CHAPTER 20

OSI Protocols

Background

In the early days of intercomputer communication, networking software was created in a haphazard, ad hoc fashion. When networks grew sufficiently popular, the need to standardize the by-products of network software and hardware development became clear. Standardization, it was believed, would allow vendors to create hardware and software systems that could communicate with one another, even if the underlying architectures were dissimilar. With this goal in mind, the International Organization for Standardization (ISO) began development of the *Open Systems Interconnection* (OSI) reference model. The OSI reference model was completed and released in 1984.

Today, the OSI reference model (discussed in detail in Chapter 1, "Introduction to Internetworking") is the world's most prominent networking architecture model. It is also the most popular tool for learning about networks. The OSI protocols, on the other hand, have had a long gestation period. While OSI implementations are not unheard of, the OSI protocols have not yet attained the popularity of many proprietary (for example, DECnet and AppleTalk) and de facto (for example, the Internet protocols) standards.

Technology Basics

The world of OSI networking has a unique terminology:

- *End system* (ES) refers to any nonrouting network device.
- *Intermediate system* (IS) refers to a router.
- *Area* is a group of contiguous networks and attached hosts that are specified to be an area by a network administrator or manager.
- *Domain* is a collection of connected areas. Routing domains provide full connectivity to all end systems within them.

Media Access

Like several other modern, seven-layer protocol stacks, the OSI stack includes many of today's popular media-access protocols. This allows other protocol stacks to exist alongside OSI on the same media. OSI includes *IEEE 802.2*, *IEEE 802.3*, *IEEE 802.5*, *FDDI*, *X.21*, *V.35*, *X.25*, and others. Most of these OSI media-access protocols are discussed elsewhere in this publication.

Network Layer

OSI offers both a connectionless and a connection-oriented network layer service. The connectionless service is described in ISO 8473 (usually referred to as *Connectionless Network Protocol* or *CLNP*). The connection-oriented service (sometimes called *Connection-Oriented Network Service*, or *CONS*) is described in ISO 8208 (*X.25 Packet-Level Protocol*, sometimes referred to as *Connection-Mode Network Protocol*, or *CMNP*) and ISO 8878 (which describes how to use ISO 8208 to provide OSI connection-oriented service). An additional document, ISO 8881, describes how to run the X.25 Packet-Level Protocol over IEEE 802 local-area networks (LANs). OSI also specifies several routing protocols, which are discussed in Chapter 28, "OSI Routing." X.25 is discussed in Chapter 12, "X.25."

In addition to the previously mentioned protocol and service specifications, other relevant OSI network-layer documents include the following:

- ISO 8648—This document is usually referred to as the *internal organization of the network layer* (IONL). It describes how the network layer can be divided into three separate and distinct sublayers to allow support for different subnetwork types.
- ISO 8348—This document is usually called the *network service definition*. It describes the connection-oriented and connectionless services provided by the OSI network layer. Network layer addressing is also defined in this document. The connectionless-mode service definition and the addressing definition were previously published as separate addenda to ISO 8348; however, the 1993 version of ISO 8348 folds all of the addenda into a single document.
- ISO TR 9575—This document describes the framework, concepts, and terminology used in OSI routing protocols.
- ISO TR 9577—This document describes how to discriminate between multiple network-layer protocols running on the same medium. Such discrimination is necessary because, unlike other protocols, OSI network-layer protocols cannot be identified through a protocol ID or similar field at the link layer.

Connectionless Service

As its name suggests, CLNP is a connectionless datagram protocol used to carry data and error indications. Functionally, it is quite similar to the *Internet Protocol* (IP) described in Chapter 18, "Internet Protocols." It has no facilities for error detection or correction, relying on the transport layer to provide these services as appropriate. It has only one phase, called *data transfer*. Each invocation of a service primitive is independent of all other invocations, requiring all address information to be completely contained within the service primitive.

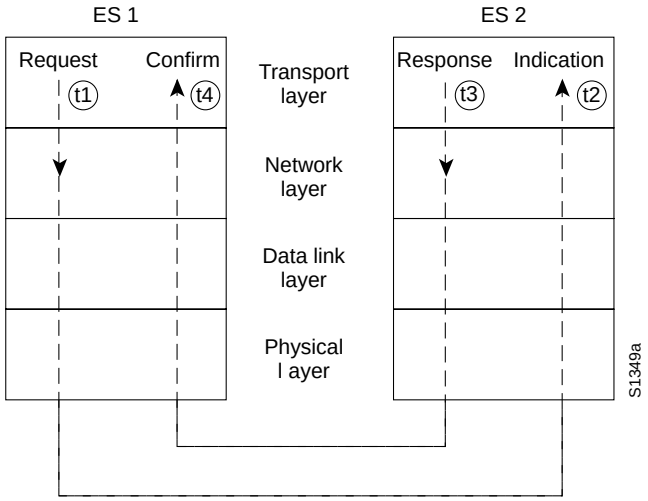
While CLNP defines the actual protocol performing typical network-layer functions, *Connectionless Network Service* (CLNS) describes a service provided to the transport layer in which a request to transfer data receives "best effort" delivery. Such delivery does not guarantee that data will not be lost, corrupted, misordered, or duplicated. Connectionless service assumes that the transport layer will correct such problems as necessary. CLNS does not provide any form of connection information or state, and connection setup is not performed. Because CLNS provides transport layers with the service interface to CLNP, CLNS and CLNP are often discussed together.

In the United States, federal agencies use the *Government Open Systems Interconnection Profile* (GOSIP) specification to ensure interoperability of CLNS network products. GOSIP requires certification of CLNP over LANs and X.25.

Connection-Oriented Service

OSI connection-oriented network service is specified by ISO 8208 and ISO 8878. OSI uses the X.25 Packet-Level Protocol for connection-oriented data movement and error indications. Six services are provided to transport-layer entities (one for connection establishment, one for connection release, and four for data transfer). Services are invoked by some combination of four *primitives: request*, *indication*, *response*, and *confirmation*. The four primitives interact as shown in Figure 20-1.

Figure 20-1 OSI Primitives



At time t-1, ES 1's transport layer sends a request primitive to ES 1's network layer. This request is placed onto ES 1's subnetwork by lower-layer subnetwork protocols, and is eventually received by ES 2, which sends the information up to the network layer. At time t-2, ES 2's network layer sends an indication primitive to its transport layer. After required upper-layer processing for this packet is complete, ES 2 initiates a response to ES 1 by using a response primitive sent from the transport layer to the network layer. The response, which occurs at time t-3, travels back to ES 1, which sends the information up to the network layer where a confirm primitive is generated and sent to the transport layer at time t-4.

Addressing

The OSI network service is provided to the transport layer through a conceptual point on the network/transport layer boundary known as a *network service access point* (NSAP). There is one NSAP per transport entity.

Each NSAP can be individually addressed in the global OSI internetwork through its *NSAP address* (often colloquially and imprecisely known as an NSAP). An OSI end system can have multiple NSAP addresses, in which case the addresses usually differ only in the last byte, known as the *n-selector.*

It is also useful to address a system's network layer without being associated with a specific transport entity—for instance, when a system participates in routing protocols or when addressing an intermediate system (router). Such addressing is done via a special network address known as a *network entity title* (NET). A NET is structurally identical to an NSAP address, but uses the special

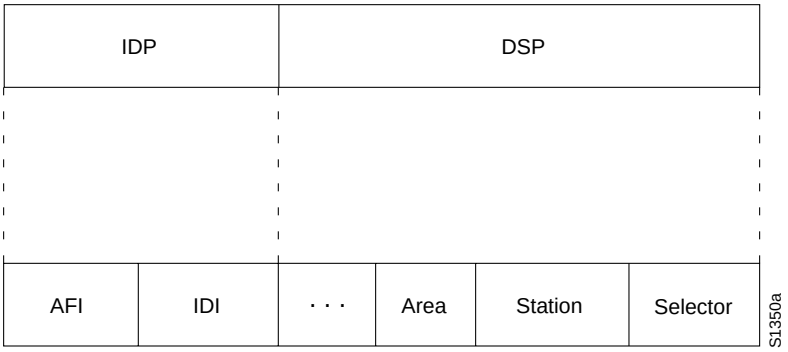
n-selector value “00.” Most end systems and intermediate systems have only a single NET, unlike IP routers which usually have one address per interface. However, an intermediate system that is participating in multiple areas or domains can choose to have multiple NETs.

NETs and NSAP addresses are hierarchical addresses. Addressing hierarchies facilitate both administration (by allowing multiple levels of address administration) and routing (by encoding network topology information). An NSAP address is first separated into two parts: the *initial domain part* (IDP) and the *domain specific part* (DSP). The IDP is then further divided into the *authority and format identifier* (AFI) and the *initial domain identifier* (IDI).

The AFI provides information about the structure and content of the IDI and DSP fields, including whether the IDI is of variable length and whether the DSP uses decimal or binary notation. The IDI specifies an entity that can assign values to the DSP portion of the address.

The DSP is then further subdivided by the authority responsible for its administration. Typically, another administrative authority identifier follows, allowing address administration to be further delegated to subauthorities. Following that comes information used for routing, such as the routing domain, the area with the routing domain, the station ID within the area, and the selector within the station. Figure 20-2 illustrates the OSI address format.

Figure 20-2 OSI Address Format



Transport Layer

As with the OSI network layer, both connectionless and connection-oriented transport services are offered. There are actually five connection-oriented OSI transport protocols: *TP0*, *TP1*, *TP2*, *TP3*, and *TP4*. All but TP4 work only with OSI’s connection-oriented network service. TP4 works with both connection-oriented and connectionless services.

TP0 is the simplest OSI connection-oriented transport protocol. Of the classical transport layer functions, it performs only segmentation and reassembly. This means that TP0 will note the smallest maximum size *protocol data unit* (PDU) supported by the underlying subnetworks, and will break the transport packet into smaller pieces that are not too big for network transmission.

In addition to segmentation and reassembly, TP1 offers basic error recovery. It numbers all PDUs and resends those that are unacknowledged. TP1 can also reinitiate connections when excessive unacknowledged PDUs occur.

TP2 can multiplex and demultiplex data streams over a single virtual circuit. This capability makes TP2 particularly useful over public data networks (PDNs), where each virtual circuit incurs a separate charge. Like TP0 and TP1, TP2 also segments and reassembles PDUs.

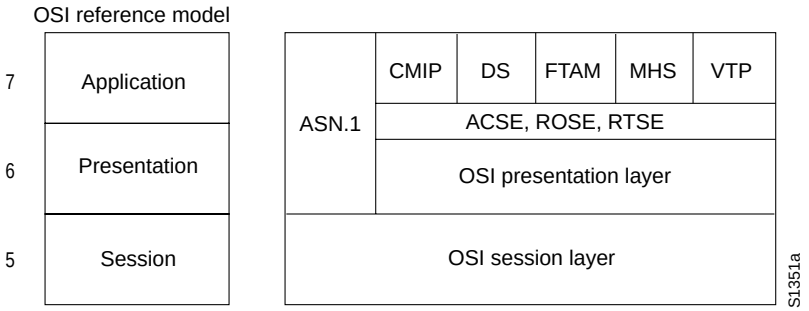
TP3 combines the features or TP1 and TP2.

TP4 is the most popular OSI transport protocol. TP4 is similar to the Internet protocol suite's *Transmission Control Protocol* (TCP) and, in fact, was based on TCP. In addition to TP3's features, TP4 provides reliable transport service. It assumes a network in which problems are not detected.

Upper-Layer Protocols

Principal OSI upper-layer protocols are shown in Figure 20-3.

Figure 20-3 Principal OSI Upper-Layer Protocols



Session Layer

The OSI session-layer protocol turns the data streams provided by the lower four layers into sessions by implementing various control mechanisms. These mechanisms include accounting, conversation control (that is, determining who can talk when), and session parameter negotiation.

Session conversation control is implemented by use of a *token*, the possession of which provides the right to communicate. The token can be requested, and ESs can be given priorities that provide for unequal token use.

Presentation Layer

The OSI presentation layer is typically just a pass-through protocol for information from adjacent layers. Although many people believe that *Abstract Syntax Notation 1* (ASN.1) is OSI's presentation-layer protocol, ASN.1 is used for expressing data formats in a machine-independent format. This allows communication between applications on diverse computer systems (ESs) in a manner transparent to the applications.

Application Layer

The OSI application layer includes actual applications as well as *application service elements* (ASEs). ASEs allows easy communication from applications to lower layers. The three most important ASEs are *Association Control Service Element* (ACSE), *Remote Operations Service Element* (ROSE), and *Reliable Transfer Service Element* (RTSE). ACSE associates application names with one another in preparation for application-to-application communication. ROSE implements a generic request-reply mechanism that permits remote operations in a manner similar to that of *remote procedure calls* (RPCs). RTSE aids reliable delivery by making session-layer constructs easy to use.

Five OSI applications receive the most attention:

- *Common Management Information Protocol* (CMIP)—Provides network management capabilities. Like SNMP and NetView, CMIP allows exchange of management information between ESs and management stations (which are also ESs). For more information about SNMP and NetView, see Chapter 32, "Simple Network Management Protocol," and Chapter 33, "IBM Network Management."
- *Directory Services* (DS)—Derived from the Consultative Committee for International Telegraph and Telephone (CCITT) (now referred to as the InternationalTelecommunication Union Telecommunication Standardization Sector [ITU-T]) X.500 specification, this service provides distributed database capabilities useful for upper-layer node identification and addressing.
- *File Transfer, Access, and Management* (FTAM)—Provides file transfer service. In addition to classical file transfer, for which FTAM provides numerous options, FTAM also offers distributed file access facilities in the spirit of NetWare from Novell, Inc. or Network File System (NFS) from Sun Microsystems, Inc.
- *Message Handling Systems* (MHS)—Provides an underlying transport mechanism for electronic messaging applications and other applications desiring store-and-forward services. Although they accomplish similar purposes, MHS is not to be confused with Novell's NetWare MHS. (For more information about NetWare MHS, see Chapter 19, "NetWare Protocols."
- *Virtual Terminal Protocol* (VTP)—Provides terminal emulation. In other words, it allows a computer system to appear to a remote ES as if it were a directly attached terminal.With VTP, users can, for example, run remote jobs on mainframes.