THE OSI MODEL: OVERVIEW ON THE SEVEN LAYERS OF COMPUTER NETWORKS

The Open Systems Interconnection model (OSI model) is a product of Open Systems Interconnection effort at the International Organization for Standardization. It is a way of sub-dividing a communications system into smaller parts called layers. A layer is a collection of conceptually similar functions that provide services to the layer above it and receives services from the layer below it. On each layer an *instance* provides services to the instances at the layer above and requests service from the layer below. For example, a layer that provides error-free communications across a network provides the path needed by applications above it, while it calls the next lower.

I. HISTORY ABOUT OSI MODEL

Work on a layered model of network architecture was started and the International Organization for Standardization (ISO) began to develop its OSI framework architecture. OSI had two major components: an *abstract model* of networking, called the Basic Reference Model or seven-layer model, and a set of specific protocols.

The concept of a seven-layer model was provided by the work of Charles Bachman, Honeywell Information Services. Various aspects of OSI design evolved from experiences with the ARPANET, the fledgling Internet, NPLNET, EIN, CYCLADES network and the work in IFIP WG6.1. The new design was documented in ISO 7498 and its various addenda. In this model, a networking system was divided into layers. Within each layer, one or more entities implement its functionality. Each entity interacted directly only with the layer immediately beneath it, and provided facilities for use by the layer above it.

Protocols enabled an entity in one host to interact with a corresponding entity at the same layer in another host.

Service

definitions abstractly described the functionality provided to an (N)-layer by an (N-1) layer, where N was one of the seven layers of protocols operating in the local host.

The OSI standards documents are available from the ITU-T as the X.200-series of recommendations. Some of the protocol specifications were also available as part of the ITU-T X series. The equivalent ISO and ISO/IEC standards for the OSI model were available from ISO, but only some of them without fees.

Introduction

The Open System Interconnection (OSI) reference model is a framework for defining the conventions and tasks required for network systems to communicate with one another. The work on the OSI model began in the late 1970s, mostly independently, by the International Organization for Standardization (ISO) and the International Telegraph and Telephone Consultative Committee or CCITT (which comes from the translation of the title in French). CCITT has been succeeded by the Telecommunications Standardization Sector of the International Telecommunications Union (ITU-TS). In 1983 the work of the two organizations was combined, and a single document describing the reference model for Open Systems Interconnection was produced. The term “open systems” refers to the fact that the specifications are publicly available to

everyone. The purpose of the OSI model was to assist vendors and communications software developers to produce interoperable network systems. Although the OSI model was designed to replace all previous computer communications standards, it is no longer viewed as such a replacement. Rather, the OSI model has succeeded as a tool for describing and defining how heterogeneous network systems communicate.

The OSI model is based on a widely accepted structuring technique called layering. According to this approach, the communications functions are partitioned into a vertical set of layers. Each layer performs a related set of functions, utilizing and enriching the services provided by the immediately lower layer. The layering approach was developed to address the following goals:

• Provide a logical decomposition of a complex communications network into smaller, more understandable and manageable parts.

• Provide standard interfaces between network functions and modules.

• Provide a standard language for describing network functions, to be used by network designers, managers, vendors, and users.

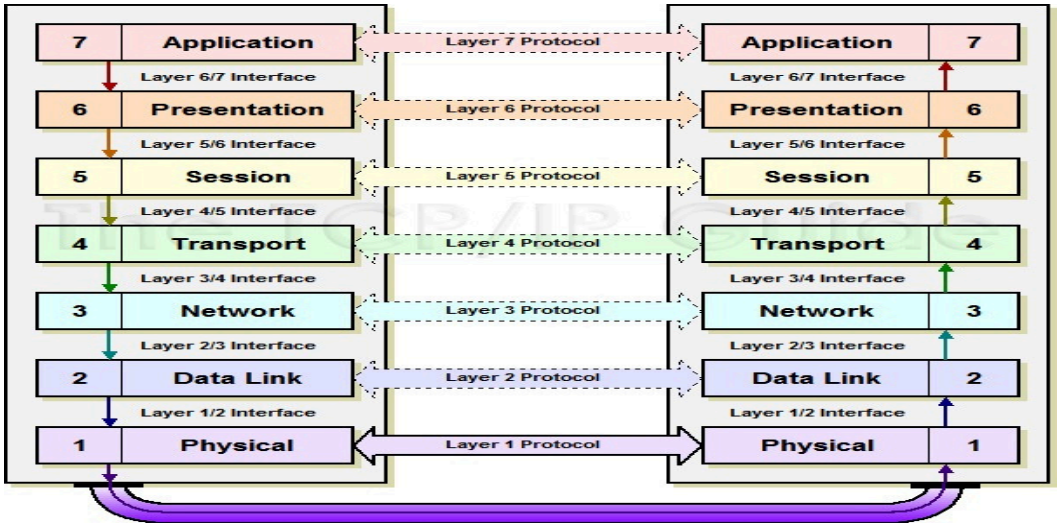
An important task in the development of the OSI model was to group similar functions into layers, while keeping each layer small enough to be manageable, and at the same time, keeping the number of layers small, since a large number of layers would increase the processing overhead. The principles used in defining the OSI layers are summarized in following list (Stallings, 1987):

1. The number of layers should not be so many as to make the task of describing and integrating the layers more difficult than necessary.
2. Layer boundaries should be created at points where the description of services is small and the number of interactions between boundaries is minimized.
3. Separate layers should be created in cases where manifestly different functions are performed or different technologies are involved.
4. Similar functions should be collected into the same layer.
5. A layer should be created where functions are easily localized. This enables the redesign of the layer to take advantage of new technologies.
6. A layer should be created where there is a need for a different level of abstraction in the handling of data.
7. Changes of functions or protocols of a layer should be made without affecting other layers.
8. For each layer, boundaries with its upper and lower layers only are created.

II. DESCRIPTION OF OSI LAYERS

The application of the above principles resulted in the seven-layer OSI reference model, which we describe next.

The recommendation X.200 describes seven layers, labeled 1 to 7. Layer 1 is the lower layer in this model.



At each level (*N*), two entities (layer N peers) exchange protocol data units (PDUs) by means of a layer-N protocol. A service data unit (SDU) is the payload of a PDU, transmitted unchanged to a peer.

The SDU is a unit of data that is passed down from one OSI layer to the next-lower layer, and which the lower layer encapsulates into a PDU. Layer N-1 adds a header or footer, or both, to the SDU, composing a PDU of layer N-1. The added framings make it possible to get the data from a source to a destination. The PDU at a layer N thus becomes the SDU of layer N-1.

Some orthogonal aspects, such as management and security, involve every layer.

Security services are not related to a specific layer: they can be related by several layers, as defined by ITU-T X.800 Recommendation.

These services are aimed to improve the CIA triad (confidentiality, integrity, and availability) of transmitted data. In practice, the availability of communication service is determined by the interaction between network design and management protocols. Appropriate choices for both of these are needed to protect against denial of service.

Layer 1: physical layer

The physical layer has the following major functions:

- It defines the electrical and physical specifications of the data connection. It defines the relationship between a device and a physical transmission medium (e.g., a copper or fiber optical cable). This includes the layout of pins, voltages, line impedance, cable specifications, signal timing, hubs, repeaters, network adapters, host bus adapters (HBA used in storage) and more.
- It defines the protocol to establish and terminate a connection between two directly connected nodes over a communications medium.
- It may define the protocol for flow control.
- It defines a protocol for the provision of a (not necessarily reliable) connection between two directly connected nodes, and the modulation or conversion between the representation of digital data in user equipment and the corresponding signals transmitted over the physical communications channel. This channel can involve physical cabling (such as copper and optical fiber) or a wireless radio link.

The physical layer of Parallel SCSI operates in this layer, as do the physical layers of Ethernet and other local-area networks, such as Token Ring, FDDI, ITU-T G.hn, and IEEE 802.11, as well as personal area networks such as Bluetooth and IEEE 802.15.4.

Layer 2: Data Link Layer

The data link layer provides reliable transmission of data (frames) between adjacent nodes, built on top of a raw and unreliable bit transmission service provided by the physical layer. To achieve this, the data link layer performs error detection and control, usually implemented with a Cyclic Redundancy Check (CRC). Note that the data link layer provides reliable transmission service over a single link connecting two systems. If the two end systems that communicate are not directly connected, then their communication will go through multiple data links, each operating independently. In this case, it is the responsibility of higher layers to provide reliable end-to-end transmission. Bridges, which connect two similar or dissimilar local area network segments, operate at this layer. Some well-known protocols for the data link layer include High-level Data Link Control (HDLC), LAN drivers and access methods such as Ethernet and Token Ring, and the LAP-D protocol in ISDN networks.

Layer 3: Network Layer

While the data link layer deals with the method in which the physical layer is used to transfer data, the network layer deals with organizing that data for transfer and reassembly. In short, the main function of this layer is Path determination and logical Addressing. This layer provides logical addresses to the packets received which in turn helps them to find their path.

“The network layer provides the functional and procedural means of transferring variable length data sequences (called datagrams) from one node to another connected to the same *network.* A network is a medium to which many nodes can be connected, on which every node has an *address* and which permits nodes connected to it to transfer messages to other nodes connected to it by merely providing the content of a message and the address of the destination node and letting the

network find the way to deliver ("route") the message to the destination node. In addition to message routing, the network may (or may not) implement message delivery by splitting the message into several fragments, delivering each fragment by a separate route and reassembling the fragments, report delivery errors, etc."

Layer 4: Transport Layer

The transport level provides end-to-end communication between processes executing on different machines. Although the services provided by a transport protocol are similar to those provided by a data link layer protocol, there are several important differences between the transport and lower layers:

1. ***User Oriented*****:** Application programmers interact directly with the transport layer, and from the programmers perspective, the transport layer is the ``network''. Thus, the transport layer should be oriented more towards user services than simply reflect what the underlying layers happen to provide. (Similar to the beautification principle in operating systems.)

2. Negotiation of Quality and Type of Services: The user and transport protocol may need to negotiate as to the quality or type of service to be provided. Examples? A user may want to negotiate such options as: throughput, delay, protection, priority, reliability, etc.

3. Guarantee Service: The transport layer may have to overcome service deficiencies of the lower layers (e.g. providing reliable service over an unreliable network layer). ***4. Addressing becomes a significant issue:*** That is, now the user must deal with it; before it was buried in lower levels.

Two solutions:

Use well-known addresses that rarely if ever change, allowing programs to ``wire in'' addresses. For what types of service
does this work? While this works for services that are well established (e.g., mail, or telnet), it doesn't allow a user to easily experiment with new services.
Use a name server. Servers register services with the name server, which clients contact to find the transport address of a
given service. In both cases, we need a mechanism for mapping high-level service names into low-level encoding that can be used
within packet headers of the network protocols. In its general
Form, the problem is quite complex. One simplification is to break the problem into two parts: have transport addresses

be
a combination of machine address and local process on that machine.

5. Storage capacity of the subne: Assumptions valid at the data link layer do not necessarily hold at the transport Layer. Specifically, the subnet may buffer messages for a potentially long time, and an ``old'' packet may arrive at a destination
at unexpected times.

6. We need a dynamic flow control mechanism: The data link layer solution of reallocating buffers is inappropriate because a machine may have hundreds of connections sharing a single physical link. In addition, appropriate settings for
the flow control parameters depend on the communicating end points (e.g., Cray supercomputers vs. PCs), not on the
protocol used.

7. Don't send data unless there is room**:** Also, the network layer/data link layer solution of simply not acknowledging frames for which the receiver has no space is unacceptable. Why? In the data link case, the line is not being used for anything else; thus retransmissions are inexpensive. At the transport level, end-to-end retransmissions are needed, which
wastes resources by sending the same packet over the same links multiple times. If the receiver has no buffer space, the
sender should be prevented from sending data. ***8. Deal with congestion control:*** In connectionless Internets, transport protocols must exercise congestion control. When
the network becomes congested, they must reduce rate at which they insert packets into the subnet, because the subnet has
no way to prevent itself from becoming overloaded. ***9. Connection establishment:*** Transport level protocols go through three phases: establishing, using, and terminating a
connection. For data gram-oriented protocols, opening a connection simply allocates and initializes data structures in the

Finally, although not as difficult as establishing a connection, terminating a connection presents subtleties too. For instance, both ends of the connection must be sure that all the data in their queues have been delivered to the remote application.

Layer 5: Session Layer

The session layer permits two parties to hold ongoing communications called a session across a network. The applications on either end of the session can exchange data or send packets to another for as long as the session lasts. The session layer handles session setup, data or message exchanges, and tears down when the session ends. It also monitors session identification so only designated parties can participate and security services to control access to session information. A session can be used to allow a user to log into a remote time-sharing system or transfer a file between two machines.

The session layer has the option of providing one-or-two-way communication called dialogue control. Sessions can allow traffic to go in both directions at the same time, or in only one direction at a time. Token management may be used to prevent both sides from attempting the same operation at the same time. To manage these activities, the session layer
provides tokens that can be exchanged. Only the side holding the token is permitted to perform the critical operation.

Another session service is synchronization. Consider the problems that occur when transferring a file between two machines and the system crashes not being able to complete the transfer. This process must be restarted from the beginning. To avoid this problem, the session layer provides a way to insert checkpoints into the data stream, so that after a crash, only the data after the last checkpoint has to be repeated. It accepts the data from presentation layer and provides
services to it and accepts the services of the transport layer. The name of data unit in the session layer is *SPDU* (Session Protocol Data Unit) or sessions.

Therefore session layer functionality includes:

a) Virtual connection between application entities

b) Synchronization of data flow

c) Creation of dialog units

d) Connection parameter negotiations

e) Partitioning of services into functional groups.

f) Acknowledgments of data received during a session

g) Retransmission of data if it is not received by a device

Layer 6: Presentation Layer

The presentation layer formats the data to be presented to the application layer. It can be viewed as the translator for the network. This layer may translate data from a format used by the application layer into a common format at the sending station, and then translate the common format to a format known to the application layer at the receiving station. The presentation layer provides:

- Character code translation: for example, ASCII to EBCDIC.
- Data conversion: bit order, CR-CR/LF, integer-floating point, and so on.
- Data compression: reduces the number of bits that need to be transmitted on the network.
- Data encryption: encrypt data for security purposes. For example, password encryption.

Layer 7: Application Layer

This is the level that the user often interacts with. This is where data turns into websites, chat programs and so on. Many protocols run at this layer, such as DNS, FTP, HTTP, HTTPS, NFS, POP3, SMTP, and SSH. “This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services.”

III. BENEFITS OF THE OSI MODEL

"By separating the network communications into logical smaller pieces, the OSI model simplifies how network protocols are designed. The OSI model was designed to ensure different types of equipment (such as network adapters, hubs, and routers) would all be compatible even if built by different manufacturers."

The OSI model has many benefits which include:

a. Compatibility: The OSI model can fit to any compatible software/hardware from different users in other parts of the world. As software/hardware differs among various users so OSI is a model that is compatible to all.

b. Easy Troubleshooting: Since each layer in an OSI is independent of each other so it makes it easier to detect and solve all errors prevailing in it.

c. Easy Understanding Nature: OSI model is very interactive and even guides us to know what a Model is, how it operates, and common methodologies, how new technologies are developed in existing networks.

d. Security: OSI model have functionality for Encryption and Decryption which has a major contribution for security purpose. This makes it Reliable.

E. Add Multiple Network Models: The OSI model is designed in such a way that user can further extend.

IV. CONCLUSION

In this paper we have tried to explain what exactly an OSI reference model is, why it is used and contribution of various researchers in this reference. OSI is basically an architecture which only gives us an idea how packets transfer over the network during any communication. OSI enhancements are done time to time for developing new technologies. Scheidell et al., proposed three different layers in his paper for improvising security in any network. Future implementation in OSI will lead to enhancement in security and many other fields.

ACKNOWLEDGEMENT

My heartful gratitude to almighty go and my parents and teachers without whose unsustained support, I could not have completed this research paper.