A Comparative Evaluation of OSI and TCP/IP Models

P. Ravali

Department of Computer Science and Engineering, Amrita Vishwa Vidhyapeetham, Bengaluru

Abstract: ***Networking can be done in a layered manner. To reduce design complexities, network designers organize protocols. Every layer follows a protocol to communicate with the client and the server end systems. There is a piece of layer n in each of the network entities. These pieces communicate with each other by exchanging messages. These messages are called as layer-n protocol data units [n-PDU]. All the processes required for effective communication are addressed and are divided into logical groups called layers. When a communication system is designed in this manner, it is known as layered architecture. The OSI model is a set of guidelines that network designers used to create and implement application that run on a network. It also provides a framework for creating and implementing networking standards, devices, and internetworking schemes. This paper explains the differences between the TCP/IP Model and OSI Reference Model, which comprises of seven layers and five different layers respectively. Each layer has its own responsibilities. The TCP/IP reference model is a solid foundation for all of the communication tasks on the Internet.***

Keywords: TCP/IP, OSI, Networking

1. Introduction

A collection of autonomous computers interconnected by a single technology is called as computer networks. Computer devices use communication languages known as network protocols. These protocols describe how a computer communicates with other computers at the bit and byte level. It tells us the rules and conventions between network devices. Protocols generally use packet switching techniques to send and receive data in the form of packets [1].

Network protocols include mechanisms for devices to identify and make connections with each other, as well as formatting rules that specify how data is packaged into messages sent and received. Some protocols also support message acknowledgement and data compression designed for reliable and/or high-performance network communication. These protocols are layered on top of each other.

At the time the Internet was developed, layering had proven to be a successful design approach for both compiler and operating system [3]. This gave rise to the concept of layered protocols which nowadays form the basis of protocol design [4].The purpose of each layer is to provide services to the higher layers. Each layer acts as a virtual machine to the layer above it. This concept can be commonly known as information hiding, abstract data types, data encapsulation, object-oriented programming. The rudimentary idea is to provide service to its users but keep the details of its internal state and algorithms hidden.

A network protocol is necessary since it allows two data communication devices to communicate with each other. Communication systems establish communications by sending and receiving data. In general, much of the following should be undertaken:

- Data formats for data exchange where digital bit strings are exchanged.
- Address mapping.
- Address formats for data exchange to identify both senders and receivers address.
- Routing using internetworking.
- Detection of transmission errors is necessary on networks which cannot assure error-free operation.
- Loss of information-timeouts and retries.
- Sequence control when long bit stings are lost, delayed or chosen different routes.
- Flow control when sender's transmission is faster than receiver's.
- Direction of information flow needs to be addressed if transmission flows in only one direction.
- Acknowledgements for correct reception of packets for connection-oriented communication.

To implement a networking protocol, the protocol software modules are interfaced with a framework implemented on the machine's operating system [2]. Systems typically do not use a single protocol to handle a transmission. Instead they use a set of cooperating protocols, sometimes called a protocol family or protocol suite. Some of the best known protocols are: IPX/SPX, AX.25, AppleTalk and TCP/IP.

Protocols are layered in modern designs. Layering is a design principle which divides the protocol design into number of smaller parts, each of which accomplishes a particular sub-task, and interacts with the other parts of the protocol only in a small number of well-defined ways [5]. The advantages of layered protocols is that the methods of passing information from one layer to another are specified clearly as part of the protocol suite, and changes within a protocol layer are prevented from affecting the other layers. Since they are divided into different functional layers, assigning protocols to perform each layers task makes it simpler. Examples of layered protocols are TCP/IP's five layer protocol stack and the OSI seven layer models. Violating the protocol will make communication difficult.

A five-layer network is illustrated below. The entries constitute the corresponding layers on different machines are called peers. The peers may be processes or hardware devices who communicate using the protocol. No data is directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Bottom layer1 is the physical medium through which communication comes about.
In figure 1, virtual communication is shown by dotted lines and physical communication by solid lines [8]. Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one. One of the most essential considerations is defining clean interfaces between the layers.

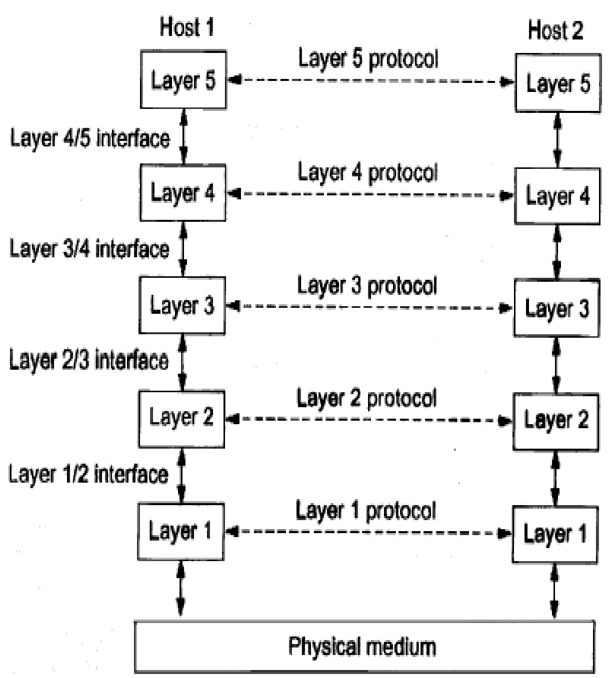

Figure 1: Layers, Protocols and Interfaces [7]

It is required that each layer performs a specific collection of well-understood functions thereby minimizing the amount of information. It is simpler to replace one layer with a completely different implementation since the new implementation offers the same set of services to the upstairs neighbor as the old implementation.

A set of layers and protocols is called network architecture. The details of the implementation and the specification of the interfaces are not a part of architecture as these are hidden away inside the machines and are not visible from outside. A list of protocols used by a certain system, one protocol per layer, is called a protocol stack. Two important network architectures are the OSI reference model and the TCP/IP reference model. Section 2 gives a brief description of the OSI reference model. Section 3 describes about the TCP/IP reference model. Section 4 will distinguish the two of the models and Section 5 will have the conclusion.

2. OSI Reference Model

The Open System Interconnection Model was introduced by ISO (International Organization for Standardization) in 1984. The system summarizes sophisticated network phenomenon and cases on seven layers. The internal functions of a communication system is characterized and standardized by partitioning into abstraction layers. The model groups' communication functions into seven logical layers. A layer serves the layer above it and is served by the layer below it. The principles applied to reach at the seven layers are:

- Each layer should perform a well-defined function.
- A layer should be created where a different abstraction is needed.
- The function of each layer should define internationally standardized protocols.
- The layer boundaries should minimize the information flow across the interfaces.
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy [9].

This model was developed to simplify network complexity, facilitate network training and introduce easy network troubleshooting [10]. The layers of OSI model are described in figure 2 below:

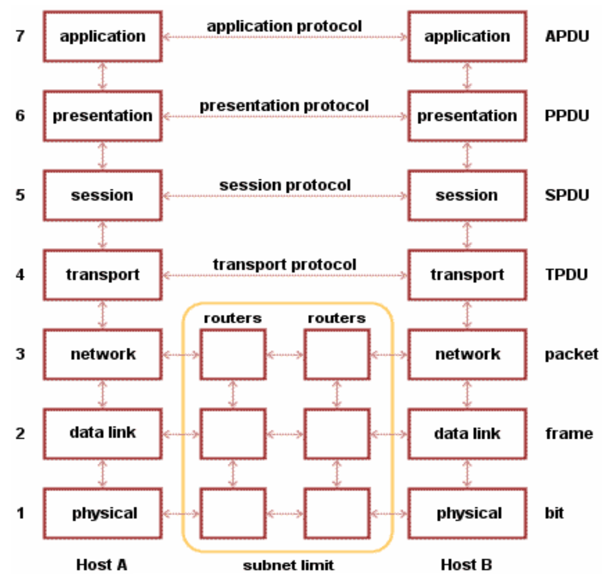

Figure 2: OSI Reference Model

3. The Physical Layer

The bottom layer of the OSI Model is the Physical Layer. It addresses the physical characteristics of the network, such as the types of cables used to connect devices, the types of connectors used, how long the cables can be, and so on [11]. The physical layer is responsible for transmitting raw bits over a communication channel. It defines the electrical and physical specifications of the data connection. It also talks about the relationship between a device and a physical transmission medium (Ex: A copper or fiber optical cable). The design issues have to make sure that one side sends a 1 bit; it is received by the other side as a 1 bit but not as a 0 bit.

Another aspect of the Physical Layer is that it specifies the electrical characteristics of the signals used to transmit data over cables from one network node to another. The Physical Layer doesn't define any particular meaning for those signals other than the basic binary values 0 and 1. The higher levels of the OSI model must assign meanings to the bits transmitted at the Physical Layer.

4. The Data Link Layer

The Data Link Layer is the lowest layer at which meaning is assigned to the bits that are transmitted over the network. Data-link protocols address things, such as the size of each packet of data to be sent, a means of addressing each packet so that it's delivered to the intended recipient, and a way to ensure that two or more nodes don't try to transmit data on the network at the same time.
The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It provides a reliable link between two directly connected nodes by detecting and possibly connecting errors that may occur in the physical layer.
The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.

It accomplishes the task by having the sender break up the input data into data frames and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgment frame. Layer 2 Data Link examples include PPP, FDDI, ATM, IEEE 802.5/ 802.2, IEEE 802.3/802.2, HDLC and Frame Relay.

5. The Network Layer

The Network Layer handles the task of routing network messages from one computer to another. It controls the operation of subnet. One important function of the Network Layer is logical addressing. Every network device has a physical address called a MAC address, which is assigned to the device at the factory. A logical address gives a network device a place where it can be accessed on the network using an address. This is created by IP or IPX, which are the network layers. The Network Layer protocol translates logical addresses to MAC addresses.
Another important function of the Network layer is routing, finding an appropriate path through the network. Routing comes into play when a computer on one network needs to send a packet to a computer on another network. In this case, a Network Layer device called a router forwards the packet to the destination network. An important feature of routers is that they can be used to connect networks that use different Layer-2 protocols. For example, a router can be used to connect a local-area network that uses Ethernet to a wide-area network that runs on a different set of low-level protocols [12].

6. The Transport Layer

The Transport Layer is the basic layer at which one network computer communicates with another network computer. This layer provides transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer. It also determines what type of service to provide to the session layer and ultimately to the users of the network.

The transport layer ensures that messages are delivered error-free, in sequence, and with no losses or duplications. It relieves the higher layer protocols from any concern with the transfer of data between them and their peers. The size and complexity of a transport protocol depends on the type of service it can get from the network layer. For a reliable network layer with virtual circuit capability, a minimal transport layer is required. If the network layer is unreliable and/or only supports data grams, the transport protocol should include extensive error detection and recovery.

In many cases, the Transport Layer protocol divides large messages into smaller packets that can be sent over the network efficiently. The Transport Layer protocol reassembles the message on the receiving end, making sure that all packets contained in a single transmission are received and no data is lost.

7. The Session Layer

The Session Layer establishes sessions between network nodes. A session must be established before data can be transmitted over the network. The session layer makes sure that these sessions are properly established and maintained. This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.

8. The Presentation Layer

The Presentation Layer establishes a context between application-layer entities, in which the application-layer entities may use different syntax and semantics if the presentation service provides a big mapping between them. If a mapping is available, presentation service data units are encapsulated into session protocol data units, and passed down the protocol stack [13]. This layer is usually a part of an operating system and converts incoming and outgoing data from one presentation format to another (for example, from clear text to encrypted text at one end and back to clear text at the other) [14]. This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.

9. The Application Layer

The application layer is the OSI layer closest to the end user, which means both the OSI application layer and the user interact directly with the software application. This layer interacts with software applications that implement a communicating component. The application layer contains a variety of protocols that are commonly needed by the users. 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. One-widely used application protocol is HTTP (Hyper Text Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail and network news. Telnet and FTP are applications that exist entirely in the application level. Tiered application architectures are part of this layer [15].

10. TCP/IP Model

TCP / IP stands for Transmission Control Protocol / Internet Protocol. It defines how electronic devices (like computers) should be connected over the Internet, and how data should be transmitted between them. TCP is responsible for breaking data down into small packets before they can be set over a network, and for assembling the packets again when they arrive. IP takes care of the communication between computers. It is responsible for addressing, sending and receiving the data packets over the Internet. TCP provides reliable, ordered and error-checked delivery of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the transport layer.

The APPARNET was a research network sponsored by the DoD (U.S. Department of Defense) which eventually connected hundreds of universities and government installations, using leased telephone lines. Later when satellites and radio networks were added, the existing protocols had trouble interworking with them, so new reference architecture was developed. The main goal was to connect multiple networks in a seamless manner. This architecture later became to be known as TCP/IP Model.

Another major goal was that the network be able to survive loss of subnet hardware with existing conversations not being broken off [16]. In other words, DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Architecture with divergent requirements was needed, ranging from transferring files to real-time speech transmission. The relationship between IP and TCP is essential. Each protocol has a special role: IP points to the path for the packets, while TCP provides a reliable transport.

TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination [17].

Each layer in this model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) model. The layers of this model are given in figure 3.

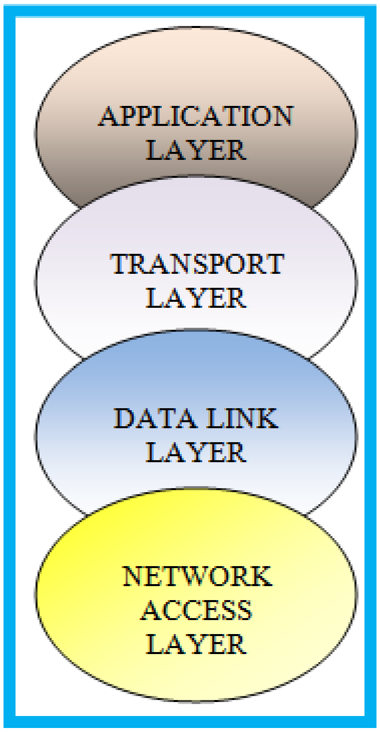

Figure 3: TCP/IP Model [18].

11. The Network Access Layer

Network Access Layer is the first layer of the four layer TCP/IP model. Network Access Layer defines details of how data is physically sent through the network, including how bits are electrically or optically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted pair copper wire [19]. This layer is responsible for placing TCP/IP packets on the network medium and receiving TCP/IP packets off the network medium. TCP/IP was designed to be independent of the network access method, frame format, and medium. In this way, TCP/IP can be used to connect differing network types. These include LAN technologies such as Ethernet and Token Ring and WAN technologies such as X.25 and Frame Relay. Independence from any specific network technology gives TCP/IP the ability to be adapted to new technologies such as Asynchronous Transfer Mode (ATM).

The Network Interface layer encompasses the Data Link and Physical layers of the OSI model. Note that the Internet layer does not take advantage of sequencing and acknowledgment services that might be present in the Data-Link layer. An unreliable Network Interface layer is assumed, and reliable communications through session establishment and the sequencing and acknowledgment of packets is the responsibility of the Transport layer.

12. Internet Layer

Internet Layer is the second layer of the four layer TCP/IP model. The position of Internet layer is between Network Access Layer and Transport layer. Internet layer packs data into data packets known as IP data grams, which contain source and destination address (logical address or IP address) information that is used to forward the data grams between hosts and across networks. The Internet layer is also responsible for routing of IP data grams.

Packet switching network depends upon a connectionless internetwork layer. This layer is known as Internet layer. Its job is to allow hosts to insert packets into any network and have them to deliver independently to the destination. At the destination side data packets may appear in a different order than they were sent. It is the job of the higher layers to rearrange them in order to deliver them to proper network applications operating at the Application layer [20].

This layer has the responsibility of sending packets across potentially multiple networks. Internetworking requires sending data from the source network to the destination network. This process is called routing [21].

The internet layer is not only agnostic of data structures at the transport layer, but it also does not distinguish between operations of the various transport layer protocols. IP carries data for a variety of different upper. These protocols are each identified by a unique protocol number: like, Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP) are protocols 1 and 2, respectively. The main protocols included at Internet layer are IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) and IGMP (Internet Group Management Protocol).

13. Transport Layer

The Transport Layer provides the means for the transport of data segments across the internet layer. The Transport Layer is concerned with end-to-end (host-to-host) communication. Transmission Control Protocol provides reliable, connection-oriented transport of data between two endpoints (sockets) on two computers that use Internet Protocol to communicate. User Datagram Protocol provides unreliable, connectionless transport of data between two endpoints (sockets) on two computers that use Internet Protocol to communicate. The Transport Layer sends data to the Internet Layer when transmitting and sends data to the application layer when receiving [22]. Transport layer provides communication session management between host computers and defines the level of service and status of the connection used when transporting data.

14. Application Layer

Application Layer defines TCP/IP application protocols and how host programs interface with transport layer services to use the network.

This layer provides the user with interface to communication. This could be the web-browser, e-mail client, or a file transfer client. This is the layer where the web-browser, a telnet, FTP, e-mail or other client application runs. Basically, any application that rides on top of TCP or UDP uses a pair of virtual network sockets and a pair of IP addresses. The application layer sends to and receives data from the transport layer.

This is the top layer of TCP/IP protocol suite. This layer includes applications or processes that use transport layer protocols to deliver the data to destination computers. At each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the second layer, the transport layer. Application layer includes all the higher-level protocols like DNS (Domain Naming System), HTTP (Hypertext Transfer Protocol), Telnet, SSH,

FTP (File Transfer Protocol), TFTP (Trivial File Transfer Protocol), SNMP (Simple Network Management Protocol), SMTP (Simple Mail Transfer Protocol), DHCP (Dynamic Host Configuration Protocol), X Windows, RDP (Remote Desktop Protocol) etc [23].

15. Comparison between OSI and TCP/IP Models

Transmission Control Protocol is used by Internet applications like email, World Wide Web, FTP, etc. TCP/IP was developed by the Department of Defense (DoD) to connect various devices to a common network (Internet). The main purpose behind developing the protocol was to build a robust and automatically recovering phone line failure while on the battlefield. On the other hand, Open Systems Interconnection was developed by the International Organization for Standardization (ISO). This model was made up of two components, namely, seven-layer model and the subset of protocols [24].

OSI Reference Model	TCP/IP
Application Layer	
Presentation Layer	
Session Layer	Application Layer
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Link Layer

Figure 4: OSI and TCP/IP Model [25].

There are seven layers in the OSI Model, only four in the TCP/IP Model. This is because TCP/IP assumes that applications will take care of everything beyond the Transport Layer. The TCP/IP Model also squashes the OSI's Physical and Data Link Layers together into the Network Access Layer [26]. The basic differences between the OSI and TCP/IP models are shown above in figure 4.

The Open Systems Interconnection (OSI) model is a standard "reference model" created by the International Organization for Standardization (ISO) to describe how the different software and hardware components involved in a network communication should divide labor and interact with one another. It defines a seven-layer set of functional elements, ranging from the physical interconnections at Layer 1 (also known as the physical layer, or PHY interface) all the way up to Layer 7, the application layer.

The Transmission Control Protocol (TCP) and the Internet Protocol (IP) are two of the network standards that define the Internet. IP defines how computers can get data to each other over a routed, interconnected set of networks. TCP defines how applications can create reliable channels of communication across such a network. Basically, IP defines addressing and routing, while TCP defines how to have a conversation across the link without garbling or losing data.

TCP/IP grew out of research by the U.S. Dept. of Defense and is based on a loose rather than a strict approach to layering. Many other key Internet protocols, such as the Hypertext Transfer Protocol (HTTP), the basic protocol of the Web, and the Simple Mail Transfer Protocol (SMTP), the core email transfer protocol, are built on top of TCP. The User Datagram Protocol (UDP), a companion to TCP, sacrifices the guarantees of reliability that TCP makes in return for faster communications. TCP/IP doesn't map cleanly to the OSI model, since it was developed before the OSI model and was designed to solve a specific set of problems, not to be a general description for all network communications. The relationships and differences between the OSI model and TCP/IP are [27]:

- IP corresponds to a subset of OSI Layer 3, the network layer
- TCP corresponds to OSI Layer 4 (transport) and some functions of Layer 5 (session)
- TCP/IP makes no assumptions about what happens above the level of a network session (part of OSI Layer 5), while OSI defines several more layers of standardized functions
- TCP/IP makes no prescriptions as to the link layers below IP, where OSI specifies two.
- Where an application needs functions not found in TCP/IP, the application has to supply them. In the OSI model, it is assumed that an application will never implement any functionally belonging in any defined layer, and because interfaces between layers abstract many details, it may not be able to anyway. Both the TCP/IP and OSI model work in a very similar fashion. But they do have very subtle differences too. The most apparent difference is the number of layers. TCP/IP is a four-layered structure, while OSI is a seven-layered model. OSI Model differs with TCP/IP model as follows [29]:
- OSI provides layer functioning and also defines functions of all the layers.
- The transport layer in OSI model guarantees the delivery of packets.
- It follows a horizontal approach.
- It has a separate presentation layer.
- It is a general model.
- The network layer of this model provides both connection oriented and connectionless services.
- OSI model has a problem of fitting the protocols in the model.
- Protocols are hidden in this model and are easily replaced as the technology changes.
- It defines services, interfaces and protocols very clearly and makes clear distinction between them.

TCP/IP model differs with OSI model as follows [29]:
- TCP/IP model is more based on protocols and protocols are not flexible with other layers.
- The transport layer does not guarantee the delivery of packets.
- It follows a vertical approach.
- It does not have a separate presentation layer.
- This model cannot be used in any other application.
- The network layer in this model provides connectionless services.
- In TCP/IP replacing protocol is not easy.
- In TCP/IP it is not clearly separated its services, interfaces and protocols.

The summary of the comparison is illustrated in figure 5 below:

TCP/IP	OSI
Implementation of OSI model	Reference model
Model around which Internet is developed	This is a theoretical model
Has only 4 layers	Has 7 layers
Considered more reliable	Considered a reference tool
Protocols are not strictly defined	Stricter boundaries for the protocols
Horizontal approach	Vertical approach
Combines the session and presentation layer in the application layer	Has separate session and presentation layer
Protocols were developed first and then the model was developed	Model was developed before the development of protocols
Supports only connectionless communication in the network layer	Supports connectionless and connection-oriented communication in the network layer
Protocol dependent standard	Protocol independent standard

Figure 5: Differences between OSI and TCP/IP Model [28].

16. Conclusion

In this paper an attempt has been made to explain the differences in TCP/IP and OSI models. OSI model is an architecture which gives an idea how packets transfer over the network during any communication. The Transmission Control Protocol / Internet Protocol (TCP/IP) was created by the Department of Defense (DoD) to make sure and protect dataintegrity, and also maintained communications in the time of disastrous war. However, if designed and deployed properly according to standard, a TCP/IP network can be a truly reliable and flexible one Efficiency and feasibility. The OSI norms tend to be prescriptive (for instance the "layer N" must go through "all layers below it"), whereas the TCP/IP protocols are descriptive, and leave a maximum of freedom for the implementers. One of the advantages of the TCP/IP approach is that each particular implementation can use operating system-dependent features, generally resulting in a greater efficiency (fewer CPU cycles, more throughput for similar functions), while still ensuring "interoperability" with other.

The TCP/IP and OSI architecture models both employ all connection and connectionless models at transport layer. However, the internet architecture refers to the two models in TCP/IP as simply "connections" and data grams. But the OSI reference model, with its penchant for "precise" terminology, uses the terms connection-mode and connection-oriented for the connection model and the term connectionless-mode for the connectionless model TCP/IP is the older of the two approaches to data communications and is well established throughout the world. The OSI model, however, is a proven concept that is used in all other data communications protocols. It will continue to be used as a guideline for all other communications applications.

TCP/IP combines the presentation and session layer into its application layer. TCP/IP combines the OSI data link and physical layers into one layer. TCP/IP appears simpler because it has fewer layers. TCP/IP transport layer using UDP does not always guarantee reliable delivery of packets as the transport layer in the OSI model does.

References

[1] http://compnetworking.about.com/od/networkprotocols/g/protocols.htm
[2] Marsden 1986, Section 6.1 - *Why are standards necessary?* pp. 64-65, uses BSC as an example to show the need for both standard protocols and a standard framework.
[3] Comer 2000, Sect. 11.2 - *The Need for Multiple Protocols*, pp. 177, explains this by drawing analogies between computer communication and programming languages.
[4] Sect. 11.10 - *The Disadvantage of Layering*, pp. 192, states: layering forms the basis for protocol design.
[5] http://en.wikipedia.org/wiki/Internetwork_protocol
[6] American Psychological Association (APA): Protocol layer. (n.d.). The Free On-line Dictionary of Computing. Retrieved November 04, 2014.Chicago Manual Style (CMS): Protocol layer. Dictionary.com. the Free On-line Dictionary of Computing. Denis Howe. Http (accessed: November 04, 2014).
[7] http://www.ee.surrey.ac.uk/Projects/CAL/networks/Intro.htm
[8] Andrew S. Tanenbaum in *Computer networks*, pp. 27 [2008].
[9] Andrew S. Tanenbaum in *Computer networks*, pp. 38 [2008].
[10] Stewart, K., Adams, A. and Reid, A. (2008). *Designing and Supporting Computer Networks, CCNA discovery Learning Guide*, Cisco Press, USA.
[11] http://www.dummies.com/how-to/content/layers-in-the-osi-model-of-a-computer-network.html
[12] http://www.dummies.com/how-to/content/layers-in-the-osi-model-of-a-computer-network.html
[13] http://en.wikipedia.org/wiki/OSI_model
[14] http://searchnetworking.techtarget.com/definition/OSI
[15] http://www.webopedia.com/quick_ref/OSI_Layers.asp Updated March 11, 2014 / Posted September 24, 1999, by Vangie Beal.
[16] Andrew S. Tanenbaum in Computer networks, pp. 41 [2008].
[17] http://searchnetworking.techtarget.com/definition/TCP-IP Posted by Margaret Rouse WhatIs.com
[18] http://en.kioskea.net/faq/1855-the-tcp-ip-model Published by-aakai1056-Latest update by deri58, November 2014.
[19] http://www.omnisecu.com/tcpip/tcpip-model.php Published by: Jajish Thomason.
[20] http://www.omnisecu.com/tcpip/tcpip-model.php Published by: Jajish Thomason.

[21] IP Packet Structure
[22] http://www.inetdaemon.com/tutorials/basic_concepts/network_models/TCP-IP_model/
[23] http://www.omnisecu.com/tcpip/tcpip-model.php Published by: Jajish Thomason.
[24] http://www.buzzle.com/articles/tcpip-model-vs-osi-model.html
[25] http://www.buzzle.com/articles/tcpip-model-vs-osi-model.html
[26] A Comparison of OSI Model vs. TCP/IP Model http://www.inetdaemon.com/tutorials/basic_concepts/network_models/comparison.shtml Last Updated: Saturday, 15-Feb-2014 12:46:31 MST | By *InetDaemon*
[27] http://searchnetworking.techtarget.com/answer/What-is-the-difference-between-OSI-model-and-TCP-IP-other-than-the-number-of-layers
[28] http://www.buzzle.com/articles/tcpip-model-vs-osi-model.html
[29] http://www.studytonight.com/computer-networks/comparison-osi-tcp-model
[30] OSI Reference Model: An Overview by Gaurav Bora, Saurabh Bora, Shivendra Singh, Sheikh Mohamad Arsalan published in International Journal of Computer Trends and Technology (IJCTT) – volume 7 number 4– Jan 2014