at the application layer of the source, and data is passed down to each lower layer to be encapsulated with supporting data until it reaches the physical layer and is put out on the media. When the data arrives at the destination, it is passed back up through layers and decapsulated by each layer. Each layer provides data services to the layer directly above by preparing information coming down the model or going up.

Table 2-5 briefly describes each layer of the OSI model. Each layer will be explored in its own chapter later in this book.

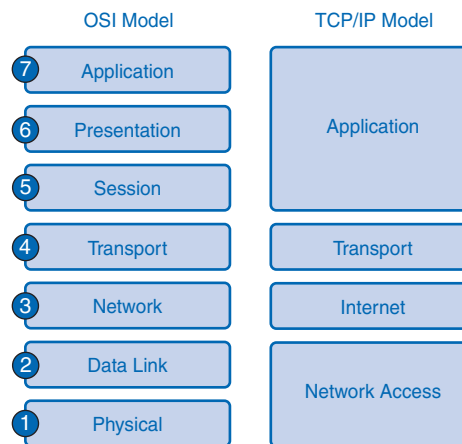
Table 2-5 OSI Model

No.	Layer Name	Description
7	Application	Performs services for the applications used by the end users.
6	Presentation	Provides data format information to the application. For example, the presentation layer tells the application layer whether there is encryption or whether it is a .jpg picture.
5	Session	Manages sessions between users. For example, the session layer will synchronize multiple web sessions and voice and video data in web conferences.
4	Transport	Defines data segments and numbers them at the source, transfers the data, and reassembles the data at the destination.
3	Network	Creates and addresses packets for end-to-end delivery through intermediary devices in other networks.
2	Data Link	Creates and addresses frames for host-to-host delivery on the local LANs and between WAN devices.
1	Physical	Transmits binary data over media between devices. Physical layer protocols define media specifications.

Comparing the OSI Model to the TCP/IP Model

The TCP/IP model evolved faster than the OSI model and is now more practical in describing network communication functions. The OSI model describes in detail functions that occur at the upper layers on the hosts, while networking is largely a function of the lower layers. Figure 2-11 shows the two models side by side for comparison.

Figure 2-11 Comparing the OSI and TCP/IP Models



When juxtaposed, you can see that the functions of the application, presentation, and session layers of the OSI model are combined into one application layer in the TCP/IP model.

The bulk of networking functions reside at the transport and the network layers, so they remain individual layers. TCP operates at the transport layer, and IP operates at the Internet layer.

The data link and physical layers of the OSI model combine to make the network access layer of the TCP/IP model.

Packet Tracer
Activity

Use of the TCP/IP Protocols and the OSI Model in Packet Tracer (2.4.8.2)

In this activity, you will see how Packet Tracer uses the OSI model as a reference to display the encapsulation details of a variety of the TCP/IP protocols. Use file e1-2482.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

Network Addressing

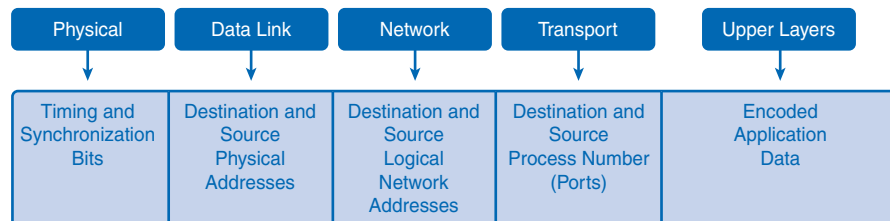
Successful communication requires that a sender and a receiver know how to get messages to each other. Postal systems use geography to deliver mail to physical addresses, but getting messages between computers is a more complicated matter. With the Internet, computers can communicate regardless of physical location.

Instead of using a geographical addressing scheme for computers, engineers devised a logical addressing scheme using numeric network addresses. The following sections introduce the addressing process. Chapter 6, “Addressing the Network: IPv4,” explores network addressing in greater detail.

Addressing in the Network

There are millions of computers in use on the web and billions of messages traversing networks at any given time, so proper addressing is essential to make sure that the sent message arrives intact at the proper destination. Addressing of data happens in three different layers of the OSI model. The PDU at each layer adds address information for use by the peer layer at the destination. Figure 2-12 depicts the different addressing information added by each layer.

Figure 2-12 Addressing Added at Each Layer



Getting Data to the End Device

During the process of encapsulation, address identifiers are added to the data as it travels down the protocol stack on the source host. There are two layers of addressing added to ensure that data is delivered to the destination.

The first identifier, the host physical address, is contained in the header of the Layer 2 PDU, called a frame. Layer 2 is concerned with the delivery of messages on a single local network. The Layer 2 address is unique on the local network and represents the address of the end device on the physical media. The physical address comes from codes placed on the NIC by the manufacturer. In a LAN using Ethernet, this address is called the MAC address. The terms *physical address* and *MAC address* are often used interchangeably. When two end devices communicate on the local Ethernet network, the frames that are exchanged

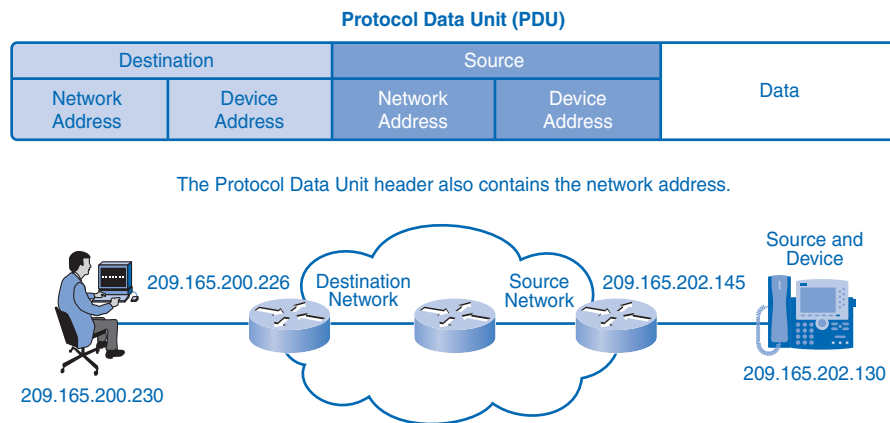
between them contain the destination and source MAC addresses. After a frame is successfully received by the destination host, the Layer 2 address information is removed as the data is decapsulated and moved up the protocol stack to Layer 3.

Getting Data Through the Internetwork

Layer 3 protocols are primarily designed to move data from one local network to another local network within an internetwork. Whereas Layer 2 addresses are only used to communicate between devices on a single local network, Layer 3 addresses must include identifiers that enable intermediary network devices to locate hosts on different networks. In the TCP/IP protocol suite, every IP host address contains information about the network where the host is located.

At the boundary of each local network, an intermediary network device, usually a router, decapsulates the frame to read the destination host address contained in the header of the packet, the Layer 3 PDU. Routers use the network identifier portion of this address to determine which path to use to reach the destination host. When the path is determined, the router encapsulates the packet in a new frame and sends it on its way toward the destination end device. When the frame reaches its final destination, the frame and packet headers are removed and the data moved up to Layer 4. The journey from source to destination is depicted in Figure 2-13.

Figure 2-13 IP Addressing



Getting Data to the Right Application

At Layer 4, information contained in the PDU header does not identify a destination host or a destination network. What it does identify is the specific process or service running on the destination host device that will act on the data being delivered. Hosts, whether they are clients or servers on the Internet, can run multiple network applications simultaneously. People using PCs often have an e-mail client running at the same time as a web browser, an instant messaging program, some streaming media, and perhaps even a game. All these separately running programs are examples of individual processes.

Viewing a web page invokes at least one network process. Clicking a hyperlink causes a web browser to communicate with a web server. At the same time, in the background, an e-mail client can be sending and receiving e-mail, and a colleague or friend can be sending an instant message.

Think about a computer that has only one network interface on it. All the data streams created by the applications that are running on the PC enter and leave through that one interface, yet instant messages do not pop up in the middle of a word processor document or e-mail showing up in a game.

This is because the transport layer adds *port* numbers to its segment header information to ensure that the destination host knows which application process is to receive the packet. The end host assigns a port number to each type of traffic going in and out. A user can send and receive many types of traffic over a single network interface, and using port numbers for each segment keeps traffic for web pages separate from e-mail traffic and so on. The segment contains both source and destination ports in case the receiver needs to contact the sender. Figure 2-14 shows different data types for two different services on an end device.

Figure 2-14 Port Addressing

