Packet Encapsulation and Lifecycle

In a *circuit-switched* network, a dedicated physical circuit path is established, maintained, and terminated between the sender and receiver across a network for each communications session. Before the development of the internet, most communications networks, such as telephone company networks, were circuit-switched. As discussed in Section 2.0.1, the internet is a *packet-switched* network comprising hundreds of millions of routers and billions of servers and user endpoints. In a packet-switched network, devices share bandwidth on communications links to transport packets between a sender and a receiver across a network. This type of network is more resilient to error and congestion than circuit-switched networks.

An application that needs to send data across the network (for example, from a server to a client computer) first creates a block of data and sends it to the TCP stack on the server. The TCP stack places the block of data into an output buffer on the server and determines the maximum segment size (MSS) of individual TCP blocks (*segments*) permitted by the server operating system. The TCP stack then divides the data blocks into appropriately sized segments (for example, 1460 bytes), adds a TCP header, and sends the segment to the IP stack on the server. The IP stack adds source (sender) and destination (receiver) IP addresses to the TCP segment (which is now called an IP packet) and notifies the server operating system that it has an outgoing message that is ready to be sent across the network. When the server operating system is ready, the IP packet is sent to the network adapter, which converts the IP packet to bits and sends the message across the network.

On their way to the destination computer, the packets typically traverse several network and security devices (such as switches, routers, and firewalls) before reaching the destination computer, where the encapsulation process described is reversed.

Key Terms

In a *circuit-switched network,* a dedicated physical circuit path is established, maintained, and terminated between the sender and the receiver across a network for each communications session.

In a *packet-switched network,* devices share bandwidth on communications links to transport packets between the sender and the receiver across a network.

A TCP *segment* is a *protocol data unit* (PDU) defined at the Transport layer of the OSI model.

A *protocol data unit* (PDU) is a self-contained unit of data (consisting of user data or control information and network addressing).

The OSI and TCP/IP models

The *Open Systems Interconnection* (OSI) and *Transmission Control Protocol/Internet Protocol* (TCP/IP) models define standard protocols for network communication and interoperability. Using a layered approach, the OSI and TCP/IP models:

- Clarify the general functions of communications processes
- Reduce complex networking processes to simpler sublayers and components
- Promote interoperability through standard interfaces
- Enable vendors to change individual features at a single layer rather than rebuild the entire protocol stack
- Facilitate logical troubleshooting

Defined by the International Organization for Standardization (ISO – not an acronym but the adopted organizational name from the Greek *isos*, meaning “equal”), the OSI model consists of seven layers:

- **Application (Layer 7 or L7).** This layer identifies and establishes availability of communication partners, determines resource availability, and synchronizes communication. Protocols that function at the Application layer include:
  - **File Transfer Protocol (FTP).** Used to copy files from one system to another on TCP ports 20 (the data port) and 21 (the control port)
  - **Hypertext Transfer Protocol (HTTP).** Used for communication between web servers and web browsers on TCP port 80
  - **Hypertext Transfer Protocol Secure (HTTPS).** Used for Secure Sockets Layer/Transport Layer Security (SSL/TLS) encrypted communications between web servers and web browsers on TCP port 443 (and other ports, such as 8443)
  - **Internet Message Access Protocol (IMAP).** A store-and-forward electronic mail protocol that allows an email client to access, manage, and synchronize email on a remote mail server on TCP and UDP port 143
  - **Post Office Protocol Version 3 (POP3).** An email retrieval protocol that allows an email client to access email on a remote mail server on TCP port 110
  - **Simple Mail Transfer Protocol (SMTP).** Used to send and receive email across the internet on TCP/UDP port 25
  - **Simple Network Management Protocol (SNMP).** Used to collect network information by polling stations and sending traps (or alerts) to a management station on TCP/UDP ports 161 (agent) and 162 (manager)
  - **Telnet.** Provides terminal emulation for remote access to system resources on TCP/UDP port 23

Presentation (Layer 6 or L6). This layer provides coding and conversion functions (such as data representation, character conversion, data compression, and data encryption) to ensure that data sent from the Application layer of one system is compatible with the Application layer of the receiving system. Protocols that function at the Presentation layer include:

- **American Standard Code for Information Interchange (ASCII).** A character-encoding scheme based on the English alphabet, consisting of 128 characters
- **Extended Binary-Coded Decimal Interchange Code (EBCDIC).** An 8-bit character-encoding scheme largely used on mainframe and midrange computers
- **Graphics Interchange Format (GIF).** A bitmap image format that allows up to 256 colors and is suitable for images or logos (but not photographs)
- **Joint Photographic Experts Group (JPEG).** A photographic compression method used to store and transmit photographs
- **Motion Picture Experts Group (MPEG).** An audio and video compression method used to store and transmit audio and video files

Session (Layer 5 or L5). This layer manages communication sessions (service requests and service responses) between networked systems, including connection establishment, data transfer, and connection release. Protocols that function at the Session layer include:

- **Network File System (NFS).** Facilitates transparent user access to remote resources on a Unix-based TCP/IP network
- **Remote Procedure Call (RPC).** A client-server network redirection protocol
- **Secure Shell (SSH).** Establishes an encrypted tunnel between a client and a server
- **Session Initiation Protocol (SIP).** An open signaling protocol standard for establishing, managing, and terminating real-time communications (such as voice, video, and text) over large IP-based networks

Transport (Layer 4 or L4). This layer provides transparent, reliable data transport and end-to-end transmission control. Specific Transport layer functions include:

- **Flow control.** Manages data transmission between devices by ensuring that the transmitting device doesn't send more data than the receiving device can process
- **Multiplexing.** Enables data from multiple applications to be simultaneously transmitted over a single physical link
- **Virtual circuit management.** Establishes, maintains, and terminates virtual circuits
- **Error checking and recovery.** Detects transmission errors and takes action to resolve any errors that occur, such as requesting that data be retransmitted

TCP and UDP port numbers assigned to applications and services are defined at the Transport layer. Protocols that function at the Transport layer include:

Transmission Control Protocol (TCP). A connection-oriented (a direct connection between network devices is established before data segments are transferred) protocol that provides reliable delivery (received segments are acknowledged, and retransmission of missing or corrupted segments is requested) of data. TCP connections are established via a *three-way handshake*. The additional overhead associated with connection establishment, acknowledgment, and error correction means that TCP is generally slower than connectionless protocols such as User Datagram Protocol (UDP).

User Datagram Protocol (UDP). A connectionless (a direct connection between network devices is not established before *datagrams* are transferred) protocol that provides best-effort delivery (received datagrams are not acknowledged and missing or corrupted datagrams are not requested) of data. UDP has no overhead associated with connection establishment, acknowledgment, sequencing, or error-checking and recovery. UDP is ideal for data that requires fast delivery, as long as that data isn't sensitive to packet loss and doesn't need to be fragmented. Applications that use UDP include Domain Name System (DNS), Simple Network Management Protocol (SNMP), and streaming audio or video.

Stream Control Transmission Protocol (SCTP). A message-oriented protocol (similar to UDP) that ensures reliable, in-sequence transport with congestion control (similar to TCP).

Network (Layer 3 or L3). This layer provides routing and related functions that enable data to be transported between systems on the same network or on interconnected networks. Routing protocols (discussed later) are defined at this layer. Logical addressing of devices on the network is accomplished at this layer using routed protocols such as Internet Protocol (IP). Routers operate at the Network layer of the OSI model.

Data Link (Layer 2). This layer ensures that messages are delivered to the proper device across a physical network link. This layer also defines the networking protocol (for example, Ethernet) used to send and receive data between individual devices and formats messages from the layers listed above into frames for transmission, handles point-to-point synchronization and error control, and can perform link encryption. Switches typically operate at Layer 2 of the OSI model (although multilayer switches that operate at different layers also exist). The Data Link layer is further divided into two sublayers:

Logical Link Control (LLC). The LLC sublayer provides an interface for the MAC sublayer; manages the control, sequencing, and acknowledgment of frames being passed up to the Network layer or down to the Physical layer; and manages timing and *flow control*.

Media access control (MAC). The MAC sublayer is responsible for framing and performs error control using a *cyclic redundancy check* (CRC), identifies MAC addresses (discussed later), and controls media access.

Physical (Layer 1 or L1). This layer sends and receives bits across the network medium (cabling or wireless links) from one device to another. It specifies the electrical, mechanical, and functional requirements of the network, including network topology, cabling and connectors, and interface types, as well as the process for converting bits to electrical (or light) signals that can be transmitted across the physical medium.

Key Terms

In TCP, a *three-way handshake* is used to establish a connection. For example, a PC initiates a connection with a server by sending a TCP SYN (Synchronize) packet. The server replies with a SYN ACK packet (Synchronize Acknowledgment). Finally, the PC sends an ACK or SYN-ACK-ACK packet acknowledging the server's acknowledgment, and data communication begins.

A UDP *datagram* is a PDU defined at the Transport layer of the OSI model.

Flow control monitors the flow of data between devices to ensure that a receiving device, which may not necessarily be operating at the same speed as the transmitting device, does not drop packets.

A *cyclic redundancy check* (CRC) is a checksum used to create a message profile. The CRC is recalculated by the receiving device. If the recalculated CRC doesn't match the received CRC, the packet is dropped, and a request to resend the packet is transmitted back to the device that sent the packet.

The TCP/IP model was originally developed by the U.S. Department of Defense (DoD) and actually preceded the OSI model. Whereas the OSI model is a theoretical model used to logically describe networking processes, the TCP/IP model defines actual networking requirements, for example, for frame construction. The TCP/IP model consists of four layers (see Figure 2-2):

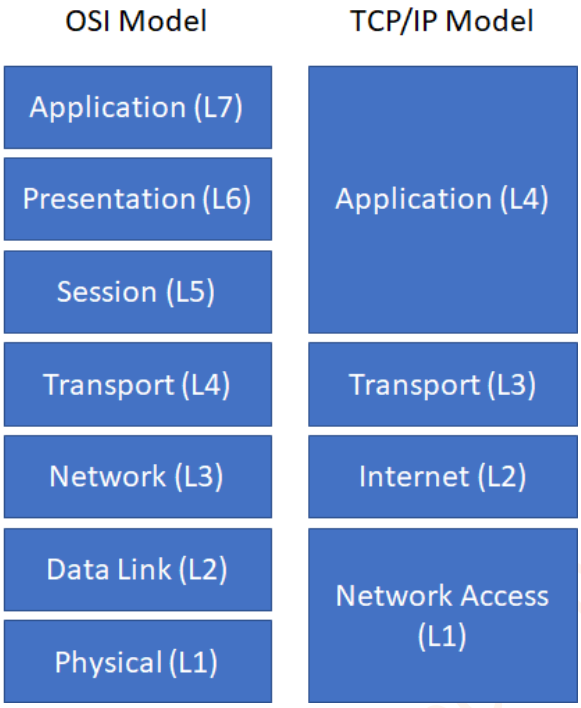
Application (Layer 4 or L4). This layer consists of network applications and processes, and it loosely corresponds to Layers 5 through 7 of the OSI model.

Transport (Layer 3 or L3). This layer provides end-to-end delivery, and it corresponds to Layer 4 of the OSI model.

Internet (Layer 2 or L2). This layer defines the IP datagram and routing, and it corresponds to Layer 3 of the OSI model.

Network Access (Layer 1 or L1). Also referred to as the Link layer, this layer contains routines for accessing physical networks, and it corresponds to Layers 1 and 2 of the OSI model.

Figure 2-2 *The OSI model and the TCP/IP model*



Data encapsulation

In the OSI and TCP/IP models, data is passed from the highest layer (L7 in the OSI model, L4 in the TCP/IP model) downward through each layer to the lowest layer (L1 in the OSI model and the TCP/IP model). It is then transmitted across the network medium to the destination node, where it is passed upward from the lowest layer to the highest layer. Each layer communicates only with the adjacent layer immediately above and below it. This communication is achieved through a process known as *data encapsulation* (or *data hiding*), which wraps protocol information from the layer immediately above in the data section of the layer immediately below.

A protocol data unit (PDU) describes a unit of data at a particular layer of a protocol. For example, in the OSI model, a Layer 1 PDU is known as a bit, a Layer 2 PDU is known as a frame, a Layer 3 PDU is known as a packet, and a Layer 4 PDU is known as a segment or datagram. When a client or server application sends data across a network, a header (and trailer in the case of Layer 2 frames) is added to each data packet from the adjacent layer below it as the data passes through the protocol stack. On the receiving end, the headers (and trailers) are removed from each data packet as it passes through the protocol stack to the receiving application.

Key Terms

Data encapsulation (or *data hiding*) wraps protocol information from the (OSI or TCP/IP) layer immediately above in the data section of the layer below.

Routed and routing protocols

Routed outed protocols, such as *Internet Protocol* (IP), address packets with routing information that enables those packets to be transported across networks using routing protocols.

Routing protocols are defined at the Network layer of the OSI model (discussed earlier) and specify how routers communicate with one another on a network. Routing protocols can either be static or dynamic.

> **Key Terms**
>
> An *Internet Protocol (IP) address* is a 32-bit or 128-bit identifier assigned to a networked device for communications at the Network layer of the OSI model or the Internet layer of the TCP/IP model.

A static routing protocol requires that routes be created and updated manually on a router or other network device. If a static route is down, traffic can't be automatically rerouted unless an alternate route has been configured. Also, if the route is congested, traffic can't be automatically rerouted over the less congested alternate route. Static routing is practical only in very small networks or for very limited, special-case routing scenarios (for example, a destination that's used as a backup route or is reachable only via a single router). However, static routing has low bandwidth requirements (routing information isn't broadcast across the network) and some built-in security (users can route only to destinations that are specified in statically defined routes).

A dynamic routing protocol can automatically learn new (or alternate) routes and determine the best route to a destination. The routing table is updated periodically with current routing information. Dynamic routing protocols are further classified as:

- **Distance-vector.** A distance-vector protocol makes routing decisions based on two factors: the distance (hop count or other metric) and vector (the egress router interface). It periodically informs its peers and/or neighbors of topology changes. *Convergence*, the time required for all routers in a network to update their routing tables with the most current information (such as link status changes), can be a significant problem for distance-vector protocols. Without convergence, some routers in a network may be unaware of topology changes, which causes the router to send traffic to an invalid destination. During convergence, routing information is exchanged between routers, and the network slows down considerably. Convergence can take several minutes in networks that use distance-vector protocols.

  Routing Information Protocol (RIP) is an example of a distance-vector routing protocol that uses *hop count* as its routing metric. To prevent routing loops, in which packets effectively get stuck bouncing between various router nodes, RIP implements a hop limit of 15, which limits the size of networks that RIP can support. After a data packet crosses 15 router nodes (hops) between a source and a destination, the destination is considered unreachable. In addition to hop limits, RIP employs four other mechanisms to prevent routing loops:

- **Split horizon.** Prevents a router from advertising a route back out through the same interface from which the route was learned.
- **Triggered updates.** When a change is detected, the update gets sent immediately instead of waiting 30 seconds to send a RIP update.
- **Route poisoning.** Sets the hop count on a bad route to 16, which effectively advertises the route as unreachable.
- **Holddown timers.** Causes a router to start a timer when the router first receives information that a destination is unreachable. Subsequent updates about that destination will not be accepted until the timer expires. This timer also helps avoid problems associated with flapping. Flapping occurs when a route (or interface) repeatedly changes state (up, down, up, down) over a short period of time.

- **Link state.** A link-state protocol requires every router to calculate and maintain a complete map, or routing table, of the entire network. Routers that use a link-state protocol periodically transmit updates that contain information about adjacent connections, or link states, to all other routers in the network. Link-state protocols are compute-intensive, but they can calculate the most efficient route to a destination. They consider numerous factors, such as link speed, delay, load, reliability, and *cost* (an arbitrarily assigned weight or metric). Convergence occurs very rapidly (within seconds) with link-state protocols.

  Open Shortest Path First (OSPF) is an example of a link-state routing protocol that is often used in large enterprise networks. OSPF routes network traffic within a single *autonomous system* (AS). OSPF networks are divided into areas identified by 32-bit area identifiers. Area identifiers can (but don't have to) correspond to network IP addresses and can duplicate IP addresses without conflicts.

- **Path vector.** A path-vector protocol is similar to a distance-vector protocol but without the scalability issues associated with limited hop counts in distance-vector protocols. Each routing table entry in a path-vector protocol contains path information that gets dynamically updated.

Border Gateway Protocol (BGP) is an example of a path-vector protocol used between separate autonomous systems. BGP is the core protocol used by internet service providers (ISPs) and network service providers (NSPs), as well as on very large private IP networks.

Key Terms

Convergence is the time required for all routers in a network to update their routing tables with the most current routing information about the network.

Hop count generally refers to the number of router nodes that a packet must pass through to reach its destination.

An *autonomous system* (AS) is a group of contiguous IP address ranges under the control of a single internet entity. Individual autonomous systems are assigned a 16-bit or 32-bit AS number (ASN) that uniquely identifies the network on the internet. ASNs are assigned by the Internet Assigned Numbers Authority (IANA).