

TCP/IP Protocol Architecture and Its Layers

TCP/IP was designed and developed by the Department of Defense (DoD) in the 1960s and is based on standard protocols. It stands for Transmission Control Protocol/Internet Protocol. The [TCP/IP model](#) is a concise version of the OSI model. It contains four layers, unlike the seven layers in the OSI model.

The number of layers is sometimes referred to as five or four. Here In this article, we'll study five layers. The [Physical Layer](#) and [Data Link Layer](#) are referred to as one single layer as the 'Physical Layer' or 'Network Interface Layer' in the 4-layer reference.

What Does TCP/IP Do?

The main work of TCP/IP is to transfer the data of a computer from one device to another. The main condition of this process is to make data reliable and accurate so that the receiver will receive the same information which is sent by the sender. To ensure that, each message reaches its final destination accurately, the TCP/IP model divides its data into packets and combines them at the other end, which helps in maintaining the accuracy of the data while transferring from one end to another end.

What is the Difference between TCP and IP?

[TCP](#) and [IP](#) are different protocols of Computer Networks. The basic difference between TCP (Transmission Control Protocol) and IP (Internet Protocol) is in the transmission of data. In simple words, IP finds the destination of the mail and TCP has the work to send and receive the mail. UDP is another protocol, which does not require IP to communicate with another computer. IP is required by only TCP. This is the basic difference between TCP and IP.

How Does the TCP/IP Model Work?

Whenever we want to send something over the internet using the TCP/IP Model, the TCP/IP Model divides the data into packets at the sender's end and the same packets have to be recombined at the receiver's end to form the same data, and this thing happens to maintain the accuracy of the data. TCP/IP model divides the data into a 4-layer procedure, where the data first

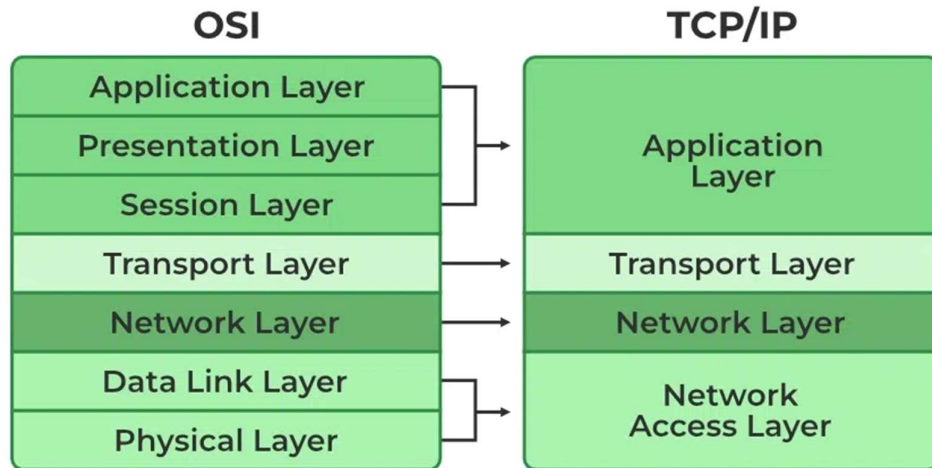
go into this layer in one order and again in reverse order to get organized in the same way at the receiver's end.

For more, you can refer to [TCP/IP in Computer Networking](#).

Layers of TCP/IP Model

1. Application Layer
2. [Transport Layer\(TCP/UDP\)](#)
3. Network/Internet Layer(IP)
4. [Data Link Layer \(MAC\)](#)
5. Physical Layer

The diagrammatic comparison of the **TCP/IP** and **OSI** model is as follows:



TCP/IP and OSI

1. Physical Layer

It is a group of applications requiring network communications. This layer is responsible for generating the data and requesting connections. It acts on behalf of the sender and the Network Access layer on the behalf of the receiver. During this article, we will be talking on the behalf of the receiver.

2. Data Link Layer

The packet's network protocol type, in this case, TCP/IP, is identified by the data-link layer. Error prevention and "framing" are also provided by the data-

link layer. [Point-to-Point Protocol \(PPP\)](#) framing and Ethernet IEEE 802.2 framing are two examples of data-link layer protocols.

3. Internet Layer

This layer parallels the functions of OSI's Network layer. It defines the protocols which are responsible for the logical transmission of data over the entire network. The main protocols residing at this layer are as follows:

- **IP:** [IP](#) stands for Internet Protocol and it is responsible for delivering packets from the source host to the destination host by looking at the IP addresses in the packet headers. IP has 2 versions: IPv4 and IPv6. IPv4 is the one that most websites are using currently. But IPv6 is growing as the number of IPv4 addresses is limited in number when compared to the number of users.
- **ICMP:** [ICMP](#) stands for Internet Control Message Protocol. It is encapsulated within IP datagrams and is responsible for providing hosts with information about network problems.
- **ARP:** [ARP](#) stands for Address Resolution Protocol. Its job is to find the hardware address of a host from a known IP address. ARP has several types: Reverse ARP, Proxy ARP, Gratuitous ARP, and Inverse ARP.

The Internet Layer is a layer in the Internet Protocol (IP) suite, which is the set of protocols that define the Internet. The Internet Layer is responsible for routing packets of data from one device to another across a network. It does this by assigning each device a unique IP address, which is used to identify the device and determine the route that packets should take to reach it.

Example: Imagine that you are using a computer to send an email to a friend. When you click “send,” the email is broken down into smaller packets of data, which are then sent to the Internet Layer for routing. The Internet Layer assigns an IP address to each packet and uses routing tables to determine the best route for the packet to take to reach its destination. The packet is then forwarded to the next hop on its route until it reaches its destination. When all of the packets have been delivered, your friend’s computer can reassemble them into the original email message.

In this example, the Internet Layer plays a crucial role in delivering the email from your computer to your friend’s computer. It uses IP addresses and routing tables to determine the best route for the packets to take, and it

ensures that the packets are delivered to the correct destination. Without the Internet Layer, it would not be possible to send data across the Internet.

4. Transport Layer

The TCP/IP transport layer protocols exchange data receipt acknowledgments and retransmit missing packets to ensure that packets arrive in order and without error. End-to-end communication is referred to as such. Transmission Control Protocol (TCP) and User Datagram Protocol are transport layer protocols at this level (UDP).

- **TCP:** Applications can interact with one another using [TCP](#) as though they were physically connected by a circuit. TCP transmits data in a way that resembles character-by-character transmission rather than separate packets. A starting point that establishes the connection, the whole transmission in byte order, and an ending point that closes the connection make up this transmission.
- **UDP:** The datagram delivery service is provided by [UDP](#), the other transport layer protocol. Connections between receiving and sending hosts are not verified by UDP. Applications that transport little amounts of data use UDP rather than TCP because it eliminates the processes of establishing and validating connections.

5. Application Layer

This layer is analogous to the transport layer of the OSI model. It is responsible for end-to-end communication and error-free delivery of data. It shields the upper-layer applications from the complexities of data. The three main protocols present in this layer are:

- **HTTP and HTTPS:** [HTTP](#) stands for Hypertext transfer protocol. It is used by the World Wide Web to manage communications between web browsers and servers. HTTPS stands for HTTP-Secure. It is a combination of HTTP with SSL(Secure Socket Layer). It is efficient in cases where the browser needs to fill out forms, sign in, authenticate, and carry out bank transactions.
- **SSH:** [SSH](#) stands for Secure Shell. It is a terminal emulations software similar to Telnet. The reason SSH is preferred is because of its ability to maintain the encrypted connection. It sets up a secure session over a TCP/IP connection.

- **NTP:** [NTP](#) stands for Network Time Protocol. It is used to synchronize the clocks on our computer to one standard time source. It is very useful in situations like bank transactions. Assume the following situation without the presence of NTP. Suppose you carry out a transaction, where your computer reads the time at 2:30 PM while the server records it at 2:28 PM. The server can crash very badly if it's out of sync.

The host-to-host layer is a layer in the OSI (Open Systems Interconnection) model that is responsible for providing communication between hosts (computers or other devices) on a network. It is also known as the transport layer.

Some common use cases for the host-to-host layer include:

1. **Reliable Data Transfer:** The host-to-host layer ensures that data is transferred reliably between hosts by using techniques like error correction and flow control. For example, if a packet of data is lost during transmission, the host-to-host layer can request that the packet be retransmitted to ensure that all data is received correctly.
2. **Segmentation and Reassembly:** The host-to-host layer is responsible for breaking up large blocks of data into smaller segments that can be transmitted over the network, and then reassembling the data at the destination. This allows data to be transmitted more efficiently and helps to avoid overloading the network.
3. **Multiplexing and Demultiplexing:** The host-to-host layer is responsible for multiplexing data from multiple sources onto a single network connection, and then demultiplexing the data at the destination. This allows multiple devices to share the same network connection and helps to improve the utilization of the network.
4. **End-to-End Communication:** The host-to-host layer provides a connection-oriented service that allows hosts to communicate with each other end-to-end, without the need for intermediate devices to be involved in the communication.

Example: Consider a network with two hosts, A and B. Host A wants to send a file to host B. The host-to-host layer in host A will break the file into smaller segments, add error correction and flow control information, and then transmit the segments over the network to host B. The host-to-host layer in host B will receive the segments, check for errors, and reassemble the file.

Once the file has been transferred successfully, the host-to-host layer in host B will acknowledge receipt of the file to host A.

In this example, the host-to-host layer is responsible for providing a reliable connection between host A and host B, breaking the file into smaller segments, and reassembling the segments at the destination. It is also responsible for multiplexing and demultiplexing the data and providing end-to-end communication between the two hosts.

Other Common Internet Protocols

TCP/IP Model covers many Internet Protocols. The main rule of these Internet Protocols is how the data is validated and sent over the Internet. Some Common Internet Protocols include:

- **HTTP (Hypertext Transfer Protocol):** [HTTP](#) takes care of Web Browsers and Websites.
- **FTP (File Transfer Protocol):** [FTP](#) takes care of how the file is to be sent over the Internet.
- **SMTP (Simple Mail Transfer Protocol):** [SMTP](#) is used to send and receive data.