CHAPTER 1

Introduction to Internetworking

Introduction

This chapter explains basic internetworking concepts. The information presented here helps readers who are new to internetworking comprehend the technical material that makes up the bulk of this publication. Sections on the Open System Interconnection (OSI) reference model, important terms and concepts, and key organizations are included.

OSI Reference Model: Introduction

Moving information between computers of diverse design is a formidable task. In the early 1980s, the International Organization for Standardization (ISO) recognized the need for a network model that would help vendors create interoperable network implementations. The OSI reference model, released in 1984, addresses this need.

The OSI reference model quickly became the primary architectural model for intercomputer communications. Although other architectural models (mostly proprietary) have been created, most network vendors relate their network products to the OSI reference model when they want to educate users about their products. Thus, the model is the best tool available to people hoping to learn about network technology.

Hierarchical Communication

The OSI reference model divides the problem of moving information between computers over a network medium into seven smaller and more manageable problems. Each of the seven smaller problems was chosen because it was reasonably self-contained and therefore more easily solved without excessive reliance on external information.

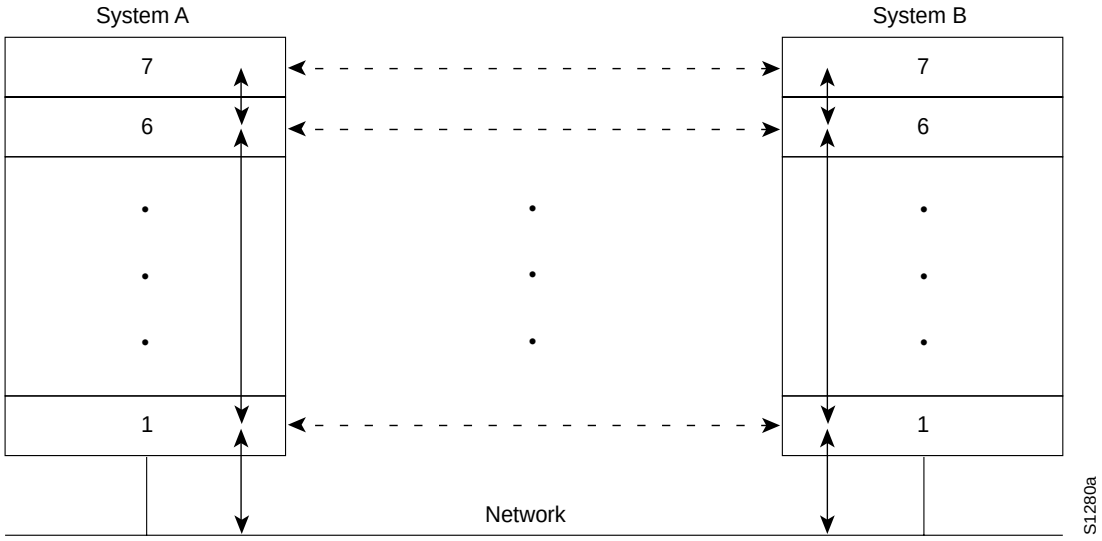
Each of the seven problem areas is solved by a *layer* of the model. Most network devices implement all seven layers. To streamline operations, however, some network implementations skip one or more layers. The lower two OSI layers are implemented with hardware and software; the upper five layers are generally implemented in software.

The OSI reference model describes how information makes its way from application programs (such as spreadsheets) through a network medium (such as wires) to another application program in another computer. As the information to be sent descends through the layers of a given system, it looks less and less like human language and more and more like the ones and zeros that a computer understands.

As an example of OSI-type communication, assume that System A in Figure 1-1 has information to send to System B. The application program in System A communicates with System A's Layer 7 (the top layer), which communicates with System A's

Layer 5, and so on until System A's Layer 1 is reached. Layer 1 is concerned with putting information on (and taking information off) the physical network medium. After the information has traversed the physical network medium and been absorbed into System B, it ascends through System B's layers in reverse order (first Layer 1, then Layer 2, and so on) until it finally reaches System B's application program.

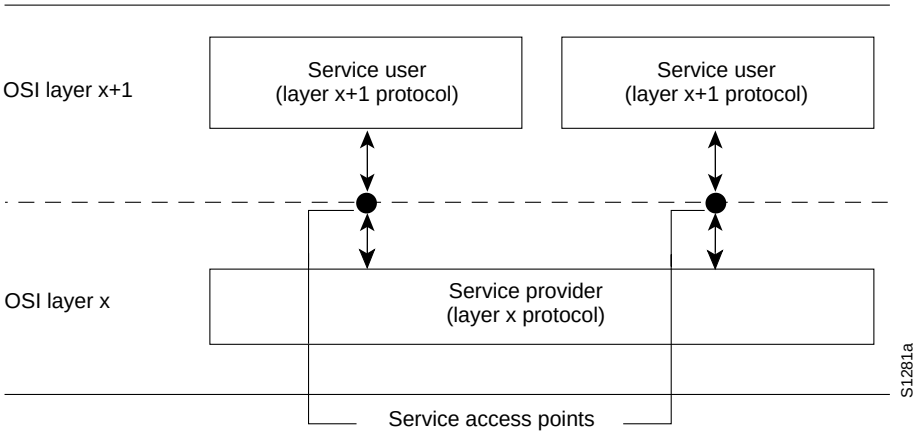
Figure 1-1 Communication between Two Computer Systems



Although each of System A's layers communicates with its adjacent System A layers, its primary objective is to communicate with its peer layer in System B. That is, the primary objective of Layer 1 in System A is to communicate with Layer 1 in System B; Layer 2 in System A communicates with Layer 2 in System B, and so on. This is necessary because each layer in a system has certain tasks it must perform. To perform these tasks, it must communicate with its peer layer in the other system.

The OSI model's layering precludes direct communication between peer layers in different systems. Each layer in System A must therefore rely on services provided by adjacent System A layers to help achieve communication with its System B peer. The relationship between adjacent layers in a single system is shown in Figure 1-2.

Figure 1-2 Relationship between Adjacent Layers in a Single System



Assume Layer 4 in System A must communicate with Layer 4 in System B. To do this, Layer 4 in System A must use the services of Layer 3 in System A. Layer 4 is said to be the *service user*, while Layer 3 is the *service provider*. Layer 3 services are provided to Layer 4 at a *service access point* (SAP), which is simply a location at which Layer 4 can request Layer 3 services. As the figure shows, Layer 3 can provide its services to multiple Layer 4 entities.

Information Formats

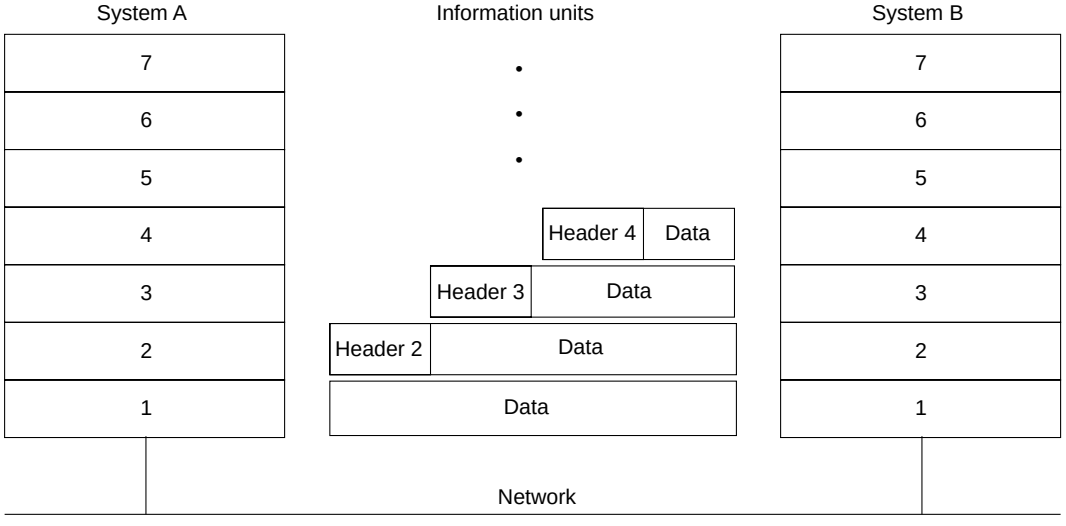
How does Layer 4 in System B know what Layer 4 in System A wants? Layer 4's specific requests are stored as *control information*, which is passed between peer layers in a block called a *header* that is prepended to the actual application information. For example, assume System A wishes to send the following text (called *data* or *information*) to System B:

The small grey cat ran up the wall to try to catch the red bird.

This text is passed from the application program in System A to System A's top layer. System A's application layer must communicate certain information to System B's application layer, so it prepends that control information (in the form of a coded header) to the actual text to be moved. This information unit is passed to System A's Layer 6, which may prepend its own control information. The information unit grows in size as it descends through the layers until it reaches the network, where the original text and all associated control information travels to System B, where it is absorbed by System B's Layer 1. System B's Layer 1 strips the Layer 1 header, reads it, and then knows how to process the information unit. The slightly smaller information unit is passed to Layer 2, which strips the Layer 2 header, analyzes the header for actions Layer 2 must take, and so forth. When the information unit finally reaches the application program in System B, it simply contains the original text.

The concept of a header and data is relative, depending on the perspective of the layer currently analyzing the information unit. For example, to Layer 3, an information unit consists of a Layer 3 header and the data that follows. Layer 3's data, however, can potentially contain headers from Layers 4, 5, 6, and 7. Further, Layer 3's header is simply data to Layer 2. This concept is illustrated in Figure 1-3. Finally, not all layers need to append headers. Some layers simply perform a transformation on the actual data they receive to make the data more or less readable to their adjacent layers.

Figure 1-3 Headers and Data



Compatibility Issues

The OSI reference model is not a network implementation. Instead, it specifies the functions of each layer. In this way, it is like a blueprint for the building of a ship. After a ship blueprint is complete, the ship must still be built. Any number of shipbuilding companies can be contracted to do the actual work, just as any number of network vendors can build a protocol implementation from a protocol specification. And, unless the blueprint is extremely (impossibly) comprehensive, ships built by different shipbuilding companies using the same blueprint will differ from each other in at least minor ways. At the very least, for example, it is likely that the rivets will be in different places.

What accounts for the differences between implementations of the same ship blueprint (or protocol specification)? In part, the differences are due to the inability of any specification to consider every possible implementation detail. Also, different implementors will no doubt interpret the blueprint in slightly different ways. And, finally, the inevitable implementation errors will cause different implementations to differ in execution. This explains why one company's implementation of protocol X does not always interoperate with another company's implementation of that protocol.

OSI Layers

Now that the basic features of the OSI layered approach have been described, each individual OSI layer and its functions can be discussed. Each layer has a predetermined set of functions it must perform for communication to occur.

Application Layer

The application layer is the OSI layer closest to the user. It differs from the other layers in that it does not provide services to any other OSI layer, but rather to application processes lying outside the scope of the OSI model. Examples of such application processes include spreadsheet programs, word-processing programs, banking terminal programs, and so on.

The application layer identifies and establishes the availability of intended communication partners, synchronizes cooperating applications, and establishes agreement on procedures for error recovery and control of data integrity. Also, the application layer determines whether sufficient resources for the intended communication exist.

Presentation Layer

The presentation layer ensures that information sent by the application layer of one system will be readable by the application layer of another system. If necessary, the presentation layer translates between multiple data representation formats by using a common data representation format.

The presentation layer concerns itself not only with the format and representation of actual user data, but also with data structures used by programs. Therefore, in addition to actual data format transformation (if necessary), the presentation layer negotiates data transfer syntax for the application layer.

Session Layer

As its name implies, the session layer establishes, manages, and terminates sessions between applications. Sessions consist of dialogue between two or more presentation entities (recall that the session layer provides its services to the presentation layer). The session layer synchronizes dialogue between presentation layer entities and manages their data exchange. In addition to basic regulation of conversations (sessions), the session layer offers provisions for data expedition, class of service, and exception reporting of session-layer, presentation-layer, and application-layer problems.

Transport Layer

The boundary between the session layer and the transport layer can be thought of as the boundary between application-layer protocols and lower-layer protocols. Whereas the application, presentation, and session layers are concerned with application issues, the lower four layers are concerned with data transport issues.

The transport layer attempts to provide a data transport service that shields the upper layers from transport implementation details. Specifically, issues such as how reliable transport over an internetwork is accomplished are the concern of the transport layer. In providing reliable service, the transport layer provides mechanisms for the establishment, maintenance, and orderly termination of virtual circuits, transport fault detection and recovery, and information flow control (to prevent one system from overrunning another with data).

Network Layer

The network layer is a complex layer that provides connectivity and path selection between two end systems that may be located on geographically diverse *subnetworks*. A subnetwork, in this instance, is essentially a single network cable (sometimes called a *segment*).

Because a substantial geographic distance and many subnetworks can separate two end systems desiring communication, the network layer is the domain of routing. Routing protocols select optimal paths through the series of interconnected subnetworks. Traditional network-layer protocols then move information along these paths.

Link Layer

The link layer (formally referred to as the data link layer) provides reliable transit of data across a physical link. In so doing, the link layer is concerned with *physical* (as opposed to *network*, or *logical*) addressing, network topology, line discipline (how end systems will use the network link), error notification, ordered delivery of frames, and flow control.

Physical Layer

The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems. Such characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other, similar, attributes are defined by physical layer specifications.

Important Terms and Concepts

Internetworking, like other sciences, has a terminology and knowledge base all its own. Unfortunately, because the science of internetworking is so young, universal agreement on the meaning of networking concepts and terms has not yet occurred. Definitions of internetworking terms will become more rigidly defined and used as the internetworking industry matures.

Addressing

Locating computer systems on an internetwork is an essential component of any network system. There are various addressing schemes used for this purpose, depending on the protocol family being used. In other words, AppleTalk addressing is different from TCP/IP addressing, which in turn is different from OSI addressing, and so on.

Two important types of addresses are *link-layer* addresses and *network-layer* addresses. Link-layer addresses (also called *physical* or *hardware* addresses) are typically unique for each network connection. In fact, for most local-area networks (LANs), link-layer addresses are resident in the interface circuitry and are assigned by the organization that defined the protocol standard represented by the interface. Because most computer systems have one physical network connection, they have only a single link-layer address. Routers and other systems connected to multiple physical networks can have multiple link-layer addresses. As their name implies, link-layer addresses exist at Layer 2 of the OSI reference model.

Network-layer addresses (also called *virtual* or *logical addresses*) exist at Layer 3 of the OSI reference model. Unlike link-layer addresses, which usually exist within a flat address space, network-layer addresses are usually hierarchical. In other words, they are like mail addresses, which describe a person's location by providing a country, a state, a zip code, a city, a street, an address on the street, and finally, a name. One good example of a flat address space is the U.S. social security numbering system, where each person has a single, unique social security number.

Hierarchical addresses make address sorting and recall easier by eliminating large blocks of logically similar addresses through a series of comparison operations. For example, we can eliminate all other countries if an address specifies the country Ireland. Easy sorting and recall is one reason that routers use network-layer addresses as the basis for routing.

Network-layer addresses differ depending on the protocol family being used, but they typically use similar logical divisions to find computer systems on an internetwork. Some of these logical divisions are based on physical network characteristics (such as the network segment a system is located on); others are based on groupings that have no physical basis (for example, the AppleTalk *zone*).

Frames, Packets, and Messages

Once addresses have located computer systems, information can be exchanged between two or more of these systems. Networking literature is inconsistent in naming the logically grouped units of information that move between computer systems. The terms *frame*, *packet*, *protocol data unit*, *PDU*, *segment*, *message*, and others have all been used, based on the whim of those who write protocol specifications.

In this publication, the term *frame* denotes an information unit whose source and destination is a link-layer entity. The term *packet* denotes an information unit whose source and destination is a network-layer entity. Finally, the term *message* denotes an information unit whose source and destination entity exists above the network layer. *Message* is also used to refer to particular lower-layer information units with a specific, well-defined purpose.

Key Organizations

Without the services of several key standards organizations, the world of networking would be substantially more chaotic than it is currently. Standards organizations provide forums for discussion, help turn discussion into formal specifications, and proliferate the specifications once they complete the standardization process.

Most standards organizations have specific processes for turning ideas into formal standards. Although these processes differ slightly between standards organizations, they are similar in that they all iterate through several rounds of organizing ideas, discussing the ideas, developing draft standards, voting on all or certain aspects of the standards, and finally formally releasing the completed standard to the public.

Some of the better-known standards organizations follow:

- International Organization for Standardization (ISO)—An international standards organization responsible for a wide range of standards, including those relevant to networking. This organization is responsible for the OSI reference model and the OSI protocol suite.
- American National Standards Institute (ANSI)—The coordinating body for voluntary standards groups within the United States. ANSI is a member of ISO. ANSI's best-known communications standard is FDDI.
- Electronic Industries Association (EIA)—A group that specifies electrical transmission standards. EIA's best-known standard is EIA/TIA-232 (formerly RS-232).
- Institute of Electrical and Electronic Engineers (IEEE)—A professional organization that defines network standards. IEEE LAN standards (including IEEE 802.3 and IEEE 802.5) are the best-known IEEE communications standards and are the predominant LAN standards in the world today.
- International Telecommunication Union Telecommunication Standardization Sector (ITU-T) (formerly the Committee for International Telegraph and Telephone [CCITT])—An international organization that develops communication standards. The best-known ITU-T standard is X.25.
- Internet Activities Board (IAB)—A group of internetwork researchers who meet regularly to discuss issues pertinent to the Internet. This board sets much of the policy for the Internet through decisions and assignment of task forces to various issues. Some *Request for Comments* (RFC) documents are designated by the IAB as Internet standards, including *Transmission Control Protocol/Internet Protocol* (TCP/IP) and the *Simple Network Management Protocol* (SNMP).