

The ISO Reference Model of Open Systems Interconnection: A First Tutorial

Leslie Jill Miller
Xerox Corporation
Rochester, NY 14644

1.0 Introduction

During the 1970's computing facilities which had previously provided stand-alone processing were linked into a variety of networks under quite different geographical, technical, financial and application constraints. The effect of these independent developments was that similar services were often provided by quite different mechanisms, and interconnection became extremely difficult.

Late in the 1970's a strong desire for a standard means of internetwork communication arose. It was realized that a common model of distributed data communications would be required to facilitate the development of these standards. Intensive activity was initiated by several national and international standards organizations, including the International Organization for Standardization (ISO). In 1978 the "Reference Model of Open Systems Interconnection" [8] was produced, and this led to the 1981 Draft International Standard 7498 "Data Processing - Open Systems Interconnection - Basic Reference Model" [2]. The documents assume an environment of systems which make themselves "open" to the exchange of data with other systems by adhering to a well-defined set of standard procedures for communication. The documents' scope includes not only the basic transfer of information between such systems, but also the capability of such systems to work together in support of distributed applications.

The Open Systems Interconnection model is intended to facilitate standards development and is primarily aimed towards the writers of future standards in each of the areas outlined by the reference model. Interest in the model, however, has increased enormously and several tutorials have been written on the model and associated standards activities [3], [7], [11]. This tutorial is much more intuitive in nature and is intended as a first tutorial for the reader with only a minimal background in network communication. It begins by motivating the set of required services and then discussing how certain architectural features are supported throughout the model as a whole.

The next section gives a very informal introduction to some common services needed to support communication within a distributed environment. The services are introduced, as required, in order to get messages from one mythical kingdom to another in Medieval times. Despite the out-of-data imagery, the communication techniques used and the conceptual model developed are directly analogous to those of the ISO reference model. In section three the seven layers of the ISO model are briefly described. A set of functions which are important across several layers of the model is then discussed in section four. These include connection management, data transfer, and error detection and correction.

2.0 Diplomatic Communication in Medieval Times

This example deals with the two Medieval kingdoms of Altdorf and Blitzen. The two kingdoms have traditionally been enemies and as a result the only communication between them consists of itinerant peddlers who carry a few goods back and forth across the long and hazardous path between them. The day arrives, however, when a new King comes to the throne of Altdorf and decides to propose an alliance with the kingdom of Blitzen. Thus, he writes a letter to the King of Blitzen, and commands his Foreign Minister to see that it is delivered.

This is a typical case of an entity in one system wishing to communicate with an entity in another system. As in today's communications environment, there are difficulties in sending minimal communications of any sort between the two end systems, and there are also problems of reliability, language differences and etiquette to overcome.

The Foreign Minister of Altdorf is somewhat more experienced in diplomacy than the young King, and he decides to ensure a favorable reception before actually sending the King's own words to these traditional enemies. He keeps the King's letter for the time being, and writes his own letter to the Foreign Minister of Blitzen, inquiring as to whether that kingdom is willing to establish diplomatic relations with Altdorf. Having

written this letter, he gives it to the Purchasing Agent of Altdorf with instructions to see that it arrives at Blitzen as soon as possible.

The Purchasing Agent is not happy with his new assignment. He understands very well the hazards of not carrying out this assignment satisfactorily. He decides to improve his chances of success by copying the Foreign Minister's letter, and then sending a copy of this letter with a note of his own to the Purchasing Agent of Blitzen. His covering note requests that the distant Purchasing Agent immediately send back a note of acknowledgement, and that the copy of the Foreign Minister's letter be passed on to the Foreign Minister of Blitzen. The copy of the Foreign Minister's letter and the Purchasing Agent's note are given to the next Peddler headed for Blitzen.

With great good fortune, the Peddler manages to escape bandits who attempt to rob him, and eventually arrives at Blitzen, where the Purchasing Agent takes the letter, passes it on to the Foreign Minister, and sends back a note of acknowledgement with the next Peddler. The Purchasing Agent of Altdorf is very relieved to receive this note. Meanwhile, the court of Blitzen is in an uproar at this inquiry into establishing diplomatic relations with their long-time enemies. Finally, however, the decision is made to return an affirmative response, and the letter from the Foreign Minister of Blitzen is passed to the local Purchasing Agent for delivery. He recalls the request for acknowledgement he had received and decides to use the same scheme. A copy of the Foreign Minister's letter and his covering note are sent off, the Peddler survives his bout with the plague, and the letter and note are received by the Purchasing Agent of Altdorf. The letter from the Foreign Minister of Blitzen is delivered to Altdorf's Foreign Minister and an acknowledgement note is returned to the Purchasing Agent of Blitzen.

The Foreign Minister of Altdorf is delighted that all has gone so well and he gives the King's letter and a covering letter addressed to the Foreign Minister of Blitzen to the Purchasing Agent for delivery. The Purchasing Agent copies the letters, attaches his note requesting acknowledgement, sends them off with the next Peddler, and sits back to wait for his acknowledgement. Three weeks later the acknowledgement has not arrived. (The peddler had an unfortunate encounter with the Purple Dragon in a remote mountain pass.) The Purchasing Agent, nervous about the passing time but delighted to still have a copy of these important letters, sends another copy. This one is received, the Purchasing Agent's acknowledgement note is sent and received, the letter to the Foreign Minister is delivered, and the enclosed letter to the King is finally delivered.

This exchange of messages is a somewhat confusing one, involving two copies of one letter written by the King of Altdorf,

a total of three copies of the two letters written by the Foreign Minister of Altdorf, one copy of the letter written by the Foreign Minister of Blitzen, plus a total of four notes and acknowledgements sent by the Purchasing Agent of Altdorf and another three sent by the Purchasing Agent of Blitzen. Thirteen notes, and letters or copies of letters are required to get one letter reliably delivered to a receptive destination!

This exchange of messages is much easier to describe if the messages for each set of actors in this Medieval pageant (Kings, Foreign Ministers, Purchasing Agents) are considered separately. The actor at each level uses services provided by lower level actors and provides services to higher level actors. In other words, we can use *layering* as a structuring technique to organize and minimize the complexity of the message exchanges, and to enhance our understanding of the communication process.

At the King's level, letters are assumed to be reliably delivered and favorably received, and their content has to do with such lofty issues as negotiating alliances and arranging royal marriages: the business of Kings.

The Foreign Minister level assumes reliable delivery of all letters, and its letters involve the establishment of a cooperating relationship or connection, in the context of which the King's highest level letters may be sent.

The Purchasing Agent level uses the rather unreliable message-carrying services of the lowest Peddler level to provide a *reliable* letter passing service for the higher Foreign Minister level. The reliability is achieved by the acknowledgement mechanism, using retransmission after the expiration of a predetermined timeout interval. The notes used in this level are notes requesting acknowledgement (containing uninterpreted higher-level data - the Foreign Minister's letters), and notes containing acknowledgements (without any higher-level data).

This example illustrates many features common to distributed communication today.

- Entities at each layer treat lower layers as simply the providers of a service, with no interest in how the service is provided.
- The lowest layers of communication tend to be unreliable.
- Only the highest layers understand the semantics of the original communication.
- At each layer of communication, a formal set of procedures is followed by corresponding entities at that layer. The Foreign Ministers, for example, know all the niceties involved in establishing diplomatic relations between kingdoms.

- Connections are often established at different layers. In this example, the Foreign Ministers explicitly establish one such connection.

- Messages tend to proliferate in managing connections and providing reliability.

- Meaningful information at one layer is encapsulated within lower layer messages, but is treated as uninterpreted data by those layers.

This section has introduced the types of services required to support communication during those difficult Medieval times. It introduced layering as an appropriate tool for managing the complexity of the full set of services. The next section presents a very similar but more general model for interconnecting "open" systems which was developed by the International Organization for Standardization.

3.0 Basic Reference Model of Open Systems Interconnection

The Basic Reference Model for Open Systems Interconnection postulates that each system include seven logical subsystems, each communicating with its next lower and next higher subsystems. The set of all subsystems at the same level across all systems is collectively referred to as a *layer* in the architecture. Each layer invokes certain services of the next lower layer, and each provides services to the next higher layer.

The flow of information between peer entities within a single layer is governed by a layer-specific *protocol* which specifies the rules for exchanging information and the formats for the required messages. Information is exchanged via *protocol-data-units* consisting of control information and optional higher-layer data. Layer N protocol-data-units are actually transmitted between peer entities by encapsulating them as uninterpreted data within layer N-1 protocol-data-units, just as the King's letter was encapsulated within the Foreign Minister's letter. Only at the lowest layer is there direct communication via the physical media connecting the two open systems.

Within one system, interface-data-units are passed across *interfaces* between the layered subsystems. Information thus originates at the highest layer of the source system; successively passes down through several interfaces, at each layer being further encapsulated within that layer's protocol-data-unit; is actually transmitted from source to destination system at the lowest layer; and then successively passed up through several interfaces at the destination system, at each layer having encapsulating headers removed until the original protocol-data-unit arrives at the highest layer.

INTERNATIONAL STANDARDS ORGANIZATION REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION

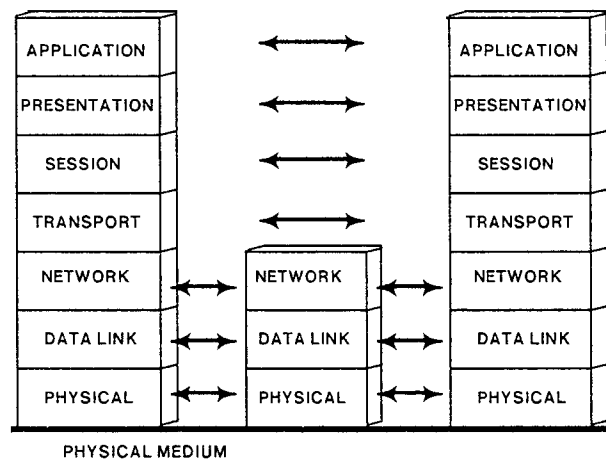


Figure 1 is a common depiction of the ISO architecture, with two open systems communicating across an intervening network, here represented by a single switch which provides the services of the lowest three layers. We now describe some of the services specified for each of the seven layers in the Reference Model, and give examples of existing protocols which belong within each layer.

Layer 7, the Application Layer, directly serves the end user by providing the appropriate distributed information service. It provides all services directly comprehensible to the user. In a computer networking environment the Application Layer protocols implement sets of services such as, for example, those required for creating, deleting, and copying files between systems, or those required for sending and retrieving electronic mail.

Layer 7 protocols correspond to the conventions the King used in sending his alliance proposal to Blitzen: identifying the recipient as the King of Blitzen, and perhaps specifying how future letters are to be authenticated or suggesting terms of the proposed alliance.

Layer 6, the Presentation Layer, deals with the representation and manipulation of structured data, including the management of data formats and the transformations on this data. The purpose of the layer is to represent information to the communicating Application Layer entities in a way that preserves meaning while resolving syntax differences. Thus, for example, in support of the Application Layer file transfer commands, the Presentation entity may map from the standard local syntax which the user prefers to employ, to the syntax used at the distant file system.

Although this layer was not present in our Medieval example, it is easily introduced. Assume that the people of Altdorf speak French and those of Blitzen speak German, but that Priests in both kingdoms speak Latin. The functions of the Presentation Layer could be carried out by a Priest at Altdorf translating the King's letter from French into Latin, and a Priest at Blitzen translating from Latin into German. The language of communication is Latin but the Presentation entities at each end present the semantics of the communication in the syntax desired by the Application Layer entities.

The Session Layer, Layer 5, supports a session administration service which provides for the binding and unbinding of two Presentation entities into a logical relationship called a session-connection, and supports a session dialogue service within the session-connection. This session dialogue service controls data exchange by delimiting and quarantining data, synchronizing the data exchange, and managing the sequencing of interactions. Within an interconnected open system, it is the session-connection established between users and providers of services that carries accounting and access control information.

In the Medieval example, the Foreign Minister Layer corresponds to the Session Layer, with diplomatic relations corresponding to the session-connection, and diplomatic conventions corresponding to the interaction management.

Layer 4, the Transport Layer, is intended to provide transparent, reliable and cost-effective data transfer end-to-end between the source and destination systems. The quality of service is selectable, with the Transport Layer selecting or enhancing the underlying Network services so as to provide the class of service requested by the Session entity. Examples of highly reliable transport protocols which provide sequenced, flow-controlled delivery of data include the Department of Defense's connection-oriented Transmission Control Protocol (TCP) [5], and the transaction-oriented Delta-t protocol [10] used on the high performance local network called Octopus at Lawrence Livermore Laboratories. The Transport protocols being developed by ISO [9] provide for five different classes of connection-oriented service, ranging from minimal Teletex compatible service up to service comparable to that provided by TCP.

The Purchasing Agents in our Medieval example correspond to Transport entities, enhancing the basic delivery service of the underlying Peddler network through their acknowledgement and retransmission protocol.

Layer 3, the Network Layer, provides the means for routing data between systems containing communicating application entities. It handles the routing, switching and relaying of data within a single administrative network or over parallel or tandem subnetworks combined to form an internetwork, and it provides

these transfer services at a known cost. CCITT's X.25 is a well known network interface standard, but there are few standard network protocols. There are, however, standard internetwork protocols such as those for handling the datagrams of the Department of Defense Internet [4] and the Xerox Pup architecture [1].

The Medieval example breaks down somewhat for the lower layers of the ISO Reference Model, with the Peddler Layer providing the services of the three lower layers, either through individuals carrying the Purchasing Agents' messages themselves or passing them on to fellow Peddlers for delivery.

Layer 2, the Data Link Layer, involves the transfer of logical blocks of data across a link in such a way as to mask the physical characteristics of Layer 1. While different degrees of error detection and correction for these blocks of data may be provided by different link layer protocols, a primary motivation for including this layer in the architecture is to allow for localized identification and correction of transmission errors across individual links. The layer provides functional and procedural means for managing Data Link connections and may provide capabilities for controlling circuit switching. Typical protocols at this layer are IBM's SDLC, ISO's HDLC, ANSI's ADCCP, and X.25 LAPB.

Layer 1, the Physical Layer, is concerned with the actual means of bit transmission across a physical medium. It provides the mechanical, electrical, functional, and procedural functions necessary to transmit and interpret signals sent across the physical medium between such entities as terminals, computers, modems and network switches. Typical protocols at this level are EIA's RS 232-C and RS 449, which provide for the transparent flow of bits across a physical medium.

The above description has been very cursory but it indicates the primary importance of layering as a structuring tool in defining systems which are "open" for communication, and it outlines the services which each layer is intended to support.

4.0 Common Protocol Functions

Protocols at different layers of the reference model include such common functions as connection management, data transfer and error detection and correction. This section illustrates the variety of protocol capabilities in each of these three areas. The design of an efficient communication system based on layered protocols requires that the different protocols provide complementary services which avoid replicating identical functions.

4.1 Connection Management

The current ISO Reference Model for Open Systems

Interconnection generally requires that a connection be explicitly established and released at each layer in order to support inter-system communication. The establishment of such a connection typically requires at least a message in one direction requesting a connection, and a second in the other direction agreeing to the connection. User data may or may not be carried by the connection establishment request and response. Different protocols also differ in the reliability of the connections which they establish, especially in their ability to differentiate between different instances over time of the desired connection. (A connection may be broken and re-established. The problem arises if data from the old connection instance arrives after the new connection is established.) Protocols also differ in their provision of a "graceful close", only terminating the connection after all transmitted data has been received at the destination. Typically connections at the higher protocol layers pay the price of more complex connection management schemes to get better performance. Lower layers can be less careful since any errors will be caught later at the higher layers.

The connection management required by the Reference model involves a large overhead. Alternatively, at most of the layers it is also possible to employ connectionless protocols, and it is expected that extensions to the current Reference Model will incorporate this fact. The Ethernet Data Link Protocol [6], for example, specifies that the destination should accept single autonomous frames addressed to that work station; there is none of the state-saving required for connections being maintained between protocol-data-units at that layer. At Layer 3 there may be connections such as the virtual circuits of an X.25 network or X.75 internetwork, or information may be transmitted in independently routed, unrelated datagrams such as occurs in the Department of Defense Internet and the Xerox Pup Architecture. At the Transport Layer connections are almost always established to achieve the desired reliability for blocks of data. The Delta-t protocol used on Lawrence Livermore Laboratory's Octopus Network is unusual in being transaction oriented, supporting the *reliable* delivery of smaller pieces of usually independent data. Similarly, there exist protocols at higher layers which mimic the effect of procedure calls, with minimal state-saving between calls.

The model explicitly includes multiplexing of connections. This allows for many higher layer connections to be mapped into one lower layer connection, as occurs in X.25 when one Physical Layer connection is multiplexed into many Data Link Layer connections, or as occurs when many Transport connections are mapped into one Network connection to minimize tariffs. Downward multiplexing or splitting is also accommodated, with one higher layer connection mapped into several lower layer connections. Thus, a Session connection might employ one Transport connection for normal data transfer and another for expedited data transfer, or one Transport connection might use

several Network connections to increase throughput or reliability.

4.2 Data Transfer

Each layer in the Reference Model not only is capable of exchanging control information local to that layer, but also is capable of transparently transferring higher layer protocol-data-units. This normal data transfer occurs when the protocol-data-units of one layer are passed across the interface to the next lower layer and are then encapsulated within the protocol-data-units of that lower layer for transmission to the destination system. The transfer across each interface may be accomplished by procedure call, message passing within the single system or temporary encapsulation within interface-data-units.

The model also incorporates expedited data flow, with smaller amounts of data which the layer guarantees will not be delivered after any subsequently sent normal or expedited data. The concept of expedited data applies mainly to layers 3, 4 and 5, and may be implemented using one or more connections.

Flow control is a function which is common to many layers in the Reference model. It may be applied to either protocol-data-units or to interface-data-units. Its intent is to prevent a source from sending more data than its destination is capable of accepting. At the Data Link Layer it may take the form of directives from destination to source to simply stop sending, or may be advisory with an indication of the number of additional packets the destination is capable of receiving. In the Data Link Layer of the Xerox Ethernet it takes the form of a collision avoidance and backoff-before-retransmission strategy, which controls flow on the link itself rather than between a source/destination pair. The virtual circuit X.25 nets provide interface flow-control and let the effects back up from the destination through the network, while datagram networks typically have no internal flow-control. Almost all Transport protocols provide a form of flow control for their normal data transfer in which the destination indicates a "window" of packet, message or byte numbers within a connection which it is willing to accept. At the Application Layer, treating communication as a series of remote procedure calls also exhibits a flow control effect, since the return from one procedure call must precede the next call (the window has size 1).

There are no restrictions on the relative sizes of protocol-data-units for different layers in the ISO Reference Model, and yet higher layer protocol-data-units must be made to fit within those of lower layers. This results in 1-to-many and many-to-1 mappings between the high level and low level data units. This is called segmentation when the higher layer data unit is larger and must be broken up, and is called blocking when the higher

level data unit is small enough to justify several being incorporated into one lower layer data unit. In the case of tandem networks with different packet sizes the breakup of big packets into several small packets is termed fragmentation and the resulting fragments must be reassembled either on exiting the immediate network or on exiting the internet.

4.3 Error Functions

At each layer of the architecture, errors which could not be detected by lower layer functions may become detectable. A wide variety of different techniques are used to enhance error detection capabilities at different layers. At the Physical Layer in the Ethernet, for example, bits are transmitted using the redundant Manchester encoding so as to decrease the probability of errors and increase the probability of error detection. It is common in Data Link Layer protocols to use a cyclic redundancy check (CRC) or frame check sequence (FCS) appended to the protocol-data-unit to check the integrity of the entire unit. This requires the computation of some function of all the bytes of the protocol-data-unit both before and after transmission and a comparison of the results. Similarly, Transport Layer Protocols often use a CRC or checksum for checking the integrity of that layer's protocol-data-unit. Even at the Application Layer, requests for service are acknowledged by responses carrying redundant information so as to increase the probability of errors being detected.

Once errors have been detected, a given layer either may take action to correct the error or may simply notify the next higher layer that an error has occurred. The SDLC / HDLC / ADCCP family of Data Link protocols, for example, incorporates a retransmission scheme designed to remove all errors detected at the destination using the CRC. The Ethernet Data Link Protocol, on the other hand, retransmits only when collisions are sensed by the source during transmission. If errors are detected at the destination through use of that protocol's frame check sequence then the next higher layer is notified but correction is not attempted within the Data Link Layer. Networks may or may not attempt internal error recovery, but highly reliable Transport protocols do support error recovery, usually based on an acknowledgement scheme with retransmission after timeouts.

5.0 Conclusions

In this paper Medieval society has been compared to current computer networks, with Medieval diplomacy playing the same role as Open Systems Interconnection. Both situations require many distinct sets of services to achieve effective communication, and conscious layering of the required

functions offers significant conceptual and organizational advantages. In both cases, an efficient layered implementation requires that the set of protocols be carefully chosen to avoid unnecessary replication of services.

Bibliography

1. Boggs, D. R., Shoch, J. F., Taft, E. A., and Metcalfe, R. M., "Pup: An Internetwork Architecture", IEEE Transactions on Communications, April 1980
2. "Data Processing - Open Systems Interconnection - Basic Reference Model", ISO/TC97/SC16/DIS7498
3. desJardins, R., and White, G., "ISO/ANSI Reference Model of Open Systems Interconnection," Proceedings of Trends and Applications: 1980 - Computer Network Protocols, NBS, May 1980, pp.47-53
4. "DoD Standard Internet Protocol, Jan. 1980," Available through Defense Advanced Research Project Agency, IPTO, Arlington VA
5. "DoD Standard Transmission Control Protocol, Jan. 1980," Available through Defense Advanced Research Project Agency, IPTO, Arlington VA
6. "The Ethernet - A Local Area Network - Data Link Layer and Physical Layer Specifications," Xerox Corporation, Stamford, CT, Sept. 30, 1980
7. Miller, L. J., "Protocol Standardization", EASCON'80, Oct. 1980, pp. 507-513
8. "Reference Model of Open Systems Interconnection" (Version 1 as of 1978), ISO/TC97/SC16/N46
9. "Transport Protocol," ISO/TC97/SC16/N563
10. Watson, R. W., "Preliminary Delta-t Protocol Specification," Lawrence Livermore Laboratory Report UCRL-52881, Nov. 1979
11. Zimmerman, H., "OSI Reference Model -- The ISO Model of Architecture for Open Systems Interconnection," IEEE Transactions on Communication, COM-28 4, Apr. 1980