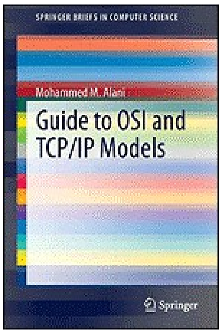


SPRINGER BRIEFS IN COMPUTER SCIENCE

Mohammed M. Alani

Guide to OSI and TCP/IP Models

Springer

Guide to OSI and TCP/IP Models

by Mohammed M. Alani

Chapter 2: OSI Model

**M. M. Alani, *Guide to OSI and TCP/IP Models*, SpringerBriefs in Computer Science,
DOI: 10.1007/978-3-319-05152-9_2, © The Author(s) 2014**

Overview

Abstract This chapter starts with a brief history of the OSI model and how it all started in the mid 1970s. Afterwards, the OSI model is explained in details along with the functions and duties of each layer in the model. Studying the OSI model is a simple first step into the networking world. At the end of the chapter, the encapsulation and decapsulation processes are introduced such that the reader would understand the end-to-end data flow from one host to another.

Keywords OSI • ISO • Seven layers • Physical layer • Datalink • Network • Transport • Session • Presentation layer • Application layer

2.1 History of OSI Model

The OSI model was officially adapted as a standard by ISO in 1979. Some might say that it is an old standard. Well, it is old. What kept this model alive for so long is its capacity of expansion to meet the evolving needs.

Most of the work that created the base for the OSI model was done by a group at Honeywell Information Systems. The head of this group was Mike Canepa. This group started addressing the lack of standardization problem in the mid 1970s of the past century and they came up with a proposal named Distributed Systems Architecture, DSA. By that time, the British Standards Institute submitted a proposal to the ISO saying that there is a need for unified standard communication architecture for distributed processing systems. Responding to this proposal, the ISO formed a subcommittee on Open System Interconnection. The ISO also made American National Standards Institution (ANSI) in-charge of preparing proposals prior to the first official meeting of the subcommittee. Canepa's group participated in the ANSI meetings to discuss their seven-layer proposal. Later, ANSI chose to provide Canepa's proposal as the only one to be submitted to the ISO subcommittee.

In March 1978, the first meeting of the subcommittee was made and Canepa and his team presented their proposal there. The ISO group thought that this proposal covered most of the needs for Open System Interconnection. In the same month that year a provisional version of the model was published. With some minor improvements, the next version of the model was published in June 1979 and was standardized.

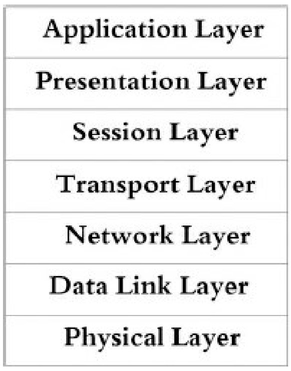
In 1995 the OSI model was revised to cover the needs arising by the rapid development in the field of computer networks [1].

2.2 OSI Layers

The ISO OSI model consists of seven layers. Figure 2.1 shows these layers. Usually, the routers and other network devices act in the bottom three layers and the hosts act in the whole seven layers.

Each layer handles the data in a way that is different from other layers. The unit in which a certain layer handles data is called a Protocol Data Unit (PDU). Some layers add layer-specific information to the data. This information added by the layers' protocols can be in the form of a header, a trailer, or both. The header information is added at the start of the PDU, while the trailer information is added at the end of the PDU. This header or trailer contains information that is useful in controlling the communication between two entities.

The OSI model works in a peer-layer strategy. This strategy implies that the control information added to the PDU by one layer is meant to reach the peer layer in the receiving entity. For example, the header information added at the network layer in the sender host is used by the network layer in the receiving host and this information is insignificant to other layers. Hence, compatible protocols must be used at both ends of the communication to succeed in delivering the user data in the right manner.



Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Fig. 2.1: The OSI-model seven layers

Before going into a brief description of each layer, we need to add two new concepts. These concepts are the modes of data transfer; *connection-oriented* and *connectionless*. In connection-oriented communication, a connection needs to be established before the start of transmitting data from the sender to the receiver. This is analogous to a phone call. You cannot start talking to the other side of the phone before a connection between you and them is established and the other end actually picks up the phone. Connectionless communications refer to the type of communication in which there is no connection establishment before the transmission of data takes place. Control information is added to the data and the data is then sent to the destination and you cannot tell, in an easy way, whether the receiver has received the data correctly or not. This is analogous to sending a written message by mail. All you can do is write the address on the message and drop it at the post office.

The idea of the model system is not to tell you how the network actually operates, but to define the elements and functions that compose a network in a way that makes these elements and functions distinct and distributable on layers. This distinction provides the ability to protocols to operate in a smooth way and to be easy to troubleshoot.

In the following subsections, we will discuss each layer's functions and how each layer handles the data. In the next section, we will go through the complete cycles of data from source host to destination host [2].

2.2.1 Physical Layer

The physical layer basically handles data as raw bits. This means that the PDU for the physical layer is a *bit*. The primitive duty of the physical layer is to provide transparent transmission of bits from the data-link layer of the sender to the data-link layer of the receiver. This is accomplished by defining the mechanical, electrical, functional and procedural means to activate, maintain, and deactivate a physical link between two data-link entities.

Beside the data transmitted from one physical entity to another, control information needs to be transferred too. This control information may be added to the data and transformed in the same channel in which the data is transferred, and this is called *in-line signaling*. Or, the control information may be transferred through a separate control channel, which is called *off-line signaling* or *out-of-line signaling*. The choice of which way to transfer the control information is left to the protocol used.

Physical layer protocols vary depending on the type of the physical medium and the type of the signal carried on it. The signal can be an electrical voltage carried over a cable, a light signal carried through a fiber link, or even an electromagnetic signal carried in the air on in the outer space.

The main functions of the physical layer are:

a. Physical connection activation and deactivation.

   The physical connection activation and deactivation is done upon request from the data-link layer.

b. PDU Transmission.

   As we have mentioned before, the physical layer PDU is bit. So, transmission of bits from the source to the destination is a physical layer function.

c. Multiplexing and demultiplexing (if needed).

   There are many cases in which two or more connections need to share the same physical channel. In this case, multiplexing these connections into the channel is required at the sender side, and demultiplexing is required at the receiver side. This function is usually done through a specialized data-circuit, and is optional in the OSI standard.

d. Sequencing.

   The physical layer must make sure that the transmitted bits arrive in the same sequence in which they were sent from the data-link layer.

e. Physical layer management.

   Some layer management aspects are left to the protocol and used medium, such as error detection. These management functions depend on the protocol and physical medium. For example, the electrical signal transmitted through a metallic wire needs different management than the optical signal transmitted through a fiber cable.

2.2.2 Data-Link Layer

The PDU of the data-link layer is a *frame*, which means, the data-link layer handles data as frames. These frames may range from few hundred bytes to few thousand bytes. The data-link layer adds its control information in the form of a header and a trailer.

Data-link layer has many complex functions as compared to other layers. Data-link layer provides different type of functions for connection-oriented and connectionless communications. Actually, all the functions provided to the connectionless communication are provided to the connection oriented, but the opposite is not true. The following is a list of functions provided for both connection-oriented and connectionless communications:

a. Control of data-circuit interconnection

   This function gives network entities the capability of controlling the interconnections of data-circuits within the physical layer.

b. Identification and parameter exchange

   Each entity needs to identify itself to other entities and some parameters governing the communication need to be exchanged, too. An example of these parameters is data rate.

c. Error detection

   Some physical channels might be susceptible to factors that prevent the data from being delivered in the right way. These factors can be Electro-Magnetic Interference (EMI), temperature, rain, etc., depending on the medium type. One of the data-link layer functions is to detect these errors.

d. Relaying

   Some network configurations require relaying between individual local networks.

e. Data-link layer management

   Similar to the physical layer management, the data-link layer leaves some management operations to the protocols used.

In addition to the functions listed above, the data-link layer provides the following functions only to the connection-oriented communications:

a. Data-link connection establishment and release

   As the name indicates, this function is responsible for the establishment and release of the data-link connections between communicating entities.

b. Connection-mode data transmission

   Connection-oriented communication requires certain mechanisms in order to assure the delivery of data. For example, in connection-oriented communication, for each transmitted frame, or group of frames, an acknowledgement frame is transmitted back from the receiver to the sender to acknowledge the reception of the frame or frames.

c. Data-link-connection splitting

   This function is aimed to split the data-link connection into multiple physical connections, if possible.

d. Sequence control

   This function assures that the data frames are received in the same order in which they were sent or at least assure that the frames can be re-arranged in the right order if they arrive out of order.

e. Framing (delimiting and synchronization)

   This function provides the recognition of a sequence of bits, transmitted over a physical connection, as a data-link frame.

f. Flow control

   In connection-oriented communication, the sender and receiver can dynamically control the rate in which the data is transferred. In connectionless communication, there is service boundary flow control but no peer flow control. This means that in connectionless communication, there is a limit imposed by the physical medium and physical layer protocol to the flow, but the rate can be controlled by the communicating entities.

g. Error recovery or error correction

   This function tries to correct the detected error based on mechanisms used by the data-link protocol. In connectionless communication the data-link layer can only detect errors, but not correct them. This function tries to correct the error, and if it fails, it informs the network entities of that error to perform retransmission.

h. Reset

   This function forces the data-link connection to reset.

2.2.3 Network Layer

The PDU for the network layer is a *packet*. The network layer handles very crucial duties regarding the routing of data from one network to another and controlling the subnet. Routing can be a complex operation in some times as many factors contribute in the choice of the best route from a source to a destination. The following is a list of the network layer functions:

a. Routing and relaying

   Routing is the operation of selecting the best path for data from source to destination and sending the data along that path.

b. Network connection and multiplexing

   This function provides network connections between transport-layer entities by employing the data-link connections available. Sometimes multiplexing is needed to optimize the use of these data-link connections by sending more than a single network connection through the same data-link connection.

c. Segmentation and blocking

   The network layer may segment and/or block the PDUs in order to facilitate the transfer. Segmentation, or sometimes referred to as Fragmentation, is basically making the PDUs smaller. This is an important function if the data is passed between networks that are using different data-link layer standards like Ethernet and Asynchronous Transfer Mode (ATM). These different data-link standards can have different maximum packet size. And thus causing the PDU of one data-link protocol incompatible with another data-link protocol.

d. Error detection and recovery

   In order to check that the quality of service provided over a network connection is maintained, error detection function is required. Network layer uses error notifications from the data-link layer and additional error detection mechanisms. The error recovery is also essential to try to correct the detected errors.

e. Sequencing and flow control

   Sequencing is used to maintain the sequential order of the packets sent to a destination upon the request of the transport layer. Flow control is used for prevent flooding the destination with data, and control the transmission rate.

f. Expedited data transfer

This function provides expedited data transfer from source to destination, if required.

g. Reset

   Reset the network connection.

h. Service selection

   This function assures the use of the same service at the source and destination as the packets pass through different subnetworks with different quality levels.

i. Network-address-to-datalink-address mapping

   This mapping is important to facilitate the proper transfer of data from the source to the networking devices and from the networking devices to the destination, and back.

j. Network layer management

   This function is to manage the network layer functions and services.

The network layer requires some facilities to be able to accomplish its functions. The facilities can guarantee the smooth operation of the layer. The most important of these facilities are network addressing (sometimes referred to as logical addressing), quality of service parameters, expedited PDU transfer, and error notification.

Network addressing gives a unique address to each host, thus, it is consistent. Hosts lying in the same subnetwork have to have a common portion of their addresses which is called a network address or a subnet address. To understand this easier, think of calling a phone number in another country. The format of the phone number is country-code—area-code—subscriber code. For example, the telephone number of a person in Virginia, USA should be in the format +1-703-XXXXXXXX. The first part, (1), identifies the country; USA. The second part, (703), identifies the area; Virginia. The third part identifies the subscriber within Virginia, USA. This hierarchy in telephone numbers is analogous to network addressing. A common part identifies the network, or subnet, address, and a unique part that identifies the particular host.

The quality-of-service (QoS) parameters define the quality limits of the network connection. Most known QoS parameters are delay, jitter, service availability, reliability, and network-connection establishment delay. These parameters are beyond the scope of this brief.

Error notification might lead to the release of the network connection, but not always. This depends on the specifications of the particular network service in which the error has been detects. Unrecoverable errors detected by the network layer are reported to the transport entities.

2.2.4 Transport Layer

Since there are two types of services that can be provided to the networking applications, connection-oriented and connectionless, the transport layer provides different kind of functions for these two types. The PDU for the transport layer is a segment. The functions of the transport layer in a connection-oriented communication are listed in the following:

a. Establishment and release of transport connections

   This function is responsible for initiating the connection between the communicating entities and releasing the connection when the data transfer is over.

b. Sequence control

   Controlling the sequence of data transferred to guarantee that the data arrive in the same sequence in which it was sent.

c. End-to-end error detection and recovery

   This function provides detection of errors occurring in segments and trying to recover these errors to their original error-free form.

d. Segmentation

   At the transport layer, the data is transformed into segments at the sender and reconstructed at the recipient.

e. End-to-end flow control

   This function controls the rate in which segments are transferred from one entity to another.

f. Monitoring QoS parameters

   This function provides the transport layer the ability to monitor the QoS parameters of the communication.

For connectionless communications, the functions of the transport layer are:

a. End-to-end error detection

   In connectionless communications, the transport layer only detects the errors and notify the session entities, but does not try to recover them.

b. Monitoring of QoS parameters

   Connectionless communications can also be monitored in terms of QoS parameters.

c. PDU delimiting

   This function introduces the ability to delimit the PDUs to maintain the continuity of communication.

The transport layer gives a great support to the session layer in terms of providing the mechanisms to differentiate which data goes to what session.

The connection-oriented communication goes into three phases in the transport layer; establishment, data transfer, and release. During the establishment phase, the transport layer sets the parameters of end-to-end communication. For example, multiplexing of sessions into a network-connection, optimum segment size, and obtaining a network connection that matches the needs of the session entities. After establishing the connection, the data transfer starts and uses error detection and correction, sequencing, segmentation, and flow control mechanisms. When the data is transferred completely, the sender notifies the recipient of the request to release the connection, and the connection is released.

2.2.5 Session Layer

The session layer does not have a PDU of its own. It handles the data in the shape they come in, without division or concatenation. Its basic purpose is to provide the ability to the presentation entities to organize the communication for multiple communication sessions taking place at the same time.

The main functions of the session layer are:

a. Session initiation and teardown

   The session layer is responsible for starting the sessions between the communicating entities. And when the session is over, the session layer is responsible for the release of the communication. Data transfer takes place in-between.

b. Token management

   This function is related to the communication mode used in the specific session (simplex, half-, or full-duplex). The session layer controls which entity own the token and can transmit data at this time. The token is the license to transmit in an environment or service where only one entity can transmit at a time. This is not the case for all applications. Some applications operate in full-duplex mode, others operate in half-duplex mode.

c. Session-connection to transport-connection mapping (in connection-oriented transfer only)

   The function provides the session layer the ability to map between the transport layer connections and the sessions currently taking place. This way the session layer can tell which data goes to what session.

2.2.6 Presentation Layer

The presentation layer, like the session layer, does not have a PDU of its own. The presentation layer, as its name indicates, is responsible for the way data is presented to the application. At the start of the communication, the presentation layer negotiates the form of data to be transferred with the other entity, the *transfer syntax*. After this negotiation, the presentation layer can provide additional services such as compression, encryption, and translation. The choice of which service(s) to be used is up to the application itself.

2.2.7 Application Layer

The application layer is responsible for defining the services presented at the user-end. Application layer protocols vary according to the specific type of data the user wants to transfer. Application layer also defines the acceptable QoS parameters for each service. For example, the voice data transmission requires different QoS parameters than transferring an email. The choice of which security aspects to use, such as authentication and access control, is up to the application layer. Also the synchronization of communicating applications in connection-oriented services is the application layer's responsibility.

Generally, the main functions of the application layer are:

a. identification of services provided to the user

b. defining QoS parameters required by the application

c. defining security mechanisms to be employed such as access control and authentication

d. synchronization of communicating applications (only in connection-oriented services).

2.3 End-to-End Data Flow

The flow of data from the application layer to the physical layer is called *encapsulation*. This is because header and trailer information is added to the data in various layers at the end and start of data, which makes it look like a capsule. The flow of data in the opposite direction, from the physical layer to the application layer, is called *decapsulation*, as it involves the removal of the headers and trailers such that the data is back to its original form to the receiver's user-end. Figure 2.2 shows the details of the encapsulation process.

Figure 2.3 shows the process of transferring data from on host to another. Since the OSI principle is to transfer data between different networks, not transferring data within the same network, we added a router in the middle between the two communicating hosts. In real life, there can be more than one router, depending on the specific networks we are dealing with. All routers act in a fairly similar way with the data received. So, we can replace the router in the middle with a series of routers.

2.3.1 Host A (Source)

1. The transmission starts from the application layer at Host A. The application decides that it needs to communicate with Host B and passes the data down to the presentation layer.

2. The presentation layer does the required transformations that need to be done on data, like compression, encryption, or translation. Then the data is passed down to the session layer.

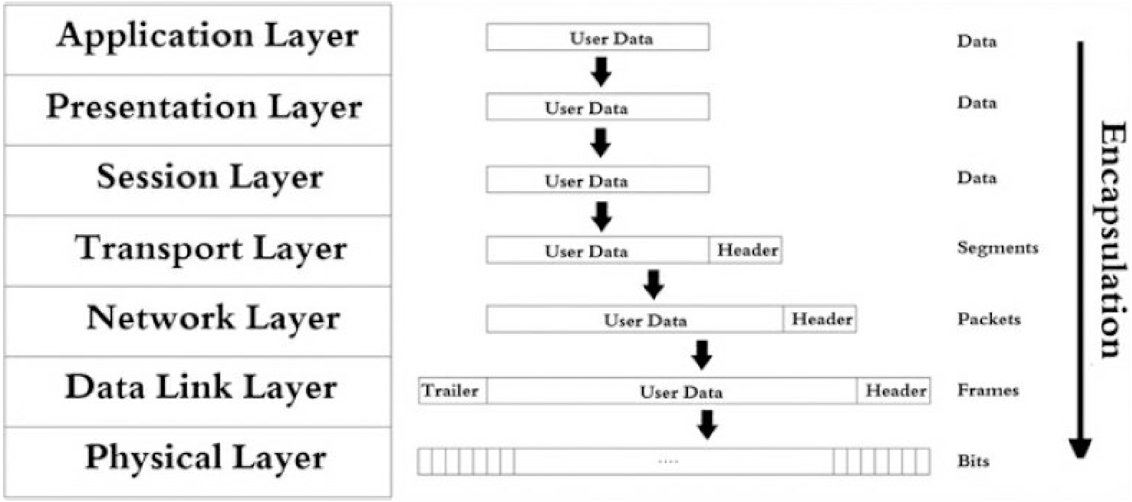


Fig. 2.2: Data encapsulation process in OSI model

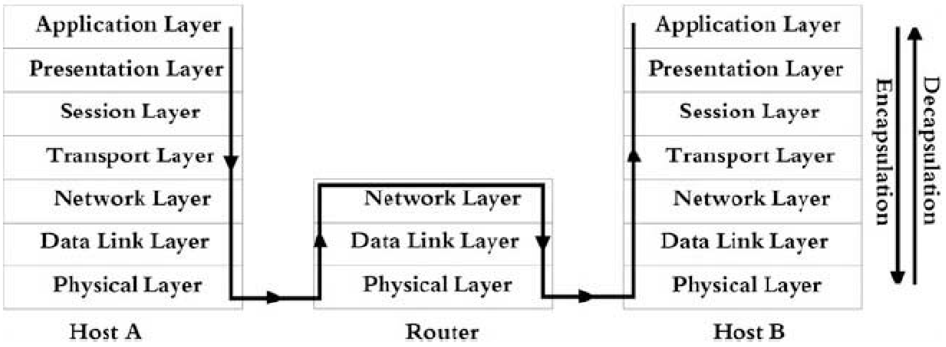


Fig. 2.3: End-to-end data flow in OSI model

3. The session layer starts initiating the communication session and passes the data to the transport layer.
4. At the transport layer, the data is segmented and a header is added to each segment of the data. This header contains transport control information such as the sequence number and acknowledgement number. The segment along with its header is passed down to the network layer.
5. The network layer deals with the whole segment, including its header, as data. The header added by the transport layer is meant to be read by the transport layer at the receiving end. So, the network layer does not read the segment header, but instead, it handles the segment and its header as a single data element. The data is then put into packets and headers are added to these packets. The network layer header contains information meant to reach the network layer on the other end. This information includes a source and destination network-layer address along with few other parameters. These packets are sent down to the data-link layer.
6. The data-link layer handles the packet and its header as a single data element, such that the network layer header is considered part of the data. The data-link layer puts the data into frames and adds a header and a trailer to each frame. The data-link header contains control information sent to the data-link layer on the other end, while the trailer usually contains error control information.
7. The frames are then sent down to the physical layer where they are dealt with as raw bits. These bits are transferred through the physical channel to the router.

2.3.2 The Router

The router does a partial decapsulation because it does not need to read the data all the way up to its application-layer shape. The router needs only to read up to the network layer header to route the data to the wanted destination.

1. The process starts at the physical layer when the data is received as raw bits. These bits are then gathered into frames and sent over to the data-link layer.
2. The data link layer reads the header and the trailer to know what to do with the data. The data-link layer then rips off the header and the trailer and sends the rest of the data as a packet to the network layer.
3. At the network layer, the network header is read to determine the destination network address. After knowing the destination network address, the router chooses the best route to send the data to the destination host.
4. Starting from the network layer, the encapsulation starts again at the router. Data goes down the data-link and physical layers and all the way through the physical link to the destination host.

If there is another router in the way, the same process that took place in the first router is repeated until reaching the destination host.

2.3.3 Host B (Destination)

1. At the destination host, the raw bits are elevated as frames to the data-link layer.
2. The header and trailer of each frame are read by the data-link layer and then removed. The rest of the data is elevated to the network layer as packets.

3. The header of the packet is read to determine if this is the correct destination for the packet and other network control information is also taken from the network layer header. The header is then removed and the rest of the data is elevated to the transport layer as segments.

4. The header of each segment is read to determine the sequence number and arrange the segments in their correct order. Then, the header is also removed and the rest of the data is elevated to the session layer. The transport header also contains information of which session this data is going to. This information is passed to the session layer with the data.

5. The session layer determines if this is the end of this session or not. If it is the last segment in the session, the session layer will wait for the request to end this session. If this is not the last segment in the session, the session layer waits for more data.

6. The data is then passed to the presentation layer to retransform the data into the shape that they were sent in by the sending-end application or to another form determined by the application. This might involve decompression, decryption, or translation.

7. The data is then transferred to the application and received by the user.

One important thing to remember is that each layer, using its header, trailer, or connection-setup parameters, communicates with the peer layer at the receiving end.

References

1. Stallings, W.: *The Origins of OSI* [Online] (1998). *http://www.williamstallings.com/Extras/OSI.html*

2. ISO: *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. *Geneva, Standard ISO/IEC 7498-1(E)* (1994)