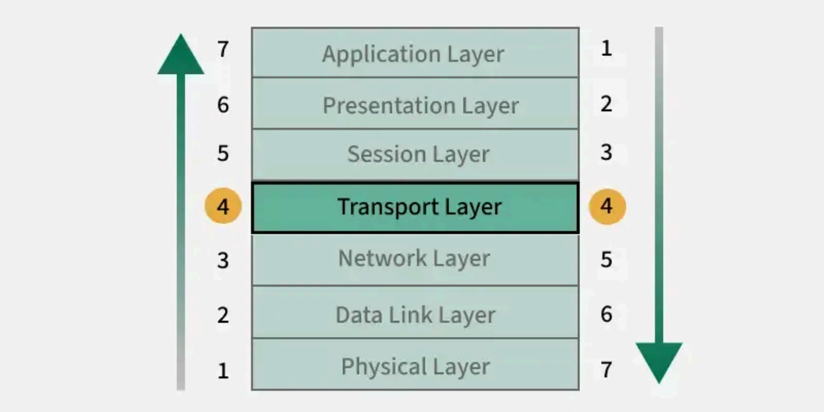
Transport Layer in OSI Model

Last Updated : 31 Jan, 2025

The transport layer, or layer 4 of the OSI model, controls network traffic between hosts and end systems to guarantee full data flows.

It is positioned between the network and session layers in the OSI paradigm. The data packets must be taken and sent to the appropriate machine by the network layer. After that, the transport layer receives the packets, sorts them, and looks for faults. Subsequently, it directs them to the session layer of the appropriate computer program. Now, the properly structured packets are used by the session layer to hold the data for the application.

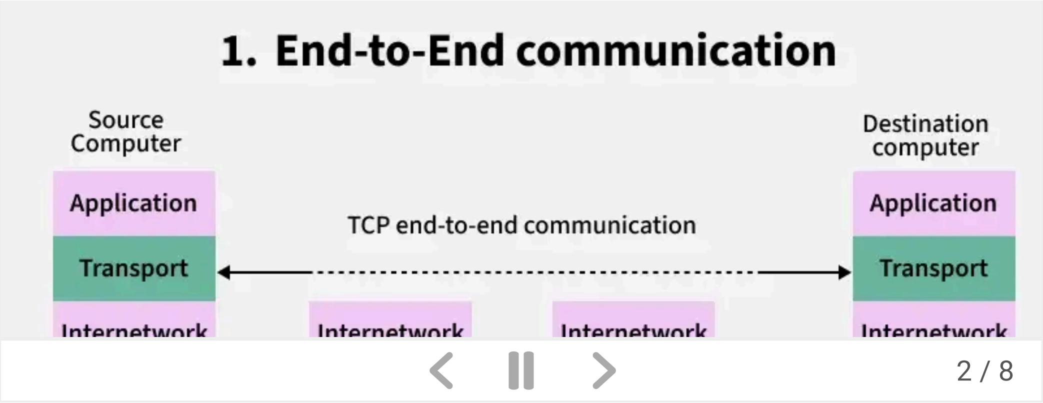


Transport Layer in OSI Model

Functions of Transport Layer

The Transport Layer is responsible for end-to-end communication of data packets. It provides a number of important functions that are responsible for reliable, efficient, and organized data transfer between host systems in a networked environment.

The primary functions of the Transport Layer are:

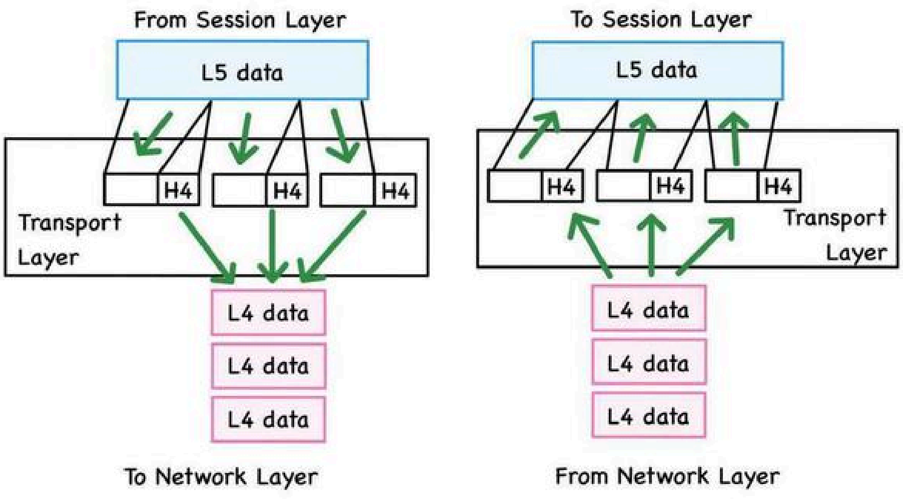


1. End-to-End communication

To read in detail about services offered by transport layer, refer to [Transport Layer Services]

Working of Transport Layer

Communication between end systems is dependable and effective thanks to the Transport Layer. Apart from regulating flow and accommodating numerous applications concurrently, it guarantees data delivery in a manner that guarantees accuracy and minimises mistakes. It accomplishes this by utilising a collection of methods and protocols that provide data transport.



- The primary function of the transport layer is to give application processes operating on several hosts direct access to communication services.
- Logical communication between application processes operating on separate hosts is facilitated by the transport layer. Application processes use the logical communication offered by the transport

layer to deliver messages to one other even when they are running on different hosts and are not physically connected.

- The network routers do not implement the transport layer protocols; only the end systems do.
- For instance, the network layer receives services from TCP and UDP, two transport layer protocols, which offer distinct functionalities.
- Protocols at the transport layer offer multiplexing and demultiplexing capabilities. In addition, it offers other services including bandwidth assurances, latency guarantees, and dependable data transport.
- Every application at the application layer is capable of sending a message via either TCP or UDP. Either of these two protocols can be used by the application to interact. The internet protocol on the internet layer will then be communicated with by both TCP and UDP. The transport layer is readable and writeable by the applications.

Transport Layer Protocols

Transport Layer Protocol uses different protocol for the better communication between two ends uses of protocol may differ from specifications. Below mention are some protocols used in Transport Layer

1. Transmission Control Protocol(TCP)

- TCP is connection-oriented Protocol.
- TCP is reliable protocol.
- As TCP is connection-oriented protocol, so first the connection is established between two ends and then data is transferred and then the connection is terminated after all data being sent.

2. User Datagram Protocol (UDP)

- UDP is not reliable protocol
- The protocol UDP is connectionless.
- When speed and size are more important than security and dependability, this kind of protocol is employed.

- The data from the higher layer is supplemented with transport-level addresses, checksum error control, and length information by UDP, an end-to-end transport level protocol.
- A user datagram is the packet that the UDP protocol generates.

3. Stream Control Transmission Protocol (SCTP)

- Many Internet applications use SCTP to perform transport layer duties, similar to User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).
- On top of a connectionless packet network like IP, SCTP is a dependable transport protocol that facilitates data transfer over the network in scenarios involving one or more IP addresses.

Difference Between TCP and UDP at Transport Layer

TCP	UDP
TCP is a connection-oriented protocol	UDP is the connection-less protocol
TCP is reliable.	UDP is not reliable.
TCP supports error-checking mechanisms.	UDP has only the basic error-checking mechanism using checksums.
An acknowledgment segment is present.	No acknowledgment segment.
TCP is slower than UDP	UDP is faster, simpler, and more efficient than TCP.

TCP	UDP
Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP)
TCP has a (20-60) bytes variable length header.	The header length is fixed of 8 bytes.

For practice, solve quiz on Transport Layer.

Comment More info
Advertise with us

Next Article
Difference Between Hub and Repeater

Similar Reads

Network Layer in OSI Model

The Network Layer is the 5th Layer from the top and the 3rd layer from the Bottom of the OSI Model. It is one of the most important layers whic...

15+ min read

Transport Layer responsibilities

The transport Layer is the second layer in the TCP/IP model and the fourth layer in the OSI model. It is an end-to-end layer used to deliver...

15+ min read

Transport Layer Protocols

The transport layer is the fourth layer in the OSI model and the second layer in the TCP/IP model. The transport layer provides with end to end...

Open In App

15+ min read

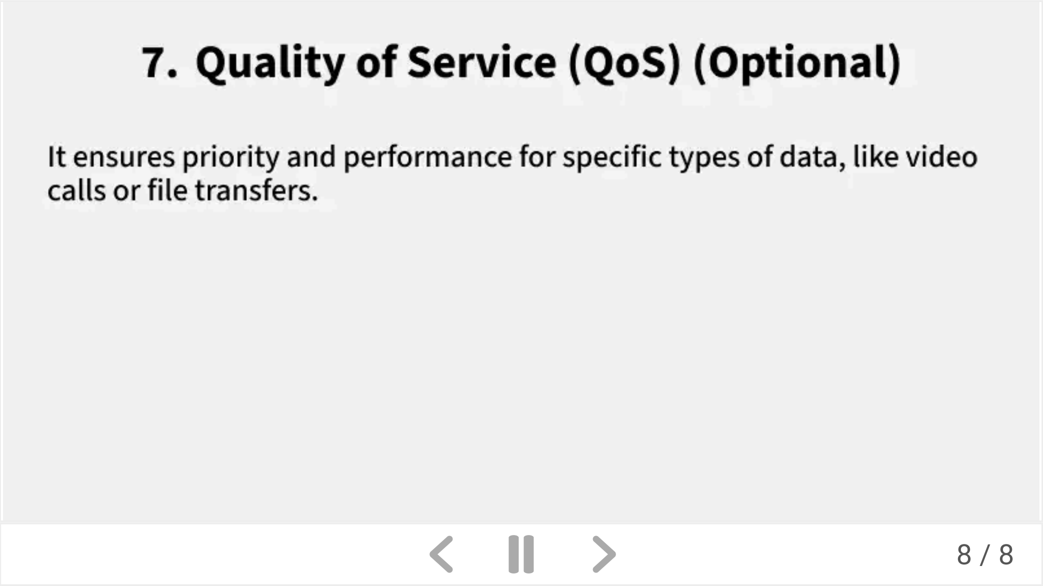
Transport Layer Services

Last Updated : 29 Jan, 2025

The transport Layer is the second layer in the TCP/IP model and the fourth layer in the OSI model. It is an end-to-end layer used to deliver messages to a host. It is termed an end-to-end layer because it provides a point-to-point connection rather than hop-to-hop, between the source host and destination host to deliver the services reliably. The unit of data encapsulation in the Transport Layer is a segment.

Services Offered by Transport Layer

The transport layer provides reliable data transfer services such as segmentation, flow control, error detection, and retransmission, end-to-end communication between devices.

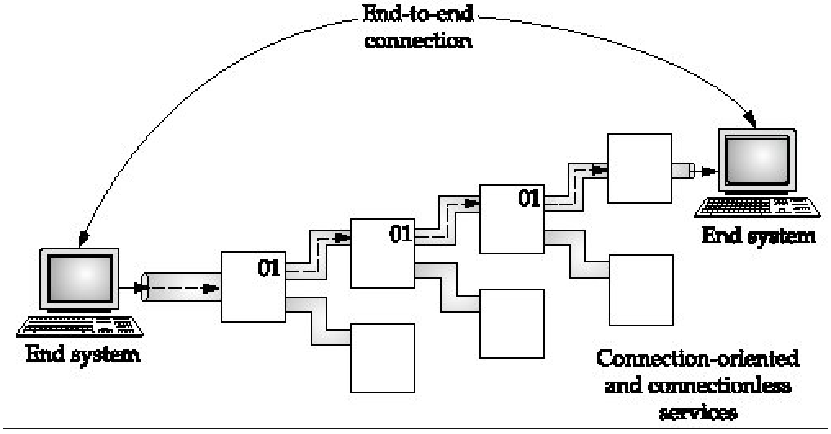


7. Quality of Service (QoS) (Optional)

It ensures priority and performance for specific types of data, like video calls or file transfers.

1.End-to-end Connection between Hosts

The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP. TCP is a secure, connection-orientated protocol that uses a handshake protocol to establish a robust connection between two end hosts. TCP ensures the reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol that ensures best-effort delivery. It is suitable for applications that have little concern with flow or error control and requires sending the bulk of data like video conferencing. It is often used in multicasting protocols.



End-to-End Connection

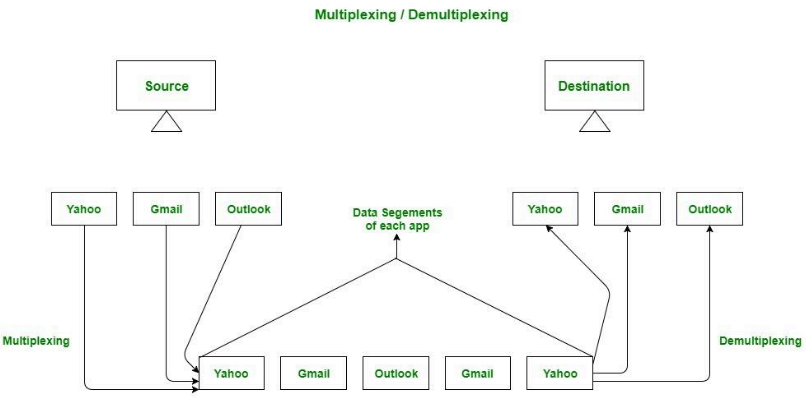
2.Flow Control

The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

Trending Now DSA Web Tech Foundational Courses Data Science Practice Problem

Multiplexing(many to one) is when data is acquired from several processes from the sender and merged into one packet along with headers and sent as a single packet. Multiplexing allows the simultaneous use of different processes over a network that is running on a host. The processes are differentiated by their port numbers. Similarly, Demultiplexing(one to many) is required at the receiver side when the message is distributed into different processes. Transport receives the segments of data from the network layer distributes and delivers it to the appropriate process running on the receiver's machine.

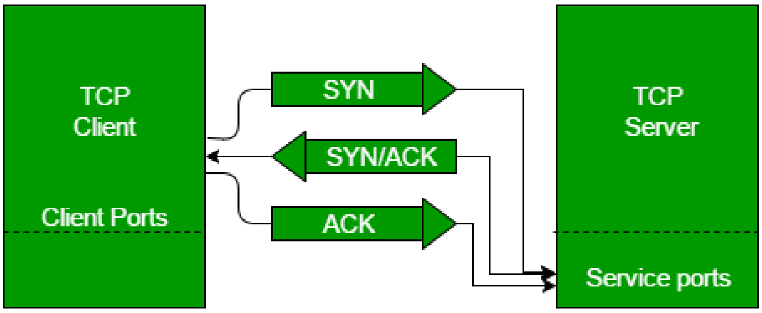
Open In App



Multiplexing and Demultiplexing

4.Connection Establishment

When two devices in a network wants to establish a connection using TCP, it is done through 3-Way Handshake Process.



- The first computer connects to the second computer by sending a SYN packet to a specified port number.
- If the second computer is listening, it will respond with a SYN/ACK.
- When the first computer receives the SYN/ACK, it replies with an ACK packet.
- After this, the two devices can communicate normally.

5.Connection Termination

In a TCP connection, we have two types of termination mechanisms:

1. In the Graceful connection release, the connection is open until both parties have closed their sides of the connection.
2. In an Abrupt connection release, either one TCP entity is forced to close the connection or one user closes both directions of data transfer.

Read more about [TCP Connection Termination].

6. Reliable Data Delivery

The transport layer checks for errors in the messages coming from the application layer by using error detection codes, and computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.

Limitations of Transport Layer Services

- The Transport Layer does not handle routing or addressing between the source and destination. It rely on the Network Layer for this.
- Transport layer do not have features like encryption and authentication unless protocols like TLS are used.
- Reliable transmission methods like TCP add extra overhead which can slow down communication.
- The Transport Layer depends on the Application Layer for specific needs and the lower layers for handling network failure.

What are the main protocols used in the Transport Layer?

The two most common protocols are

- ***TCP (Transmission Control Protocol)*** *for reliable communication*
- ***UDP (User Datagram Protocol)*** *for faster connectionless transmission.*

What is the difference between flow control and congestion control?

Flow control *prevents the sender from overwhelming the receiver, while* ***congestion control*** *prevents excessive data from causing network congestion.*

What is the role of ports in the Transport Layer?

Ports help identify specific applications on a device. It allows multiple services (like web browsing and email) to run

What is TCP (Transmission Control Protocol)?

Last Updated : 01 Feb, 2025

Transmission Control Protocol (TCP) is a **connection-oriented protocol for communications** that helps in the exchange of messages between different devices over a network. It is one of the main protocols of the TCP/IP suite. In OSI model, it operates at the transport layer(Layer 4). It lies between the Application and Network Layers which are used in providing reliable delivery services. The Internet Protocol (IP), which establishes the technique for sending data packets between computers, works with TCP.

- TCP establishes a reliable connection between sender and receiver using the **three-way handshake (SYN, SYN-ACK, ACK)** and it uses a **four-step handshake (FIN, ACK, FIN, ACK)** to close connections properly.
- It ensures **error-free, in-order delivery** of data packets.
- It uses **acknowledgments (ACKs)** to confirm receipt.
- It prevents data overflow by adjusting the data transmission rate according to the receiver’s buffer size.
- It prevents network congestion using algorithms like **Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery.**
- TCP header uses **checksum** to detect corrupted data and requests retransmission if needed.
- It is used in applications requiring **reliable** and **ordered** data transfer, such as web browsing, email, and remote login.

Internet Protocol (IP)

Internet Protocol (IP) is a method that is useful for sending data from one device to another from all over the internet. It is a set of rules governing how data is sent and received over the internet. It is responsible for addressing and routing packets of data so they can

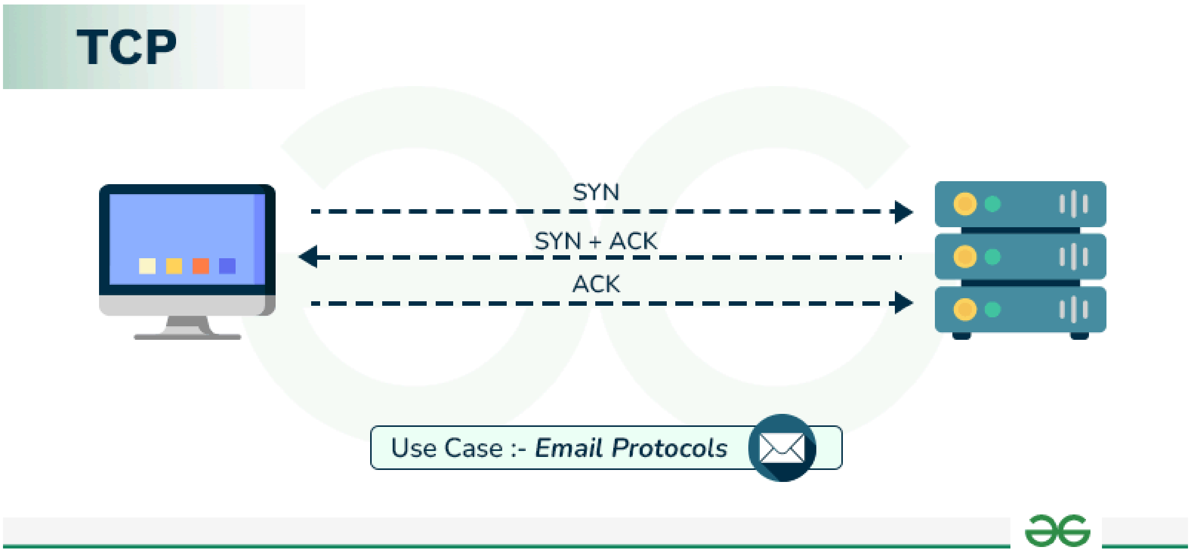
Open In App

travel from the sender to the correct destination across multiple networks. Every device contains a unique IP Address that helps it communicate and exchange data across other devices present on the internet.

Working of Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP) model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end to make sure that each message reaches its target location intact. Sending the information in little bundles of information makes it simpler to maintain efficiency as opposed to sending everything in one go.

After a particular message is broken down into bundles, these bundles may travel along multiple routes if one route is jammed but the destination remains the same.



TCP

For Example: When a user requests a web page on the internet, somewhere in the world, the server processes that request and sends back an HTML Page to that user. The server makes use of a protocol called the HTTP Protocol. The HTTP then requests the TCP layer to set the required connection and send the HTML file.

Now, the TCP breaks the data into small packets and forwards it toward the Internet Protocol (IP) layer. The packets are then sent to the destination through different routes.

The TCP layer in the user's system waits for the transmission to get finished and acknowledges once all packets have been received.

Features of TCP

Some of the most prominent features of Transmission control protocol are mentioned below.

- **Segment Numbering System:** TCP keeps track of the segments being transmitted or received by assigning numbers to each and every single one of them. A specific Byte Number is assigned to data bytes that are to be transferred while segments are assigned sequence numbers. Acknowledgment Numbers are assigned to received segments.
- **Connection Oriented:** It means sender and receiver are connected to each other till the completion of the process. The order of the data is maintained i.e. order remains same before and after transmission.
- **Full Duplex:** In TCP data can be transmitted from receiver to the sender or vice – versa at the same time. It increases efficiency of data flow between sender and receiver.
- **Flow Control:** Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery. The receiver continually hints to the sender on how much data can be received (using a sliding window).
- **Error Control:** TCP implements an error control mechanism for reliable data transfer. Error control is byte-oriented. Segments are checked for error detection. Error Control includes – Corrupted Segment & Lost Segment Management, Out-of-order segments, Duplicate segments, etc.
- **Congestion Control:** TCP takes into account the level of congestion in the network. Congestion level is determined by the amount of data sent by a sender.

Advantages of TCP

- It is a reliable protocol.
- It provides an error-checking mechanism as well as one for recovery.
- It gives flow control.

- It makes sure that the data reaches the proper destination in the exact order that it was sent.
- It is a well-documented and widely implemented protocol, maintained by standards organizations like the IETF (Internet Engineering Task Force).
- It works in conjunction with IP (Internet Protocol) to establish connections between devices on a network.

Disadvantages of TCP

- TCP is made for Wide Area Networks, thus its size can become an issue for small networks with low resources.
- TCP runs several layers so it can slow down the speed of the network.
- It is not generic in nature. It cannot represent any protocol stack other than the TCP/IP suite. E.g., it cannot work with a Bluetooth connection.
- No modifications since their development around 30 years ago.

Comment More info

Advertise with us

Next Article

TCP 3-Way Handshake Process

Similar Reads

TCP 3-Way Handshake Process

The TCP 3-Way Handshake is a fundamental process that establishes a reliable connection between two devices over a TCP/IP network. It...

15+ min read

Differences between TCP and UDP

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) both are protocols of the Transport Layer protocols. TCP is a connectio...

Open In App

TCP/IP Model

Last Updated : 05 Feb, 2025

The TCP/IP model is a fundamental framework for computer networking. It stands for Transmission Control Protocol/Internet Protocol, which are the core protocols of the Internet. This model defines how data is transmitted over networks, ensuring reliable communication between devices. It consists of four layers: the Link Layer, the Internet Layer, the Transport Layer, and the Application Layer. Each layer has specific functions that help manage different aspects of network communication, making it essential for understanding and working with modern networks.

TCP/IP was designed and developed by the Department of Defense (DoD) in the 1960s and is based on standard protocols. The TCP/IP model is a concise version of the OSI model. It contains four layers, unlike the seven layers in the OSI model. In this article, we are going to discuss the TCP/IP model in detail.

TCP/IP model was developed alongside the creation of the ARPANET, which later became the foundation of the modern internet. It was designed with a focus on the practical aspects of networking at the time. The lower-level hardware details and physical transmission medium were largely abstracted away in favor of higher-level networking protocols.

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. The TCP/IP model is used in the context of the real-world internet, where a wide range of physical media and network technologies are in use. Rather than specifying a particular Physical Layer, the TCP/IP model allows for flexibility in adapting to different physical implementations.

Difference Between TCP and IP

Feature	TCP (Transmission Control Protocol)	IP (Internet Protocol)
Purpose	Ensures reliable, ordered, and error-checked delivery of data between applications.	Provides addressing and routing of packets across networks.
Type	Connection-oriented	Connectionless
Function	Manages data transmission between devices, ensuring data integrity and order.	Routes packets of data from the source to the destination based on IP addresses.
Error Handling	Yes, includes error checking and recovery mechanisms.	No, IP itself does not handle errors; relies on upper-layer protocols like TCP.
Flow Control	Yes, includes flow control mechanisms.	No
Congestion Control	Yes, manages network congestion.	No
Data Segmentation	Breaks data into smaller packets and	Breaks data into packets but does not

Feature	TCP (Transmission Control Protocol)	IP (Internet Protocol)
	reassembles them at the destination.	handle reassembly.
Header Size	Larger, 20-60 bytes	Smaller, typically 20 bytes
Reliability	Provides reliable data transfer	Does not guarantee delivery, reliability, or order.
Transmission Acknowledgment	Yes, acknowledges receipt of data packets.	No

How Does the TCP/IP Model Work?

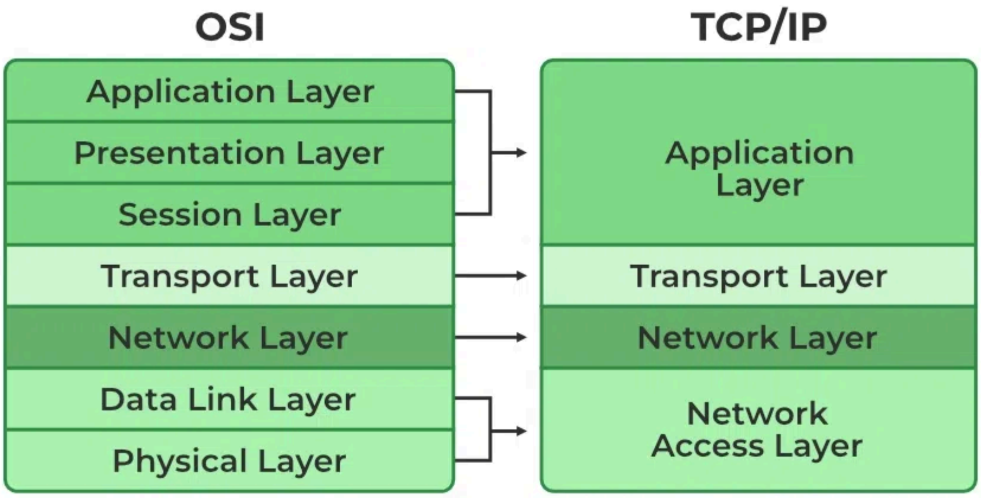
Whenever we want to send something over the internet using the TCP/IP Model, the TCP/IP Model divides the data into packets at the sender's end and the same packets have to be recombined at the receiver's end to form the same data, and this thing happens to maintain the accuracy of the data. TCP/IP model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

- Application Layer
- Transport Layer(TCP/UDP)
- Network/Internet Layer(IP)
- Network Access Layer

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



TCP/IP and OSI

1. Network Access Layer

The Network Access Layer represents a collection of applications that require network communication. This layer is responsible for generating data and initiating connection requests. It operates on behalf of the sender to manage data transmission, while the Network Access layer on the receiver's end processes and manages incoming data. In this article, we will focus on its role from the receiver's perspective.

The packet's network protocol type, in this case, TCP/IP, is identified by network access layer. Error prevention and "framing" are also provided by this layer. Point-to-Point Protocol (PPP) framing and Ethernet IEEE 802.2 framing are two examples of data-link layer protocols.

2. Internet or Network 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.

3. 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.

4. Application Layer

The Application Layer in the TCP/IP model combines the functions of three layers from the **OSI model**: the **Application**, **Presentation**, and **Session** layers. 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:

- **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.
- **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.
- **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.
- **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.

Why TCP/IP Model Does Not Have Physical Layer

The physical layer is not covered by the TCP/IP model because the data link layer is considered the point at which the interface occurs between the TCP/IP stock and the underlying network hardware. Also, it is designed to be independent of the underlying physical media. This allows TCP/IP to be flexible and adaptable to different types of physical connections, such as Ethernet, Wi-Fi, fiber optics, or even older technologies like dial-up modems. The physical layer is typically handled by hardware components and standards specific to the physical medium being used, like Ethernet cables or radio waves for Wi-Fi.

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.

Difference between TCP/IP and OSI Model

TCP/IP	OSI
TCP refers to Transmission Control Protocol.	OSI refers to Open Systems Interconnection.
TCP/IP uses both the session and presentation layer in the application layer itself.	OSI uses different session and presentation layers.
TCP/IP follows connectionless a horizontal approach.	OSI follows a vertical approach.
The Transport layer in TCP/IP does not provide assurance delivery of packets.	In the OSI model, the transport layer provides assurance delivery of packets.
Protocols cannot be replaced easily in TCP/IP model.	While in the OSI model, Protocols are better covered and are easy to replace with the technology change.
TCP/IP model network layer only provides connectionless (IP) services. The transport layer (TCP) provides connections.	Connectionless and connection-oriented services are provided by the network layer in the OSI model.

Advantages of TCP/IP Model

- **Interoperability** : The TCP/IP model allows different types of computers and networks to communicate with each other, promoting compatibility and cooperation among diverse systems.
- **Scalability** : TCP/IP is highly scalable, making it suitable for both small and large networks, from local area networks (LANs) to wide area networks (WANs) like the internet.
- **Standardization** : It is based on open standards and protocols, ensuring that different devices and software can work together without compatibility issue

- **Flexibility** : The model supports various routing protocols, data types, and communication methods, making it adaptable to different networking needs.
- **Reliability** : TCP/IP includes error-checking and retransmission features that ensure reliable data transfer, even over long distances and through various network conditions.

Disadvantages of TCP/IP Model

- **Complex Configuration** : Setting up and managing a TCP/IP network can be complex, especially for large networks with many devices. This complexity can lead to configuration errors.
- **Security Concerns** : TCP/IP was not originally designed with security in mind. While there are now many security protocols available (such as SSL/TLS), they have been added on top of the basic TCP/IP model, which can lead to vulnerabilities.
- **Inefficiency for Small Networks** : For very small networks, the overhead and complexity of the TCP/IP model may be unnecessary and inefficient compared to simpler networking protocols.
- **Limited by Address Space** : Although IPv6 addresses this issue, the older IPv4 system has a limited address space, which can lead to issues with address exhaustion in larger networks.
- **Data Overhead** : TCP, the transport protocol, includes a significant amount of overhead to ensure reliable transmission. This can reduce efficiency, especially for small data packets or in networks where speed is crucial.

Conclusion

In conclusion, the TCP/IP model is the backbone of modern internet communication, allowing different devices and networks to connect and share information reliably. Despite some complexity and security concerns, its flexibility, scalability, and widespread adoption make it essential for both small and large networks. Overall, the TCP/IP model is crucial for ensuring efficient and effective network communication.

Frequently Asked Questions on TCP/IP Model – FAQs

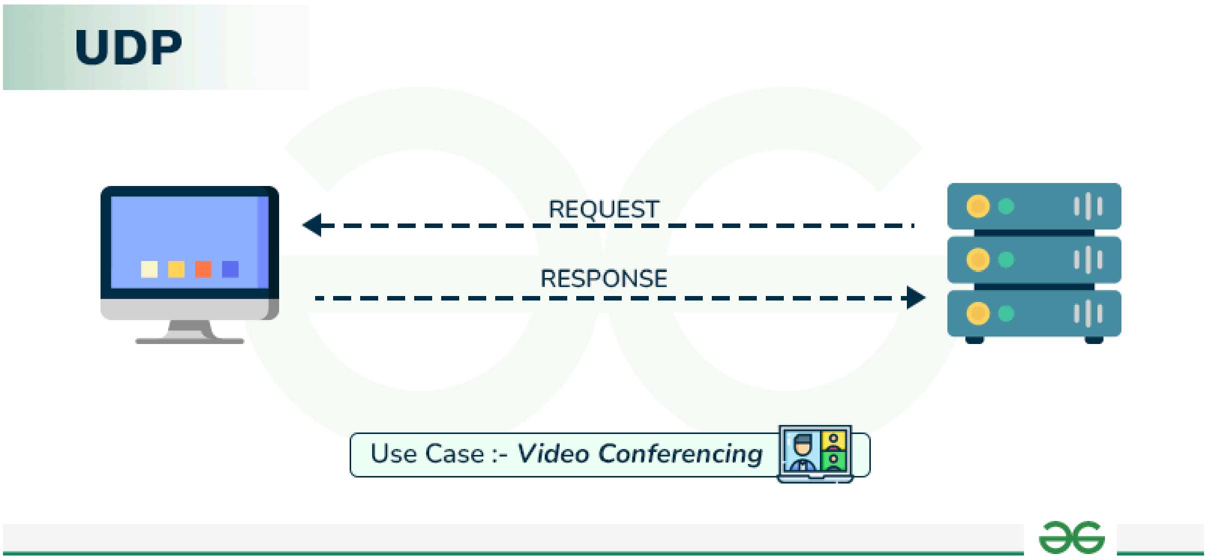
User Datagram Protocol (UDP)

Last Updated : 27 Dec, 2024

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an **unreliable and connectionless protocol.** So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections over the network. The UDP enables process-to-process communication.

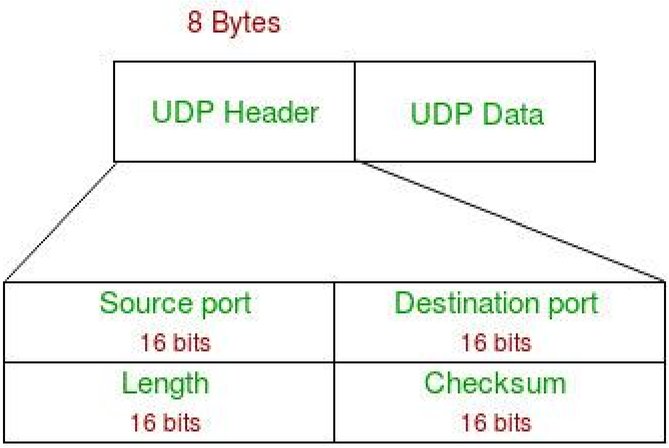
What is User Datagram Protocol?

User Datagram Protocol (UDP) is one of the core protocols of the Internet Protocol (IP) suite. It is a communication protocol used across the internet for time-sensitive transmissions such as video playback or DNS lookups . Unlike Transmission Control Protocol (TCP), UDP is connectionless and does not guarantee delivery, order, or error checking, making it a lightweight and efficient option for certain types of data transmission.



UDP Header

UDP header is an **8-byte** fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contain all necessary header information and the remaining part consists of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.



UDP Header

- **Source Port:** Source Port is a 2 Byte long field used to identify the port number of the source.
- **Destination Port:** It is a 2 Byte long field, used to identify the port of the destined packet.
- **Length:** Length is the length of UDP including the header and the data. It is a 16-bits field.
- **Checksum:** Checksum is 2 Bytes long field. It is the 16-bit one's complement of the one's complement sum of the UDP header, the pseudo-header of information from the IP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Notes – Unlike TCP, the Checksum calculation is not mandatory in UDP. No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting. Also UDP provides port numbers so that is can differentiate between users requests.

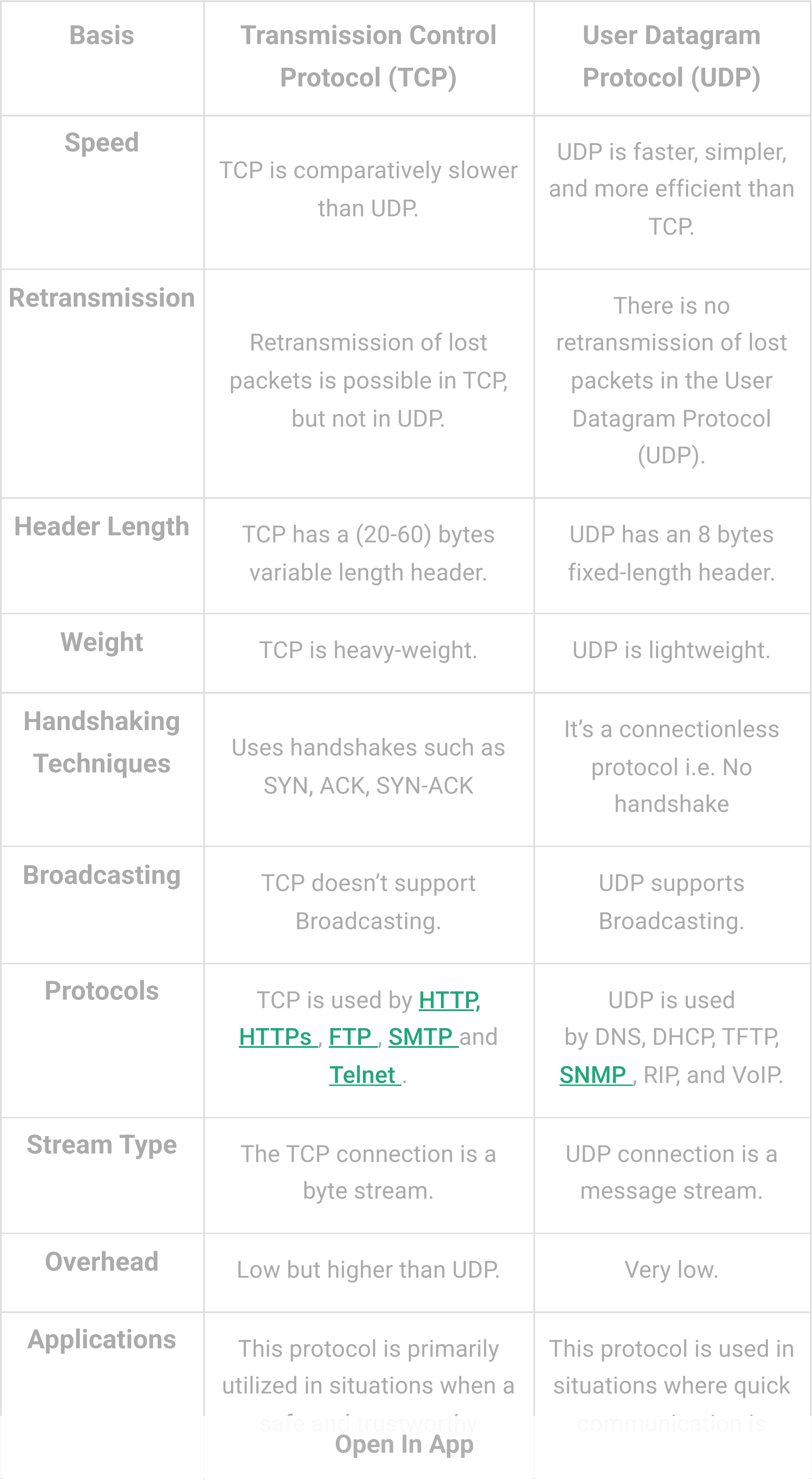
Applications of UDP

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.

- It is a suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP(Routing Information Protocol).
- Normally used for real-time applications which can not tolerate uneven delays between sections of a received message.
- VoIP (Voice over Internet Protocol) services, such as Skype and WhatsApp, use UDP for real-time voice communication. The delay in voice communication can be noticeable if packets are delayed due to congestion control, so UDP is used to ensure fast and efficient data transmission.
- DNS (Domain Name System) also uses UDP for its query/response messages. DNS queries are typically small and require a quick response time, making UDP a suitable protocol for this application.
- DHCP (Dynamic Host Configuration Protocol) uses UDP to dynamically assign IP addresses to devices on a network. DHCP messages are typically small, and the delay caused by packet loss or retransmission is generally not critical for this application.
- Following implementations uses UDP as a transport layer protocol:
  - NTP (Network Time Protocol)
  - DNS (Domain Name Service)
  - BOOTP, DHCP.
  - NNP (Network News Protocol)
  - Quote of the day protocol
  - TFTP, RTSP, RIP.
- The application layer can do some of the tasks through UDP-
  - Trace Route
  - Record Route
  - Timestamp
- UDP takes a datagram from Network Layer , attaches its header, and sends it to the user. So, it works fast.

TCP vs UDP

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.



Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn't support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs , FTP , SMTP and Telnet .	UDP is used by DNS, DHCP, TFTP, SNMP , RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy	This protocol is used in situations where quick communication is

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
	communication procedure is necessary, such as in email, on the web surfing, and in military services.	necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.

Advantages of UDP

- **Speed:** UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.
- Lower latency: Since there is no connection establishment, there is lower latency and faster response time.
- **Simplicity:** UDP has a simpler protocol design than TCP, making it easier to implement and manage.
- **Broadcast support:** UDP supports broadcasting to multiple recipients, making it useful for applications such as video streaming and online gaming.
- **Smaller packet size:** UDP uses smaller packet sizes than TCP, which can reduce network congestion and improve overall network performance.
- User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

Disadvantages of UDP

- **No reliability:** UDP does not guarantee delivery of packets or order of delivery, which can lead to missing or duplicate data.
- **No congestion control:** UDP does not have congestion control, which means that it can send packets at a rate that can cause network congestion.
- **Vulnerable to attacks:** UDP is vulnerable to denial-of-service attacks , where an attacker can flood a network with UDP packets.

overwhelming the network and causing it to crash.

- **Limited use cases:** UDP is not suitable for applications that require reliable data delivery, such as email or file transfers, and is better suited for applications that can tolerate some data loss, such as video streaming or online gaming.

How is UDP used in DDoS attacks?

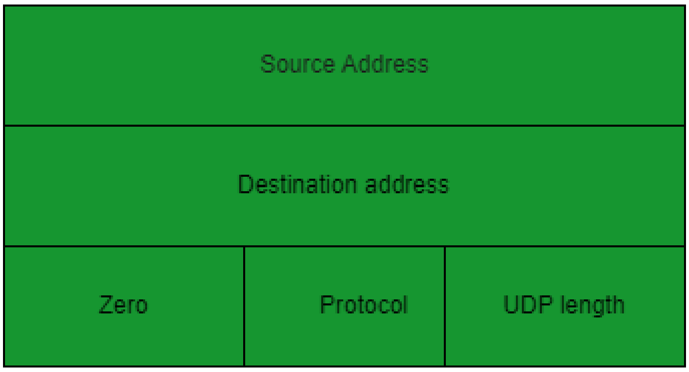
A **UDP flood attack** is a type of **Distributed Denial of Service (DDoS)** attack where an attacker sends a large number of **User Datagram Protocol (UDP)** packets to a target port.

- **UDP Protocol** : Unlike TCP, UDP is connectionless and doesn't require a handshake before data transfer. When a UDP packet arrives at a server, it checks the specified port for listening applications. If no app is found, the server sends an **ICMP "destination unreachable"** packet to the supposed sender (usually a random bystander due to spoofed IP addresses).
- **Attack Process** :
    - The attacker sends UDP packets with spoofed IP sender addresses to random ports on the target system.
    - The server checks each incoming packet's port for a listening application (usually not found due to random port selection).
    - The server sends ICMP "destination unreachable" packets to the spoofed sender (random bystanders).
    - The attacker floods the victim with UDP data packets, overwhelming its resources.
- **Mitigation** : To protect against UDP flood attacks, monitoring network traffic for sudden spikes and implementing security measures are crucial. Organizations often use specialized tools and services to detect and mitigate such attacks effectively.

UDP Pseudo Header

- The purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination

- The correct destination consist of a specific machine and a specific protocol port number within that machine



UDP pseudo header

UDP Pseudo Header Details

- The UDP header itself specify only protocol port number.thus , to verify the destination UDP on the sending machine computes a checksum that covers the destination IP address as well as the UDP packet.
- At the ultimate destination, UDP software verifies the checksum using the destination IP address obtained from the header of the IP packet that carried the UDP message.
- If the checksum agrees, then it must be true that the packet has reached the intended destination host as well as the correct protocol port within that host.

User Interface

A user interface should allow the creation of new receive ports, receive operations on the receive ports that returns the data octets and an indication of source port and source address, and an operation that allows a datagram to be sent, specifying the data, source and destination ports and address to be sent.

IP Interface

- The UDP module must be able to determine the source and destination internet address and the protocol field from internet header
- One possible UDP/IP interface would return the whole internet datagram including the entire internet header in response to a receive operation
- Such an interface would also allow the UDP to pass a full internet datagram complete with header to the IP to send. the IP would verify certain fields for consistency and compute the internet header checksum.
- The IP interface allows the UDP module to interact with the network layer of the protocol stack, which is responsible for routing and delivering data across the network.
- The IP interface provides a mechanism for the UDP module to communicate with other hosts on the network by providing access to the underlying IP protocol.
- The IP interface can be used by the UDP module to send and receive data packets over the network, with the help of IP routing and addressing mechanisms.

GATE Questions for Practice

- GATE CS 2013, Question 12
- GATE CS 2012, Question 65
- GATE CS 2007, Question 20
- GATE CS 2005, Question 23
- GATE IT 2008, Question 66
- GATE Mock 2015, Question 5

Conclusion

The User Datagram Protocol (UDP) is an important Transport Layer protocol in the Internet Protocol (IP) suite, identified for its speed and efficiency due to its connectionless and lightweight design. While UDP lacks TCP’s stability and error-checking features, it used in applications that need low latency and real-time performance, such as streaming, online gaming, and DNS lookups. [illegible] simplicity and support for

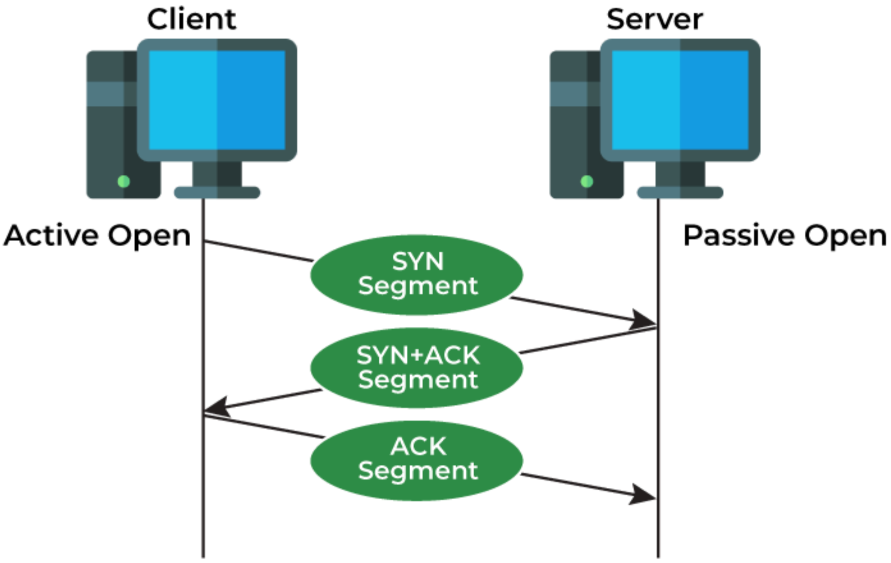
Differences between TCP and UDP

Last Updated : 27 Dec, 2024

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) both are protocols of the Transport Layer Protocols. TCP is a connection-oriented protocol whereas UDP is a part of the Internet Protocol suite, referred to as the UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. In this article, we will discuss the differences between TCP and UDP.

What is Transmission Control Protocol (TCP)?

TCP (Transmission Control Protocol) is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between different devices over a network. The Internet Protocol (IP), which establishes the technique for sending data packets between computers, works with TCP.



Transmission Control Protocol

Features of TCP

- TCP keeps track of the segments being transmitted or received by assigning numbers to every single one of them.
- Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery.
- TCP implements an error control mechanism for reliable data transfer.
- TCP takes into account the level of congestion in the network.

Applications of TCP

- **World Wide Web (WWW)** : When you browse websites, TCP ensures reliable data transfer between your browser and web servers.
- **Email** : TCP is used for sending and receiving emails. Protocols like **SMTP** (Simple Mail Transfer Protocol) handle email delivery across servers.
- **File Transfer Protocol (FTP)** : FTP relies on TCP to transfer large files securely. Whether you're uploading or downloading files, TCP ensures data integrity.
- **Secure Shell (SSH)** : SSH sessions, commonly used for remote administration, rely on TCP for encrypted communication between client and server.
- **Streaming Media** : Services like Netflix, YouTube, and Spotify use TCP to stream videos and music. It ensures smooth playback by managing data segments and retransmissions.

Advantages of TCP

- It is reliable for maintaining a connection between Sender and Receiver.
- It is responsible for sending data in a particular sequence.
- Its operations are not dependent on Operating System .
- It allows and supports many routing protocols.
- It can reduce the speed of data based on the speed of the receiver.

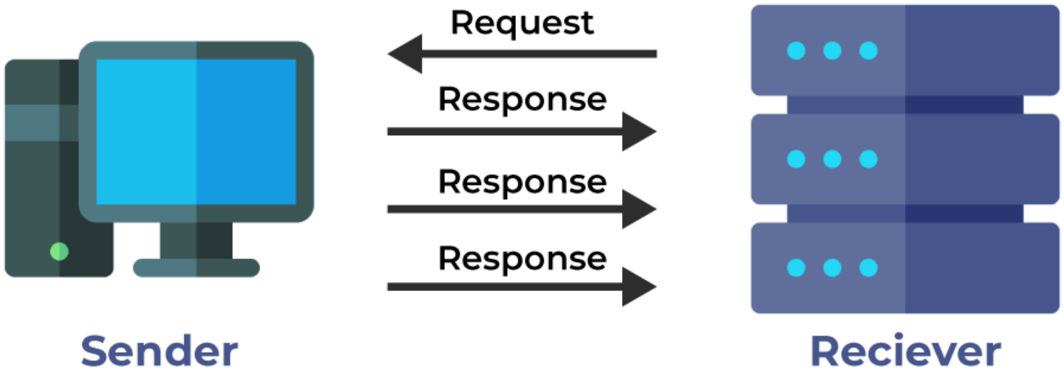
Disadvantages of TCP

- It is slower than UDP and it takes more bandwidth.
- Slower upon starting of tra[illegible]e.

- Not suitable for LAN and PAN Networks.
- It does not have a multicast or broadcast category.
- It does not load the whole page if a single data of the page is missing.

What is User Datagram Protocol (UDP)?

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as the UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections establish over the network. The UDP enables process-to-process communication.



User Datagram Protocol

Features of UDP

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.
- It is a suitable protocol for multicasting as UDP supports packet switching .
- UDP is used for some routing update protocols like RIP(Routing Information Protocol) .
- Normally used for real-time applications which can not tolerate uneven delays between sections of a received message.

Application of UDP

- **Real-Time Multimedia Streaming** : UDP is ideal for streaming audio and video content. Its low-latency nature ensures smooth playback, even if occasional data loss occurs.
- **Online Gaming** : Many online games rely on UDP for fast communication between players.
- **DNS (Domain Name System) Queries** : When your device looks up domain names (like converting “www.example.com” to an IP address), UDP handles these requests efficiently .
- **Network Monitoring** : Tools that monitor network performance often use UDP for lightweight, rapid data exchange.
- **Multicasting** : UDP supports packet switching, making it suitable for multicasting scenarios where data needs to be sent to multiple recipients simultaneously.
- **Routing Update Protocols** : Some routing protocols, like RIP (Routing Information Protocol), utilize UDP for exchanging routing information among routers.

Advantages of UDP

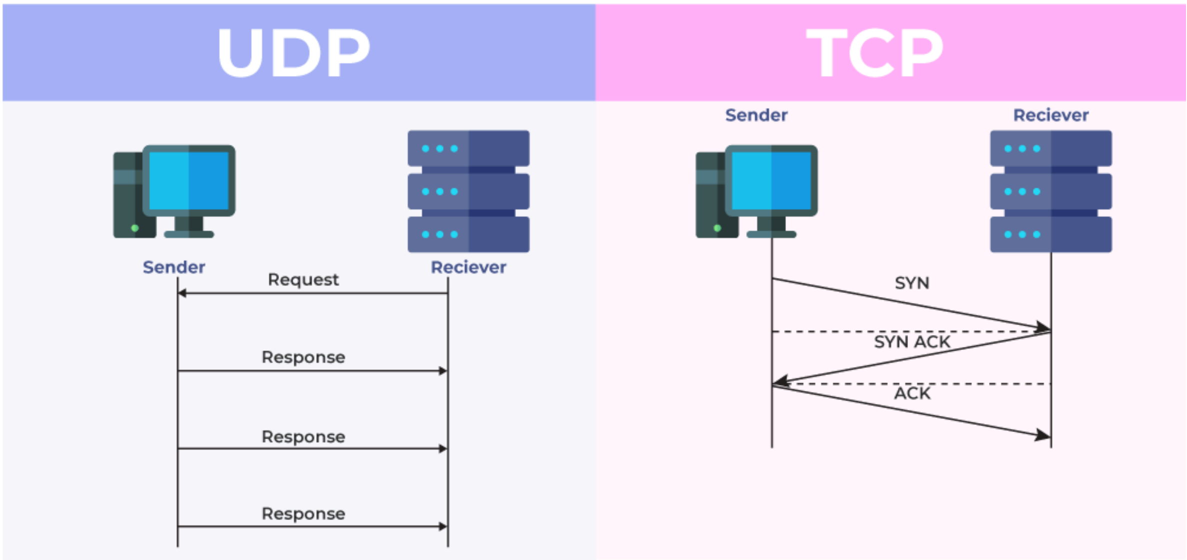
- It does not require any connection for sending or receiving data.
- Broadcast and Multicast are available in UDP.
- UDP can operate on a large range of networks.
- UDP has live and real-time data.
- UDP can deliver data if all the components of the data are not complete.

Disadvantages of UDP

- We can not have any way to acknowledge the successful transfer of data.
- UDP cannot have the mechanism to track the sequence of data.
- UDP is connectionless, and due to this, it is unreliable to transfer data.
- In case of a Collision, UDP packets are dropped by Routers in comparison to TCP.
- UDP can drop packets in case of detection of errors.

Which Protocol is Better: TCP or UDP?

The answer to this question is difficult because it totally depends on what work we are doing and what type of data is being delivered. UDP is better in the case of online gaming as it allows us to work lag-free. TCP is better if we are transferring data like photos, videos, etc. because it ensures that data must be correct has to be sent. In general, both TCP and UDP are useful in the context of the work assigned by us. Both have advantages upon the works we are performing, that's why it is difficult to say, which one is better.



Difference Between TCP and UDP

Where TCP is Used?

- Sending Emails
- Transferring Files
- Web Browsing

Where UDP is Used?

- Gaming
- Video Streaming
- Online Video Chats

Differences between TCP and UDP

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn't support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs , FTP , SMTP and Telnet .	UDP is used by DNS , DHCP , TFTP, SNMP , RIP , and VoIP .
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy	This protocol is used in situations where quick communication

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
	communication procedure is necessary, such as in email, on the web surfing, and in military services.	is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.

Example: Suppose there are two houses, H1 and H2, and a letter has to be sent from H1 to H2. But there is a river in between those two houses. Now how can we send the letter?
Solution 1: Make a bridge over the river and then it can be delivered.
Solution 2: Get it delivered by a pigeon.

- Consider the first solution as **TCP** . A connection has to be made (bridge) to get the data (letter) delivered. The data is reliable because it will directly reach another end without loss of data or error.
- The second solution is **UDP** . No connection is required for sending the data. The process is fast as compared to TCP, where we need to set up a connection(bridge). But the data is not reliable: we don't know whether the pigeon will go in the right direction, will drop the letter on the way, or some issue is encountered mid-travel.

Conclusion

To summarise, TCP and UDP are both important Transport Layer protocols with distinct properties and uses. TCP offers dependable, orderly, and error-free data transmission, making it ideal for operations that require precision, such as file transfers and web browsing. UDP, on the other hand, provides quicker, connectionless communication that is excellent for real-time applications such as gaming and video streaming, when speed is critical and minor data loss is acceptable. The exact requirements of the task at hand determine whether TCP or UDP should be used.

Stream Control Transmission Protocol (SCTP)

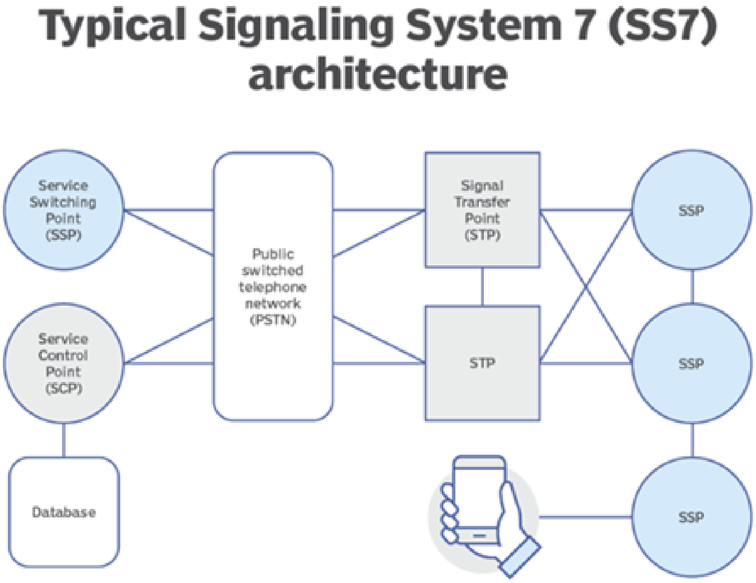
Last Updated : 11 Dec, 2023

Stream Control Transmission Protocol (SCTP) is a network protocol that is connection-oriented and used for transmitting multiple streams of data simultaneously between any two endpoints that have established a connection in a computer network. SCTP is a transport layer of Internet Protocol (IP).

> *SCTP support telephone connection over the internet.*

History of SCTP Protocol

SCTP is a standard protocol that was coined by The Transport Area Working Group (TSVWG) of the IETF (Internet Engineering Task Force). The reason for the development of the protocol is to develop a system that is similar to the telephone Signaling System 7 (SS7) switching network for carrying call control signals using networks.


Typical Signaling System 7 (SS7) architecture
Service Switching Point (SSP)
Public switched telephone network (PSTN)
Signal Transfer Point (STP)
SSP
Service Control Point (SCP)
STP
SSP
Database
SSP

TSVWG and IETF developed SCTP as a standard protocol

The SCTP is similar to TCP protocol but the advantage is that it also provides message oriented data transfer like User Datagram Protocol (UDP) which makes it useful for end to end communication over internet. Both TCP and UPD protocol are based on the concept that made SCTP possible. Unlike TCP SCTP make ensure that it complete the concurrent transmission over several streams of data in units called message between the end points which are connected to each other.

Understanding Stream Control Transmission Protocol

As we know SCTP is an transport layer protocol it exist at an equivalent level with UDP and TCP which provides the transport layer functions properties to many other Internet applications. As it is a reliable transport protocol which operates on top of connectionless packet networks like IP and supports transfer of data over the network in single or multiple Ip cases.

It transport the signaling message to and from Signaling System (SS7) for 3G mobiles networks with help of M3UA, M2Ua or SUA. It is a packet based transport protocol. It is both reliable and secure transport which minimize the end to end delay.

This protocol is optimized to :-

- It avoids problem related to he multithread infrastructure during the high traffic.
- It also improves the SCTP association searching rate by SCTP hash table optimization on the SPU(Services Processing Unit).
- It improves the FSM for retransmission of cases.

What is Multihoming in SCTP?

First we will understand multihoming so multihoming is the process of connecting a network or a host to multiple network simultaneously which is done due to increase reliability or performance.

Telecommunication systems are highly prone to time delays. Multihoming system enables with multiple interfaces to use one over the other without waiting. SCTP multihoming means that the endpoints which are connected can have different IP addresses associated to it. In simpler way multihoming refers to sending data to an alternate IP address if in case due to any issue the primary or original IP address is unreachable. Therefore the SCTP can connect or establish multiple connection paths between two endpoints.

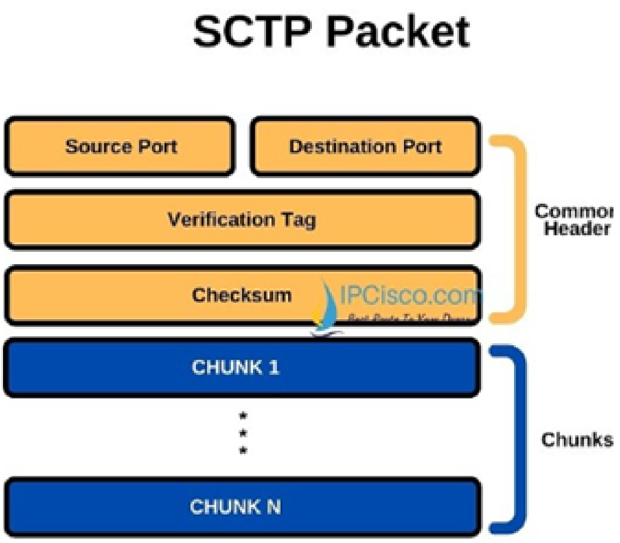
In this there is a original or primary interface or secondary interfaces. So during establishment of connections a acknowledgment process validates the IP address and manages the round trip time (RTT) for each individual address. The RTT calculation enables the communication to migrate to a secondary interface.

SCTP Packet

SCTP protocol packet consist of two main parts Header and Payload. The Header is common but Payload have variable chunks.

The Common SCTP header is 12 byte long and made of the 4 parts

- **Port Number (Source):** shows the sending port
- **Port Number (Destination):** shows the receiving port
- **Verification tag:** a 32 bit random value which differentiate the packets from the previous connection
- **Checksum:** a CRC32 algorithm for detection of error.



SCTP Packet

Security

This protocol provides certain security features related to transport such as resistance against blind DOS attack (Denial of Service), masquerades and monopolization of any type of service during operation. SIGTRAN (Signaling Transport) protocols does not define any type of new security mechanism as current available security protocols provide necessary steps for securing the transmission of SS7 message over IP networks

SCTP Services

- Aggregate Server Access Protocol (ASAP)
- Bearer-independent Call Control (BICC)
- Direct Data Placement Segment chunk (DDP-segment)
- Direct Data Placement Stream session control (DDP-stream)
- Diameter in a DTLS/SCTP DATA chunk (Diameter-DTLS)

Understanding Central Point Architecture Support for SCTP

As we know that the SCTP association is a connection between two SCTP endpoints. Each endpoint identifies it's association with a tag. During it's setup the SCTP endpoints exchange their tags for receiving packets. So during the exchange f packets between two SCTP endpoints the both source and destination address can change in the association life cycle.

Before the release of Junos OS 15.1X49-D40 all the sessions of the SCTP association are hashed to the same SPU with the help of the fixed per association SCTP port pair. In many of the cases multiple SCTP association use the same port pair, which results a bad load balancing with all the traffic handled by single SPU. When the version Junos OS release 15.1X49-D40 and Junos OS Release 17.3R1 to handle load balancing issue, the tag based hash distribution is used to ensure the even distribution of the traffic of SCTP from various associations among all the SPU's. It's flow session utilizes a connection tag to more finely distribute SCTP traffic across all the SPU's on the SRX1500, SRX4100, SRX4200, SRX5400, SRX600, and SRX800 devices that supports the SCTP ALG. The decoding of connection tag is from SCTP vtag.

Advantages of SCTP

As SCTP is a full duplex connection, it enables the data to be sent and receive simultaneously. The data is delivered in chunks and in a ordered way which are independent to each stream this help in isolating the data from other streams.

Like TCP and unlike UDP the SCTP provides the following advantage

- **Flow control:** It adjust the data transmission in a particular order and quantity.
- **Congestion control:** It checks for network prior transmission to prevent the congestion over the links.
- **Fault tolerance:** It uses the IP address from different internet services providers. So, if in case ISP fails another connection can be used for establishing the connection.
- It is a message oriented rather than byte oriented as of UDP.
- It provides a path selection functionality to select the primary data transmission and a monitoring function to test the connectivity of transmission path.

Limitation and Constraints of SCTP Protocol

- **IP address**
    - In this protocol a maximum of eight IP address and eight destination IP address are used in communication.
    - In this only static IP NAT is supported.
- **SCTP Payload Protocol Blocking**
    - If there is any change in the protocol blocking configuration it immediately impacts the traffic of existing associations.
    - The protocol which is supported is in decimal value ranging from 0 to 63, which includes 48 IANA protocol and 16 unassigned protocols.
- All the static NAT the interfaces packets (client or server side) should belong from the same zone.
- The sessions of SCTP are not deleted with associations they have a time out of 30 min (default).
- Only the Static NAT is supported for SCTP protocol.

Application Of SCTP Protocol

- **Telephone Communication:** It was developed foe the communication of telephony over the internet.
- **Multihoming Support:** It provides multihoming support, in which both endpoints of the connection can have multiple IP address which help helps in detection of failure in between the communication path.
- Transport for various Application: It is used in transport signaling messages to and from SS7(Signaling System 7) on the devices supporting 3G networks through M3UA , M2UA.
- **Roaming Security and RAN Security:** In mobile infrastructure it is used in roaming security and RAN (Radio Access Network) security.
- Reliable and Secure Transport: This protocol provides reliable and highly secure transport or communication which minimizes the end to end delay.

Conclusion

Stream Control Transmission Protocol (SCTP) is a connection oriented protocol which allows transmission of multiple data streams. SCTP was first coined by the Transport Area Working Group (TSVWG) of the Internet Engineering Task Force (IETF) to create a system similar to the telephone Signaling System 7 (SS7) switching network for carrying call control signals using IP networks. SCTP make sure that it completes transmission of several streams of data in units called messages between the connected endpoints. It supports the multihoming concept, , which increases the reliability and performance. it is a reliable and very secure and trustworthy transport protocol which minimizes end to end delay and provides security features like, resistance against blind DOS attacks, masquerades, and monopolization of services during operation. It is used in various applications such as Aggregate Server Access Protocol (ASAP), Bearer-independent Call Control (BICC), and others. It's advantages include full duplex connection, message oriented data transfer, flow control, congestion control, and fault tolerance. It has limitations of a maximum

Computer Network | Quality of Service and Multimedia

Last Updated : 28 Jun, 2024

Quality of Service (QoS) is an important concept, particularly when working with multimedia applications. Multimedia applications, such as video conferencing, streaming services, and VoIP (Voice over IP), require certain bandwidth, latency, jitter, and packet loss parameters. QoS methods help ensure that these requirements are satisfied, allowing for seamless and reliable communication.

What is Quality of Service?

Quality-of-service (QoS) refers to traffic control mechanisms that seek to differentiate performance based on application or network-operator requirements or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates. The basic phenomenon for QoS is in terms of packet delay and losses of various kinds.

QoS Specification

- Delay
- Delay Variation(Jitter)
- Throughput
- Error Rate

Types of Quality of Service

- **Stateless Solutions –** Routers maintain no fine-grained state about traffic, one positive factor of it is that it is scalable and robust. But it has weak services as there is no guarantee about the kind of delay or performance in a particular application which we have to encounter.

- **Stateful Solutions –** Routers maintain a per-flow state as flow is very important in providing the Quality-of-Service i.e. providing powerful services such as guaranteed services and high resource utilization, providing protection, and is much less scalable and robust.

QoS Parameters

- **Packet loss:** This occurs when network connections get congested, and routers and switches begin losing packets.
- **Jitter:** This is the result of network congestion, time drift, and routing changes. Too much jitter can reduce the quality of voice and video communication.
- **Latency:** This is how long it takes a packet to travel from its source to its destination. The latency should be as near to zero as possible.
- **Bandwidth:** This is a network communications link's ability to transmit the majority of data from one place to another in a specific amount of time.
- **Mean opinion score:** This is a metric for rating voice quality that uses a five-point scale, with five representing the highest quality.

How does QoS Work?

Quality of Service (QoS) ensures the performance of critical applications within limited network capacity.

- **Packet Marking**: QoS marks packets to identify their service types. For example, it distinguishes between voice, video, and data traffic.
- **Virtual Queues**: Routers create separate virtual queues for each application based on priority. Critical apps get reserved bandwidth.
- **Handling Allocation**: QoS assigns the order in which packets are processed, ensuring appropriate bandwidth for each application

Benefits of QoS

- Improved Performance for Critical Applications
- Enhanced User Experience
- Efficient Bandwidth Utilization
- Increased Network Reliability

- Compliance with Service Level Agreements (SLAs)
- Reduced Network Costs
- Improved Security
- Better Scalability

Why is QoS Important?

- Video and audio conferencing require a bounded delay and loss rate.
- Video and audio streaming requires a bounded packet loss rate, it may not be so sensitive to delay.
- Time-critical applications (real-time control) in which bounded delay is considered to be an important factor.
- Valuable applications should provide better services than less valuable applications.

Implementing QoS

- **Planning:** The organization should develop an awareness of each department's service needs and requirements, select an appropriate model, and build stakeholder support.
- **Design:** The organization should then keep track of all key software and hardware changes and modify the chosen QoS model to the characteristics of its network infrastructure.
- **Testing:** The organization should test QoS settings and policies in a secure, controlled testing environment where faults can be identified.
- **Deployment:** Policies should be implemented in phases. An organization can choose to deploy rules by network segment or by QoS function (what each policy performs).
- **Monitoring and analyzing:** Policies should be modified to increase performance based on performance data.

Models to Implement QoS

1. Integrated Services(IntServ)

- An architecture for providing QoS guarantees in IP networks for individual application sessions.
- Relies on resource reservation, and routers need to maintain state information of allocated resources and respond to new call setup requests.
- Network decides whether to admit or deny a new call setup request.

2. IntServ QoS Components

- Resource reservation: call setup signaling, traffic, QoS declaration, per-element admission control.
- QoS-sensitive scheduling e.g WFQ queue discipline.
- QoS-sensitive routing algorithm(QSPF)
- QoS-sensitive packet discard strategy.

3. RSVP-Internet Signaling

It creates and maintains distributed reservation state, initiated by the receiver and scales for multicast, which needs to be refreshed otherwise reservation times out as it is in soft state. Latest paths were discovered through “PATH” messages (forward direction) and used by RESV messages (reserve direction).

4. Call Admission

- Session must first declare it’s QoS requirement and characterize the traffic it will send through the network.
- **R-specification:** defines the QoS being requested, i.e. what kind of bound we want on the delay, what kind of packet loss is acceptable, etc.
- **T-specification:** defines the traffic characteristics like bustiness in the traffic.
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required.
- Routers will admit calls based on their R-spec, T-spec and based on the current resource allocated at the routers to other calls.

5. Diff-Serv

Differentiated Service is a stateful solution in which each flow doesn’t mean a different state. It provides reduced state services i.e. maintaining state only for larger granular flows rather than end-to-end flows tries to achieve the best of both worlds. Intended to address the following difficulties with IntServ and RSVP:

- **Flexible Service Models:** IntServ has only two classes, want to provide more qualitative service classes: want to provide ‘relative’ service distinction.
- **Simpler signaling:** Many applications and users may only want to specify a more qualitative notion of service.

QoS Tools

- Traffic Classification and Marking
- Traffic Shaping and Policing
- Queue Management and Scheduling
- Resource Reservation
- Congestion Management

What is Multimedia?

The word **multi** and **media** are combined to form the word **multimedia.** The word “**multi” signifies “many**.” **Multimedia is a type of medium that allows information to be easily transferred from one location to another**. Multimedia is the presentation of **text**, **pictures**, **audio**, and **video** with links and tools that allow the user to navigate, engage, create, and communicate using a computer.

Components of Multimedia

- **Text**: Characters are used to form words, phrases, and paragraphs in the text. The text can be in a variety of fonts and sizes to match the multimedia software’s professional presentation.
- **Graphics**: Non-text information, such as a sketch, chart, or photograph, is represented digitally. Graphics add to the appeal of the multimedia application. The use of visuals in multimedia

enhances the effectiveness and presentation of the concept. Windows Picture, Internet Explorer, and other similar programs are often used to see visuals.

- **Animations**: Animation is the process of making a still image appear to move. A presentation can also be made lighter and more appealing by using animation. In multimedia applications, the animation is quite popular. The following are some of the most regularly used animation viewing programs: Fax Viewer, Internet Explorer, etc.
- **Video**: Photographic images that appear to be in full motion and are played back at speeds of 15 to 30 frames per second. The term video refers to a moving image that is accompanied by sound, such as a television picture.
- **Audio**: Any sound, whether it's music, conversation, or something else. Sound is the most serious aspect of multimedia, delivering the joy of music, special effects, and other forms of entertainment. Decibels are a unit of measurement for volume and sound pressure level. Audio files are used as part of the application context as well as to enhance interaction. Audio files must occasionally be distributed using plug-in media players when they appear within online applications and webpages. MP3, WMA, Wave, MIDI, and RealAudio are examples of audio formats. The following programs are widely used to view videos: Real Player, Window Media Player, etc.

Conclusion

QoS is critical for ensuring that multimedia applications run smoothly and effectively across a network. QoS techniques contribute to the quality and reliability of real-time applications by regulating bandwidth, latency, jitter, and packet loss. To fulfill the distinct requirements of various forms of network traffic, QoS is implemented using a combination of categorization, prioritization, resource reservation, and traffic management techniques.

Token Bucket Algorithm

Last Updated : 14 Feb, 2024

The Token Bucket algorithm is a popular and simple method used in computer networking and telecommunications for traffic shaping and rate limiting. It is designed to control the amount of data that a system can send or receive in some sort of period, ensuring that the traffic conforms to a specified rate.

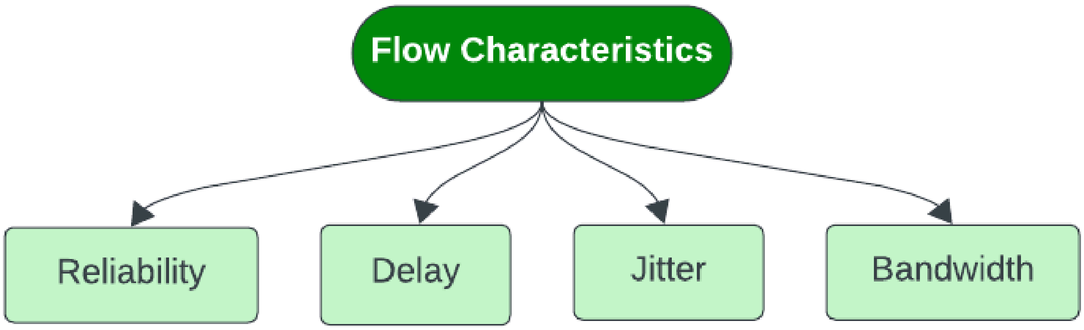
It refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates. It is something that data flow seeks to attain.

Need for Token Bucket Algorithm

- Video and audio conferencing require a bounded delay and loss rate.
- Video and audio streaming requires a bounded packet loss rate, it may not be so sensitive to delay.
- a -critical applications (real-time control) in which bounded delay is considered to be an important factor.
- Valuable applications should provide better services than less valuable applications.

Flow Characteristics of Token Bucket Algorithm

Four types of characteristics are attributed to a flow: reliability, delay, jitter, and bandwidth.



Types of Characteristics for Quality of Service

Reliability

It implies packet reached or not, information lost or not. Lack of reliability means losing a packet or acknowledgement, which entails re-transmission. Reliability requirements may differ from program to program. For example, it is more important that electronic mail, file transfer and internet access have reliable transmissions than telephony or audio conferencing.

Delay

It denotes source-to-destination delay. Different applications can tolerate delay in different degrees. Telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

Jitter

Jitter is the variation in delay for packets belonging in same flow. High jitter means the difference between delays is large; low jitter means the variation is small. For example, if packets 0,1,2,3s arrive at 6,7,8,9s it represents same delay. Jitter would signify that packets departed at 0,1,2,3s reach destination at 4,6,10,15s. Audio and video applications don't allow jitter.

Bandwidth

Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

Techniques to Improve QoS

There are several ways to improve QoS like Scheduling and Traffic shaping ,We will see each and every part of this in brief.

Scheduling

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Three scheduling techniques are:

1. FIFO Queuing
2. Priority Queuing
3. Weighted Fair Queuing

To learn more about the scheduling techniques visit this article on packet queuing and dropping.

Traffic Shaping

It is a mechanism to control the amount and the rate of the traffic sent to the network. The techniques used to shape traffic are: leaky bucket and token bucket.

Difference Between Token Bucket Algorithm and Leaky Bucket Algorithm

The differences between leaky and token bucket algorithm are:

Token Bucket Algorithm	Leaky Bucket Algorithm
It depends on tokens.	It does not depend on tokens.
If bucket is full, token is discarded but not the packet.	If bucket is full, then packets are discarded.
Packets can only transmit when there are enough tokens.	Packets are transmitted continuously.

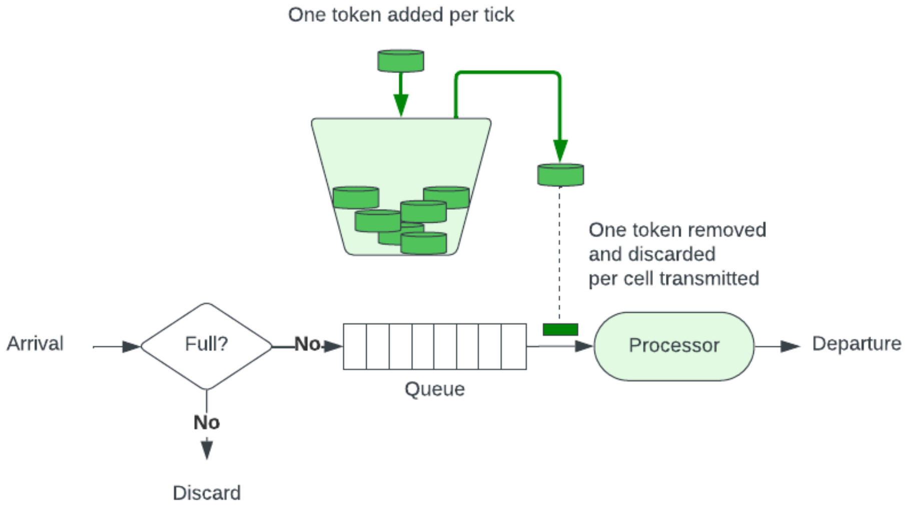
Token Bucket Algorithm	Leaky Bucket Algorithm
Allows large bursts to be sent at faster rate. Bucket has maximum capacity.	Sends the packet at a constant rate.
The bucket holds tokens generated at regular intervals of time.	When the host has to send a packet , packet is thrown in bucket.
If there is a ready packet , a token is removed from Bucket and packet is send.	Bursty traffic is converted into uniform traffic by leaky bucket.
If there is no token in the bucket, then the packet cannot be sent.	In practice bucket is a finite queue outputs at finite rate.

Leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full. But this technique is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. If the host has bursty data, the leaky bucket allows only an average rate. The time when the host is idle is not take into account. On the other hand, token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. And that is how it overcomes the shortcoming of leaky bucket algorithm.

Working of Token Bucket Algorithm

It allows bursty traffic at a regulated maximum rate. It allows idle hosts to accumulate credit for the future in the form of tokens. The system removes one token for every cell of data sent. For each tick of the clock the system send n tokens to the bucket. If n is 100 and host is idle for 100 ticks, bucket collects 10000 tokens. Host can now consume all these tokens with 10 cells per tick.

Token bucket can be easily implemented with a counter. The token is initiated to zero. Each time a token is added, counter is incremented to 1. Each time a unit of data is sent, counter is decremented by 1. When the counter is zero, host cannot send data.



Process depicting how token bucket algorithm works

Steps Involved in Token Bucket Algorithm

Step 1: Creation of Bucket: An imaginative bucket is assigned a fixed capacity, known as "rate limit". It can hold up to a certain number of tokens.

Step 2: Refill the Bucket: The bucket is dynamic; it gets periodically filled with tokens. Tokens are added to the bucket at a fixed rate.

Step 3: Incoming Requests: Upon receiving a request, we verify the presence of tokens in the bucket.

Step 4: Consume Tokens: If there are tokens in the bucket, we pick one token from it. This means the request is allowed to proceed. The time of token consumption is also recorded.

Step 5: Empty Bucket: If the bucket is depleted, meaning there are no tokens remaining, the request is denied. This precautionary measure prevents server or system overload, ensuring operation stays within predefined limits.

Advantage of Token Bucket over Leaky Bucket

- If a bucket is full in tokens, then tokens are discarded and not the packets. While in leaky bucket algorithm, **packets are discarded**.
- Token bucket can send large bursts at a **faster rate** while leaky bucket always sends packets at constant rate.
- Token bucket ensures **predictable traffic shaping** as it allows for setting token arrival rate and maximum token count. In leaky bucket, such control may not be present.
- Premium Quality of Service(QoS) is provided by prioritizing different traffic types through distinct token arrival rates. Such **flexibility in prioritization** is not provided by leaky bucket.
- Token bucket is suitable for high-speed data transfer or streaming video content as it allows **transmission of large bursts of data**. As leaky bucket operates at a constant rate, it can lead to less efficient bandwidth utilization.
- Token Bucket provides more **granular control** as administrators can adjust token arrival rate and maximum token count based on network requirements. Leaky Bucket has limited granularity in controlling traffic compared to Token Bucket.

Disadvantages of Token Bucket Algorithm

- Token Bucket has the tendency to generate tokens at a fixed rate, even when the network traffic is not present. This is leads of accumulation of unused tokens during times when there is no traffic, hence leading to wastage.
- Due to token accumulation, delays can introduced in the packet delivery. If the token bucket happens to be empty, packets will have to wait for new tokens, leading to increased latency and potential packet loss.
- Token Bucket happens to be less flexible than leaky bucket when it comes to network traffic shaping. The fixed token generation rate cannot be easily altered to meet changing network requirements, unlike the adaptable nature of leaky bucket.
- The implementation involved in token bucket can be more complex, especially due to the fact that different token generation rates are