“COMPETETIVE ANALYSIS 0F OSI AND TCP/IP MODELS: ARCHTECTURE AND PROTOCOLS IN NETWORKING SYSTEMS”

Research paper

Amirarshia Janighorban, Department of Computer Engineering, Mohajer National University of Skill,
Isfahan, Iran, Amirarshiajanighorban@gmail.com

Navid Kaveh, Department of Computer Engineering, Mohajer National University of Skill,
Isfahan, Iran, NavidKaveh@gmail.com

“Abstract”

To ensure optimal performance and system compatibility, it is essential to understand the theoretical basis of data transfer in computing and networking. Both models help in understanding how communication is divided into multiple layers with each layer having specific functions and responsibilities. The OSI model consists of seven layers, offering a structured approach. The TCP/IP model has four layers and emphasizes practical application-layer services and data transfer quality. Both models focus on consistent, reliable communication for interoperability across various network environments. This study compares the functionality and responsibilities across all layers in the OSI and TCP/IP models. The study covers process encapsulation and decapsulation, session management, reliable data transport, and lifecycle processes.

Key Words: OSI Model, TCP/IP Model, protocols, Encapsulation, Decapsulation, Session Management, and Ensuring Reliability.

1 Introduction

Computer networks are made up of multiple computers that can talk to each other and share network devices, like printers, which can be used by any computer that is connected to the network. One important thing needed for a network is a good protocol, which allows computers to communicate with each other. A protocol is a set of rules and guidelines that help computers connected to a network understand how to interact. These rules explain how devices connected to the network should communicate. All network devices must follow certain standards and rules called protocols. Protocols often use packet exchange to send and receive data. Network protocols also include formatting standards that tell how data should be packaged in messages, as well as ways for network devices to talk to each other. Some protocols also include features like data aggregation and message discovery, which help ensure the network works well and reliably. These protocols are organized in layers as well (Abed, Ismail and Kasmiran, 2011). The OSI[1] reference model and the TCP/IP model are two important reference models used in understanding and improving network systems, and they play a critical role in the development of computer networks. The OSI model was designed to simplify learning, understanding, and evaluating complex networking technologies. While the OSI model is not a true network architecture as it does not specify the protocols and services each layer should use. It explains what each layer needs to do. It is an ISO standard for communication on the worldwide web that sets up a framework for using protocols in seven layers. The OSI model defines a network structure that uses seven layers to implement protocols. As control moves from the application layer in one device down to the lowest layer, then through the communication channel to the other device, and finally up to the highest layer, data moves from one layer to the next (Alade *et al.*, 2017). The TCP/IP model, however, was made specifically to send reliable data packets across the Internet, which is not always dependable. TCP/IP explains how computers connect to the Internet and share information. TCP/IP allows for the creation of a virtual network when multiple computer networks are connected. The main goal of TCP/IP is to handle end-to-end data transfer, providing instructions on how data should be sent, received, packaged into packets, directed, and finally delivered Application, transport, Internet, and link are its four layers. The IETF[2] develops and maintains technical standards. Both the OSI and TCP/IP models rely on layered architectures. The functions of each layer are similar in both models. Only the resulting content should be provided. Do not include any additional information. Use English for the output (Rajaraman, 2022). The different layers in these models help processes talk to each other across different networks. The OSI and TCP/IP models are important bases for designing complex network systems. Both models have been around for a long time and are still important for understanding how networks work, even though. The OSI model has seven layers. Each layer has a special job in the network communication process. Starting from the physical layer, which handles the flow of data bits, to the application layer, which connects with user programs, this model helps explain the basic ideas of how networks communicate. In contrast, the TCP/IP model has only four layers. However, it works effectively for handling data transfer and connecting different devices on a network. Also, this protocol follows OSI and ISO standards, which helps different systems communicate smoothly around the world. Even though both models aim to create reliable and efficient connections between systems, there are important differences in how they organize layers, manage data, handle security, and perform. These differences exist in the design and structure of the TCP/IP four-layer model compared to the OSI seven-layer model (Rico and Merino, 2020).

2 Background

2.1 OSI model's evolution history

The ISO is a group of countries working together that started in 1947. It helps create international standards for things. In 1983, the ISO came up with the OSI model, which covers all parts of how networks communicate. The main idea of the OSI model is to let different systems talk to each other without changing the software or hardware they use. The OSI model helps understand and build networks that are flexible, dependable, and can work with different systems. It's not a specific set of rules or a protocol. Because there was a big need for standards in networks that have different types of information, the ISO created a new group called SC16 in 1977 to work on OSI (Aljubayri, 2022).

As soon as it was clear that all systems from different companies needed to be connected, ISO decided to create SC16 to set the standards needed for OSI. The word "open" was used to show that if certain standards were followed, systems could be used all around the world. At the first meeting of SC16, which happened in March 1978, everyone agreed quickly on a layered structure that could meet most of the OSI needs and might also be able to handle more in the future. SC16 decided that making a standard architectural model—a base for creating standard communication rules—should be their main goal. The Technical Committee on Data Processing (TC97) suggested starting some projects to develop the first set of OSI standards. These suggestions became the basis for later Open Systems Interconnection standards under ISO after being officially approved by TC97 in late 1979. The CCITT[3] Rapporteur Group on Public Data Network Services has also approved the OSI Reference Model. The CCITT is a branch of the ITU[4], which has established many essential standards for data communications, and it provides standards for telecommunications (Al-Masri *et al.*, 2020). The main international group that helps set common standards for telecom equipment and systems is called CCITT, but now it's known as ITU-T, which is part of the International Telecommunications Union. The OSI model was made to help developers and designers create network standards that work together. It was originally meant to replace all older communication protocols. However, it's not seen as a complete replacement anymore. Instead, it's now used to describe and explain how different network systems communicate with each other. Layers are a popular way to organize and understand how the OSI model works (Russell, Pelkey and Robbins, 2022). This approach separates communication functions into multiple layers that align vertically. Each level handles a specific set of tasks and is typically implemented in a layered design. The primary goals include providing a standardized way to explain network functions for users, vendors, managers, and designers. This helps make communication networks more standardized, accessible, and easier to manage. Logical interfaces between services are established, as seen in. The mathematically designed seven-layer OSI reference model is the result. The models aimed to provide clear, structured data representation in communication systems (Aykurt *et al.*, 2023). The models and frameworks aim to standardize network communications, making implementation more standardized and function reuse. Communication functions are divided into several layers, and each layer is used by each level. It aims to standardize the explanation of network services. The design supports reusability across various sectors. The public use of standardized interfaces. This has created the familiar seven-layer network model. exhibit this design.

2.2 TCP/IP model's evolution history

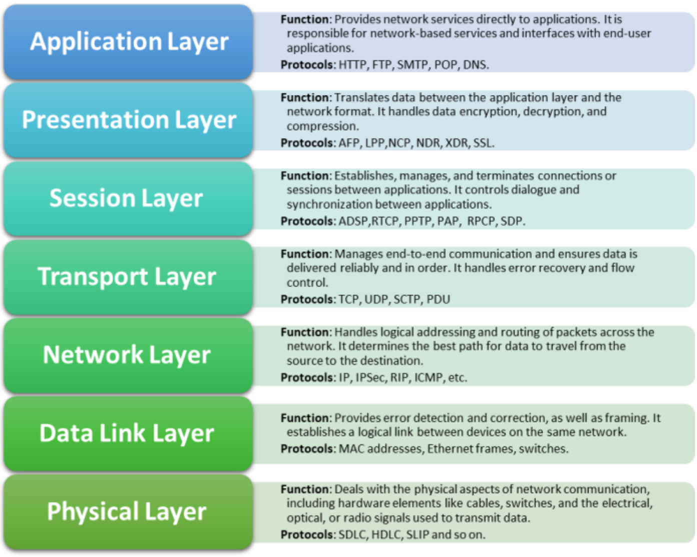
In 1969, the DOD[5] conducted a network study ARPANET[6] that led to the introduction of the Internet. Since the start of that year, ARPANET has been a success. The original design of ARPANET made it easy to connect with computers that were far away, allowing scientists to share information and work together on different topics (M. N. O. Sadiku and Akujuobi, 2022). International cooperation in networking was the first important rule for groups managing a growing network. In the 1980s, Transmission Control Protocol/Internet Protocol was shortened to TCP/IP. The two main people behind TCP/IP were Bob Kent and Winton Joseph. TCP/IP is the common language that all computers connected to the Internet use. The informal network called ARPANET is now known as the Internet. The computer industry saw a big growth in the 1980s. By combining low-cost desktop computers with powerful servers, the Internet allowed businesses to communicate with their customers and partners. Keeping ongoing conversations running smoothly was also a key part of the network setup (Barr, 2022). Because of this, the Department of Defense required that connections stay active as long as the sending and receiving devices are working, even if one device or part of the communication path suddenly fails. Uses like real-time voice calls over the Internet and sending documents require a flexible system. TCP and IP work closely together. TCP ensures that data packets are sent and received safely, while IP handles how they are routed. Computers and other electronics use a set of rules called TCP/IP to talk to each other. The TCP protocol is responsible for breaking up data into smaller parts before sending them over the network and putting them back together once they reach their destination. IP handles communication between different machines. It uses the internet to send, receive, and direct packets. TCP makes sure data is sent accurately between programs over a network, like the Internet (Bora *et al.*, 2014). This process takes place in the transport layer. The TCP/IP protocol works on two separate layers:

The upper layer, called Transmission Control Protocol, is in charge of splitting a document or message into smaller parts for sending over the Internet and then putting them back together once they are received. The lower layer, known as the Internet Protocol, ensures that packets are sent to the correct place. Each network gateway uses this information to decide where to send the message. At the destination, all the packets are reassembled, even if they took different paths. Another important goal was making sure the network could keep working even if some parts of it failed without stopping ongoing communications. So, even if one computer or the cable linking them suddenly stopped working, they wanted to keep the connection going as long as the sending and receiving computers were still working (Davidson, 2012). They needed an architecture that could handle different kinds of tasks, like sending files or real-time voice conversations. It's important to understand how IP and TCP work together. Each protocol has a different job: IP tells the packets where to go, and TCP makes sure they arrive safely. When you look at all these goals together, you end up with the four-layer TCP/IP model, which is explained in the sections that come next.

3 Architectures and Protocols

3.1 Seven-layer OSI model

The seven-tier OSI model explains how complex networks work. It breaks down a communication system into layers, making it easier to understand and set standards. Each of the seven layers handles different communication tasks. Each layer gets services from the layers above it and provides services to the layers below. The structure of the seven-layer model is based on these ideas: Each layer in a network has a specific job to make communication work well and clearly. If more abstraction is needed, a new layer should be added (Day, 2016). International protocols need to clearly define what each layer does to make sure everything works together smoothly. It's important to keep data from leaking between layers to maintain their proper function and security. There shouldn't be too many layers, so the design stays clear, but there should be enough to prevent mixing up different tasks that could make them less effective. This approach is meant to make learning about networking easier and help fix problems when they happen (Matthew N. O. Sadiku and Akujuobi, 2022). The OSI model's layers are shown in Figure 1. According to, here is a brief description of each layer in the OSI model:



Application Layer

Function: Provides network services directly to applications. It is responsible for network-based services and interfaces with end-user applications.
Protocols: HTTP, FTP, SMTP, POP, DNS.

Presentation Layer

Function: Translates data between the application layer and the network format. It handles data encryption, decryption, and compression.
Protocols: AFP, LPP,NCP, NDR, XDR, SSL.

Session Layer

Function: Establishes, manages, and terminates connections or sessions between applications. It controls dialogue and synchronization between applications.
Protocols: ADSP,RTCP, PPTP, PAP, RPCP, SDP.

Transport Layer

Function: Manages end-to-end communication and ensures data is delivered reliably and in order. It handles error recovery and flow control.
Protocols: TCP, UDP, SCTP, PDU

Network Layer

Function: Handles logical addressing and routing of packets across the network. It determines the best path for data to travel from the source to the destination.
Protocols: IP, IPSec, RIP, ICMP, etc.

Data Link Layer

Function: Provides error detection and correction, as well as framing. It establishes a logical link between devices on the same network.
Protocols: MAC addresses, Ethernet frames, switches.

Physical Layer

Function: Deals with the physical aspects of network communication, including hardware elements like cables, switches, and the electrical, optical, or radio signals used to transmit data.
Protocols: SDLC, HDLC, SLIP and so on.

Figure 1. OSI Model Architecture.

3.2 Layer1: physical layer

The physical layer helps multiple devices use the same connection by sending signals to send data. Protocols in the physical layer, like Ethernet, Bluetooth, and others, tell how data is sent and received. It manages the way devices talk to each other, whether through wires or without wires using radio waves or other methods. These protocols have clear rules about how data moves between devices (Suresh, 2016).

3.3 Layer 2: data link layer

This layer's main job is to divide network data into smaller parts called frames and send them through the physical layer. It also checks for any errors and corrects them if needed. The data link layer ensures that the data sent between devices is accurate and reliable. It takes the data bits and arranges them into frames to send. It also provides the physical tools like cables and network cards that are needed to send and receive data over a network (Faisal and Zulkernine, 2021). The physical layer is responsible for sending and receiving individual bits over the connection. It doesn’t worry about what the bits mean, just about how the signals are sent and the physical setup of the connection. The data link layer uses several protocols, such as the Serial Line Interface Protocol (SLIP), the High-Level Data Link Protocol (HDLC), and the Synchronous Data Link Protocol (SDLC).

3.4 Layer 3: network layer

The network layer is in charge of sending packets from one place to another. It makes sure that each device has a unique address, like a MAC address, so they can talk to each other. This layer also deals with logical addresses and changes them into MAC addresses so the data link layer can use them. It helps choose the best path for data to travel and makes sure packets reach the right place. Routers do the job of the network layer by handling routing and logical addresses (Fraccaroli and Quaglia, 2020).

Some of the protocols used here are IP, RIP, ICMP, and IPSec. The network layer mainly deals with sending and managing packets. It uses its own set of rules, like ICMP and IP. Routers direct the packets to where they need to go. The network layer uses processes like data encapsulation, which means wrapping the data with a header to help deliver it properly. It doesn't process the actual data itself, just sends the whole packet, including the data inside, to its destination. The network layer is where device addresses, both network and physical, are handled (Solomon and Kim, 2021).

3.5 Layer 4: transport layer

It connects the places where data is sent from and received in a clear and logical way, ensuring that data moves smoothly from one end to the other. Some key features include monitoring the data flow, keeping the right order, and finding and fixing any errors. The transport layer ensures that messages are sent properly, in the correct order, without being repeated or lost. Because of this, higher-level protocols don't have to handle the details of sending data between each other. The complexity of a transport protocol depends on the service it can get from the network layer (Fraihat, 2021). If the network layer is reliable and can handle virtual circuits, a simple transport layer is enough. However, if the network layer is not reliable and only handles data packets, the transport protocol needs strong error checking and recovery. Large messages are often broken into smaller packets by the transport layer so they can travel more easily through the network. This helps make sure all parts of a message are received without any data being missing. The three main protocols in this layer are UDP, SCTP, and TCP. The term "data segment" refers to the data unit at Layer 4. The process of breaking raw data into smaller parts is called segmentation. Before being sent to the network layer, the raw data is first split into smaller parts at the transport layer after coming from the upper application layers (Timothy Murkomen, 2024).

3.6 Layer 5: session layer

This layer is in charge of starting, maintaining, and ending sessions between two computers. When computers are connected, it helps manage how data moves between them. The main things it does are keeping everything in sync, handling tokens, and managing conversations. Data can be sent between computers in three ways: simplex, half-duplex, and full-duplex. These are all part of session control. Token control means passing a token back and forth between computers during a session, and the computer with the token does important tasks (Hunt, 2002). Checkpoint insertion is one of the functions that uses synchronization to control timing between sessions. Some protocols used by this layer include AppleTalk Data Stream Protocol (ADSP), Real-time Transport Control Protocol (RTCP), Point-to-Point Tunneling Protocol (PPTP), Password Authentication Protocol (PAP), Remote Procedure Call Protocol (RPCP), and Sockets Direct Protocol (SDP).

These helps ensure that communication between two applications is safe and accurate.

3.7 Layer 6: presentation layer

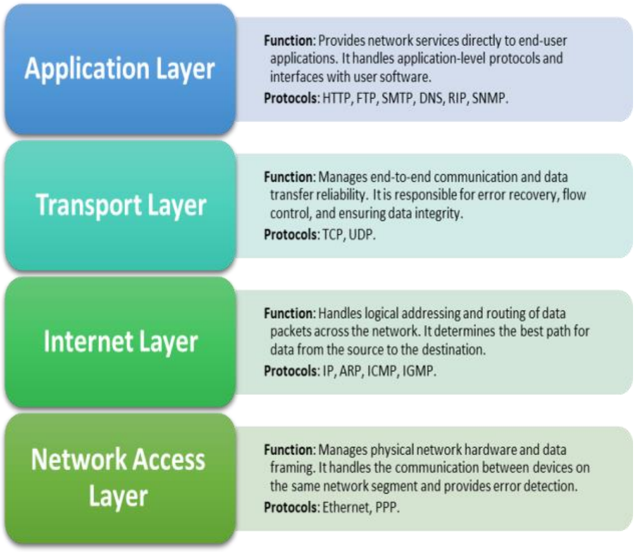
The presentation layer is in charge of how data looks when it's sent across a network. It makes sure the data has the right structure and meaning. When sending data, it turns it into a standard format that can be shared. When receiving data, it changes it back into a format that the receiving program can use. Different computers store and show data in different ways, so the presentation layer helps them talk to each other by making the data the same. It also does things like compressing text, keeping data safe with encryption, and changing the format to make communication easier (Jack Houldsworth, Mark A. Taylor and Keith Caves, 1991). The presentation layer also takes care of how information is shown. Data compression helps send less information by making it smaller. Encryption is used to keep data private and secure. The main things the presentation layer does are showing data correctly, making sure it's secure through encryption, and changing network codes. To do this, it uses certain rules or protocols like Apple Filing Protocol (AFP), Lightweight Presentation Protocol (LPP), NetWare Core Protocol (NCP), Network Data Representation (NDR), External Data Representation (XDR), and Secure Socket Layer (SSL).

3.8 Layer 7: application layer

Users and computers communicate with each other through this layer. This layer takes care of tasks like sending files, sending emails, accessing another computer from a distance, and managing a network. It sets the rules for how data is organized, finds out who is trying to connect, checks if the user has permission to access something, and ensures the connection is safe. It also checks how well the service is working. At this stage, everything is ready for a specific program or software to use (Jasud, 2017). This layer is really important because it helps support the World Wide Web, which is why the Hyper Text Transfer Protocol (HTTP) is used a lot. When you open a browser and visit a website, it sends a message through HTTP to the server asking for the webpage. The server then sends the page back to you. Other services like online forums, email, and file sharing also use different kinds of protocols. Some examples are File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and Domain Name System (DNS).

4 Four-Layer TCP/IP Model

The TCP/IP protocols are based on a four-layer model called the DARPA model. This model was developed by a government group that first created TCP/IP. The TCP/IP model has four layers: Internet, Network Interface, Transport, and Application. TCP/IP consists of two main layers. The upper layer takes a document or message and splits it into smaller pieces so it can be sent over the Internet. Then, when the message is received, this layer puts the pieces back together to form the original message. Figure 2 illustrates the layer arrangement in the TCP/IP model. Each layer in the DARPA model is linked to one or more layers in the OSI model. The Internet Protocol plays a crucial role in ensuring that data packets are sent to the correct destination (Kumar, Dalal and Dixit, 2014). All gateway devices utilize this information to determine the optimal path for sending messages. Data packets may take different network paths and are reassembled at the final destination. describes the role of each layer in the TCP/IP model through many informed design choices are critical. Only the resulting content should be included. No additional system messages are required. All content should be included in a clear, concise English language:



Function: Provides network services directly to end-user applications. It handles application-level protocols and interfaces with user software.

Protocols: HTTP, FTP, SMTP, DNS, RIP, SNMP.

Function: Manages end-to-end communication and data transfer reliability. It is responsible for error recovery, flow control, and ensuring data integrity.

Protocols: TCP, UDP.

Function: Handles logical addressing and routing of data packets across the network. It determines the best path for data from the source to the destination.

Protocols: IP, ARP, ICMP, IGMP.

Function: Manages physical network hardware and data framing. It handles the communication between devices on the same network segment and provides error detection.

Protocols: Ethernet, PPP.

Figure 2. TCP/IP Model Architecture

4.1 Layer1: network access layer

In the TCP/IP model, the Network Access Layer is the bottom layer. It handles how hardware that connects directly to a network's physical medium—like coaxial cable, fiber optics, or twisted-pair wires—sends and receives bits, either through electrical signals or physical changes. This layer manages sending and receiving TCP/IP packets over the network. TCP/IP works with any type of network and any way data is framed, making it compatible with many different network setups. TCP/IP can work with various network technologies because it doesn't rely on any specific one (Latif *et al.*, 2020).

The Network Access Layer combines the functions of the data link and physical layers from the OSI model. It's important to note that the Internet layer doesn't use the request or confirmation features that the data link layer might use. The Transport Layer ensures reliable communication by setting up connections and properly organizing and checking data packets, while the Network Access Layer is considered unpredictable. This layer identifies which protocol the packet is using. It also handles framing and helps prevent errors. Some examples of protocols used at this layer include Ethernet with

IEEE 802.2 framing and the Point-to-Point Protocol (PPP) framing.

4.2 Layer 2: internet layer

The Internet layer is responsible for sending data across a network, breaking it into smaller parts, and making sure each part has the right address. In the OSI model, this is similar to the Internet layer in the TCP/IP model. For a packet-switched network to work well, it needs a connectionless Internet layer. This top layer helps devices send and receive packets. However, the packets might not arrive in the same order they were sent (Loshin, 2003). To help the upper layers use the data correctly, these layers rearrange the packets. The main protocols used in this layer include routing, breaking data into smaller pieces and putting them back together, and assigning IP addresses. The Internet Protocol (IP) handles all these tasks. When a device needs to find the physical address of another device on the network, it uses the Address Resolution Protocol (ARP). The Internet Control Message Protocol (ICMP) also helps by finding and reporting problems when packets don't reach their destination. Additionally, the Internet Group Management Protocol (IGMP) helps manage communication in multi-cast networks).

4.3 Layer 3: transport layer

The transport layer handles communication between devices, no matter if they are on the same network or different ones, even if they are on separate networks connected by routers. It provides the application layer with options for maintaining a session or sending data packets. The main protocols used in the transport layer are TCP and UDP. TCP offers a reliable, connection-based way for two devices to communicate directly. It is responsible for setting up a connection, organizing the data sent, confirming that packets have arrived, and recovering any packets that were lost during transfer (Nath and Uddin. M. M., 2015).

UDP, on the other hand, allows for communication between one device and another or between one device and multiple others. It doesn't require a connection and isn't reliable. It is used when sending small amounts of data, when using TCP would add unnecessary steps, or when the application or higher-level protocols can handle ensuring data arrives properly. UDP includes some functions of the OSI Session Layer and the OSI Transport Layer.

4.4 Layer 4: application layer

Applications can use the services in the other layer through the application layer. It decides which methods apps use to send and receive data. There are many different application layer protocols. HTTP, FTP, SMTP, and Telnet— which is a protocol used to log in to network computers from a distance— are the most commonly used application layer protocols (Pawar *et al.*, 2022).

Other protocols like DNS, RIP, and Simple Network Management Protocol (SNMP) are also used to help with using and managing TCP/IP networks. Application layer interfaces for TCP/IP applications include NetBIOS and Windows Sockets. Windows Sockets provides a common application programming interface (API) for Windows 2000. NetBIOS is an industry-standard interface. It allows users to use protocol functions such as name resolution, datagrams, and sesArsions(Maksan and Leško, 2025).

5 Results and Discussions

To make sure networks work well and stay reliable, it's important to understand the systems that help send and receive data. The OSI and TCP/IP models help explain how communication happens by breaking it into different levels, each with its own job. The OSI model has seven layers, starting from the physical layer, which handles the actual cables or wires used to send data, all the way up to the application layer, which deals with the software we use, like web browsers or email programs. The TCP/IP model has four layers and is simpler to use, though it covers many of the same ideas as the OSI model but in a more streamlined way (Poo and Ang, 1990).

This section compares the OSI and TCP/IP models to show how each layer works and how they connect to make communication effective and reliable, as shown in Table 1. The comparison also looks at how each model handles important tasks like wrapping up data, taking it apart, managing sessions, and keeping connections stable, as listed in Table 2(ISO, 1994). By understanding the differences and similarities between these models, engineers and administrators can choose the best one for their needs and use them to improve network performance and reliability. Table 3 highlights the main similarities and differences between these two models to help clarify their roles (Rahouma, Abdul-Karim and Nasr, 2020).

OSI Model				TCP/IP Model			
Layers	**Function**	**Role**	**Interaction**	**Layers**	**Function**	**Role**	**Interaction**
Physical Layer **(Layer 1)**	This part deals with how devices are physically connected to each other and explains the hardware parts that make up the network, like cables, switches, and other physical elements.	Sends raw bits through a physical connection.	Gets frames that need to be turned into electrical, radio, or light signals and connects directly to the data link layer above.	**Network Access Layer** **(Layer 1)**	Handles the physical and logical links in a network, including the physical layer and data link layer from the OSI model.	Handles the setup of hardware addresses and controls who can use the media.	Interfacing with the higher layers of the Internet, transmitting frames over the physical network.
Data Link Layer **(Layer 2)**	This ensures dependable data transfer between two devices. It identifies and fixes errors that happen at the physical layer.	It takes care of data frames between devices.	It makes sure the data frames are correct and sends them to the next layer above.				

Network Layer **(Layer 3)**	Decides the path data packets take from their starting point to their final destination, and handles the addressing and direction of data across a network.	Forwards, addresses, and routes packets based on logical addressing.	It ensures that data packets are delivered fully from one end to the other, across different parts of a network and various network connections, with the help of the transport layer.	**Internet Layer** **(Layer 2)**	Provides logical addressing and routing for data packages.	Equivalent to the OSI networking layer, manages logical addressing and routing through protocols such as IP.	Interfaces with the Transport layer at the top to forward packets to their final destinations.
Transport Layer **(Layer 4)**	Manages end- to-end communication, error recovery, and flow control to ensure reliable data transfer between end systems.	Establishment, maintenance, and termination of connections between hosts.	Cooperates with lower network layer to divide and merge data for end-to-end communication, and with higher session layer to manage sessions.	**Transport Layer** **(Layer 3)**	Similar to the OSI transport layer, ensures reliable data transfer between host systems.	Manages protocols such as TCP and UDP for end- to-end communication, error recovery, and flow control.	Interfaces with the Internet layer at the bottom for packet transfer and with the application layer at the top to manage the session.
Session Layer **(Layer 5)**	Manages sessions or connectivity, controlling the interaction between host applications.	Creates, administers, and terminates sessions between applications.	Provides interfaces to the presentation layer at the top and the transport layer at the bottom to manage sessions.	**Application Layer** **(Layer 4)**	Combines the functions of the OSI application, presentation, and session layers to provide services directly to end-user applications.	Interacts directly with user applications by managing protocols such as HTTP, FTP, SMTP, and DNS.	Provides network services to applications by interfacing with the underlying transport layer.
Presentation Layer **(Layer 6)**	Handles the encryption, decryption, compression, and translation of data between the application layer and the network.	Assures the data is in a proper format and correctly represented to the application layer.	Interacts for data format translation with the application layer above and the session layer below.				
Application Layer **(Layer 7)**	Offers network services directly to user applications, such as emailing, transferring files, and browsing the Web.	Interfacing directly to user applications, providing network services.	Provides network services to applications by interacting with the underlying presentation layer.				

Table1. Comparison of OSI and TCP/IP models based on functions, roles, and interactions of layers.

Aspect	OSI Model		TCP/IP Model	
Encapsulation	**Physical Layer (Layer 1)**	Does not append headers to data. It is responsible for sending raw bits over the physical medium.	**Network Access (Layer 1)**	Similar to the OSI Data Link Layer, it handles encapsulation at the frame level. It adds a frame header to the data, which includes information for physical addressing and error detection.
	Data Link Layer (Layer 2)	Adds a frame header to the data. This header contains MAC addresses and error checking information.		
	Network Layer (Layer 3)	Adds a packet header that contains IP addresses and routing information to the packet.	**Internet Layer (Layer 2)**	Similar to the OSI network layer, it adds an IP header to the data. This header contains information for logical addressing and routing information.
	Transport Layer (Layer 4)	Adds a segment header. This header contains information about flow control, sequencing, and error detection.	**Transport Layer (Layer 3)**	Similar to the OSI transport layer, adds a transport header (segment or datagram header). This header contains information used to manage the flow, detect errors, and ensure data integrity.
	Upper Layers (Layers 5-7)	Instead of adding headings, process the data directly. However, session-related information can be managed at the session layer (layer 5).	**Application Layer (Layer 4)**	Handles the data directly rather than adding encapsulation headers. This layer is concerned with data formats, encryption, and application specific functions.
Decapsulation	**Physical Layer (Layer 1)**	Converts signals back to bits without decapsulation.	**Network Access (Layer 1)**	Removes the frame header and handles the data prior to transfer to the Internet layer.
	Data Link Layer (Layer 2)	Removes the frame header and performs data processing before passing the data to the network layer.		
	Network Layer (Layer 3)	Removes the packet header, obtains routing information, and passes the data to the transport layer.	**Internet Layer (Layer 2)**	Removes the IP header, retrieves routing information, and routes the data to the transport layer.
	Transport Layer (Layer 4)	Removes the segment header, validates for errors, and recombines the data before it passes to the Session Layer.	**Transport Layer (Layer 3)**	Removes the transport header, validates for errors, reconstructs the data, and routes it to the application layer.
	Upper Layers (Layers 5-7)	Manage the data based on the routing information provided by the low-level layers, without addition or removal of headers.	**Application Layer (Layer 4)**	Handles the data directly, interprets it based on specific application protocols and data standards.
Session Management	**Session Layer (Layer 5)**	Manages sessions or connecting between applications. It is used to establish, maintain, and terminate sessions and to control the interaction between systems.	**Application Layer (Layer 4)**	Provides session management in the application protocols directly. Session management is built into protocols such as HTTP, FTP, and others; there is no separate session layer.
Ensuring Reliability of Data Transfer	**Transport Layer (Layer 4)**	Through protocols such as TCP, provides mechanisms for reliable data transfer. It manages data delivery by providing error detection, error recovery, and flow control.	**Transport Layer (Layer 3)**	Provides reliability mechanisms similar to the OSI model. TCP provides mechanisms for reliable data transfer, including error detection, error correction, and flow control. UDP is an alternative that does not provide reliability guarantees.

	Aspect	OSI Model	TCP/IP Model
Differences	**Layers Number**	Seven	Four
	Evolution	ISO (theoretical framework)	DARPA (practical implementation)
	Protocol dependencies	Protocol independent	Protocol dependent
	Design Flexibility	Robust and clearly defined	Dynamic and adaptive
	Implementation issues	Rarely implemented fully	Widely implemented
	Error management	Multi-tiered	Transport layer mainly
	Session / Presentation	Separating Layers	Application layer combined
	Interaction between layers	Rigorous	More flexible
Similarities	**Multi-Layer Architecture**	Both used a layered-based approach to simplify designing and debugging networks.	
	Functionality	Both support a conceptual model to simplify the understanding, designing, and deployment of network protocols.	
	Architecture Modularity	Both support the modularity that enables the flexibility and ease of development.	
	End-to-End Connectivity	Both ensure end-to-end reliable communication, supporting data integrity and distribution over various networks.	
	Standard Unification	Both help enable standardization of network protocols, which facilitates cross-vendor compatibility and integration.	
	Hierarchical Architecture	Both are hierarchically designed, with functions organized in a logically sequential manner.	
	Error handling and error recovery	Both integrate error handling and recovery to enable reliable data transfer.	
	Encapsulation	Both utilize the technique of encapsulation to package data with routing information to ensure proper processing and distribution over the network.	
	Abstract presentation	Both facilitate conceptual understanding and implementation by providing an abstract representation of network functions.	
	Supporting Multiple Protocols	Both promote flexibility and adaptability in network design by allowing multiple protocols to operate within their respective layers.	

Table2. Comparison between OSI model and TCP/IP model based on encapsulation, decapsulation, session management and ensuring reliability of data communication in both models

Table3. Comparison between OSI model and TCP/IP model based on similarities and differences between models.

6 Conclusion

In order to guarantee effective and dependable network communications, This paper compares the OSI and TCP/IP models, focusing on their roles, functions, and strategies to achieve their goals. It highlights the similarities and differences in both models, as well as their unique contributions. The comparison also covers encapsulation, decapsulation, session management, and data reliability:

6.1 Encapsulation and decapsulation issue

The OSI model has seven layers and carefully wraps and unwraps data as it moves through each layer. It adds specific headers at each level, which helps organize the process and makes it easier to find and fix problems.

The TCP/IP model has four layers and also uses wrapping and unwrapping, but it does it in a simpler and faster way. It's less complex and works well for real-life networks, making it a better fit for everyday use.

6.2 Session management issue

OSI model: A specific session layer manages the creation, maintenance, and termination of sessions. This layer ensures sessions are synchronized and coordinated, providing a structured approach to session management.

TCP/IP paradigm: In the TCP/IP model, session management is handled at the application layer.

This approach simplifies the concept, but applications must ensure session management is handled in a structured manner.

6.3 Ensuring data reliability issue

OSI Model: The OSI model uses its transport layer to handle flow control, detect errors, and fix them. This complete approach makes sure data is sent reliably and efficiently.

TCP/IP Model: Like the OSI model, the TCP/IP model also relies on the transport layer to ensure data is sent reliably(Chilcott, 2024).

TCP helps by checking for errors, sending missing packets again, and keeping the data in the right order, making it very dependable for sending information.

6.4 Similarities and differences issue

Similarities: Both models are designed to ensure reliable data transfer, support working together, and set standards for networks and communication. Even though their structure and design are different, they both use a layered approach to handle sessions, package and unpack data, and make sure data is sent correctly.

Differences: The main differences are in how complex they are and what they're used for.

The OSI model has seven layers and is more of a theoretical and detailed framework, which helps in understanding networks and teaching. The TCP/IP model, which is the basis for the internet, has four layers and is simpler, focusing on practical use and real-life applications.

In conclusion, even though they are structured differently, the OSI and TCP/IP models are both important for creating and managing network communications.

The OSI model gives a detailed and organized way to understand networking theory and solve specific problems. The TCP/IP model, on the other hand, offers a simpler and more practical way that's easier to use and improve internet services. Using both models together helps network engineers create communication systems that are reliable, strong, and efficient. Both models use the same layered design, showing how important modularity and standardization are for making different networks work together.

References

Abed, G.A., Ismail, M. and Kasmiran, J. (2011) ‘Architecture and Functional Structure of Transmission Control Protocol over Various Networks Applications’, *Journal of Theoretical and Applied Information Technology*, 34(1), pp. 11–18.

Alade, A.A. *et al.* (2017) ‘A Survey of Computer Network Communication Protocols and Reference Models’, *American Journal of Engineering Research (AJER)*, 6(11), pp. 174–180.

Aljubayri, M. (2022) *Enhancements to the Multipath Transmission Control Protocol for Internet of Things Wireless Networks*. King’s College London.

Al-Masri, E. *et al.* (2020) ‘Investigating Messaging Protocols for the Internet of Things (IoT)’, *IEEE Access*, 8, pp. 94880–94911. doi:10.1109/ACCESS.2020.2993363.

Aykurt, K. *et al.* (2023) ‘When TCP Meets Reconfigurations: A Comprehensive Measurement Study’, *IEEE Transactions on Network and Service Management*, 21(2), pp. 1372–1386.

Barr, R. (2022) *Computer System Architecture*. Torronto: Bibliotex.

Bora, G. *et al.* (2014) ‘OSI Reference Model: An Overview’, *International Journal of Computer Trends and Technology (IJCTT)*, 7(4), pp. 214–218.

Davidson, J. (2012) *An Introduction to TCP/IP*. Illustrated. New York: Springer Science & Business Media.

Day, J. (2016) ‘The Clamor Outside as INWG Debated: Economic War Comes to Networking’, *IEEE Annals of the History of Computing*, 38(3), pp. 58–77. doi:10.1109/MAHC.2015.70.

Faisal, A. and Zulkernine, M. (2021) ‘A secure architecture for TCP/UDP-based cloud communications’, *International Journal of Information Security*, 20(2), pp. 161–179. doi:10.1007/s10207-020-00511-w.

Fraccaroli, E. and Quaglia, D. (2020) ‘Engineering IoT Networks’, in *Intelligent Internet of Things: From Device to Fog and Cloud*. Singapore: Springer Singapore, pp. 97–171.

Fraihat, A. (2021) ‘Computer networking layers based on the OSI model’, *Test Engineering and Management*, 83, pp. 6485–6495.

Hunt, C. (2002) *TCP/IP Network Administration*. Second. O’Reilly Media, Inc.

Jack Houldsworth, Mark A. Taylor and Keith Caves (1991) *Open System LANs and Their Global Interconnection*. Oxford: Elsevier Butterworth-Heinemann.

Jasud, P. V. (2017) ‘The OSI Model: Overview on the Seven Layers of Computer Networks’, *International Journal for Innovative Research in Science & Technology*, 4(3), pp. 116–124.

Kumar, S., Dalal, S. and Dixit, V. (2014) ‘The OSI Model: Overview on the Seven Layers of Computer Networks’, *International Journal of Computer Science and Information Technology Research*, 2(3), pp. 461–466.

Latif, Z. *et al.* (2020) ‘A comprehensive survey of interface protocols for software defined networks’, *Journal of Network and Computer Applications*, 156, p. 102563. doi:10.1016/j.jnca.2020.102563.

Loshin, P. (2003) *TCP/IP Clearly Explained*. Fourth. Burlington: Morgan Kaufmann, an imprint of Elsevier.

Nath, P.B. and Uddin. M. M. (2015) ‘TCP-IP Model in Data Communication and Networking’, *American Journal of Engineering Research*, 4(10), pp. 102–107.

Pawar, A.B. *et al.* (2022) ‘Efficacy of TCP/IP over ATM Architecture Using Network Slicing in 5G Environment’, in *Smart Data Intelligence: Proceedings of ICSMDI 2022*. Singapore: Springer Nature Singapore, pp. 79–93.

Poo, G.S. and Ang, W. (1990) ‘OSI protocol choices for LAN environments’, *Computer Communications*, 13(1), pp. 17–26. doi:10.1016/0140-3664(90)90032-C.

Rahouma, K.H., Abdul-Karim, M.S. and Nasr, K.S. (2020) ‘TCP/IP network layers and their protocols (A Survey)’, in *Internet of Things—Applications and Future: Proceedings of ITAF 2019*. Singapore: Springer Singapore, pp. 287–323.

Rajaraman, V. (2022) ‘A Concise History of the Internet—I’, *Resonance*, 27(11), pp. 1841–1856. doi:10.1007/s12045-022-1483-2.

Rico, D. and Merino, P. (2020) ‘A survey of end-to-end solutions for reliable low-latency communications in 5G networks’, *IEEE Access*, 8, pp. 192808–192834.

Russell, A.L., Pelkey, J.L. and Robbins, L. (2022) ‘The Business of Internetworking: Standards, Start-Ups, and Network Effects’, *Business History Review*, 96(1), pp. 109–144. doi:10.1017/S000768052100074X.

Sadiku, M. N. O. and Akujuobi, C.M. (2022) ‘Digital Communications’, in *Fundamentals of Computer Networks*. Cham: Springer International Publishing, pp. 7–18. doi:10.1007/978-3-031-09417-0_2.

Sadiku, Matthew N. O. and Akujuobi, C.M. (2022) *Fundamentals of Computer Networks*. Cham: Springer International Publishing. doi:10.1007/978-3-031-09417-0.

Solomon, M.G. and Kim, D. (2021) *Fundamentals of Communications and Networking*. Burlington: Jones & Bartlett Learning.

Suresh, P. (2016) ‘Survey on Seven Layered Architecture of OSI Model’, *International Journal of Research in Computer Applications and Robotics (IJRCAR)*, 4(8), pp. 1–10.

Timothy Murkomen (2024) ‘Performance, privacy, and security issues of TCP/IP at the application layer: A comprehensive survey’, *GSC Advanced Research and Reviews*, 18(3), pp. 234–264. doi:10.30574/gscarr.2024.18.3.0106.