1. **Explain the function of transport layer.**

- **End-to-end communication** is the ability of the transport layer to provide the application a way to send and receive a stream of data. The network layer segments the data stream into packets that are sent over the network and reconstructs the data on the other end. If the data packets arrive out of order, it can reorder them by segment numbering and present the data in the correct order.
- **Reliability** is the ability to correct errors that can happen during data transmission over the network. If data were to be accidentally changed in transit, error correcting and checksums would catch it. If a packet were to be lost, it would be caught and retransmitted. If a single packet were to be duplicated, it could be detected and dropped. It can also send an acknowledgement of received packets for guaranteed delivery.
- **Flow control** is the ability for the transport layer to avoid sending more data than can be reliably transmitted. It can buffer sending and receiving data until there is enough network capacity for it to go through. If the receiver buffer becomes full, it can reduce the sending rate. It also implements congestion control. If a network were to become flooded with too many retransmit messages, it would be overwhelmed and not able to recover. Congestion control prevents this by using dynamic retransmission timers and slow start.
- **Addressing** is the ability to communicate with the correct application on the computer. Addressing typically uses network ports to assign each sending and receiving application a specific port number on the machine. By combining the IP address used in the network layer and the port on the transport layer, each application can have a unique address.
- **Multiplexing** is the ability for any number of applications to use any number of network connections. For example, a typical desktop computer may only have one Ethernet network connection but have several connections to the internet running at the same time, such as a web browser, video streaming and a mail client. Conversely, a large server may only have one application, such as a SQL server, but have two physical Ethernet connections to provide as much bandwidth as possible. The transport layer ensures that each application gets a fair amount of shared network connections.

2. Explain process to process communication.

Process-to-process communication at the transport layer involves the exchange of data between specific processes running on different devices within a network. This layer provides services that ensure reliable and efficient transmission of data between these processes. The two main transport layer protocols, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), offer different communication models to meet varying application needs. TCP establishes a connection-oriented communication channel between sender and receiver processes, guaranteeing delivery, sequencing, and error detection through mechanisms like acknowledgment and retransmission. In contrast, UDP provides a connectionless communication model, where data packets, called datagrams, are sent without establishing a connection, offering low overhead and minimal delay but sacrificing reliability. Both protocols use port numbers to identify the processes on the sending and receiving devices, enabling multiplexing and demultiplexing of data streams. Process-to-process communication at the transport layer plays a vital role in facilitating various networked applications, including web browsing, file transfer, email, and multimedia streaming, by ensuring seamless data exchange between different processes across the network.

3. Differentiate between reliable and unreliable communication

	Reliable Communication:	**Unreliable Communication:**
Guaranteed Delivery	In reliable communication, the sender ensures that data reaches the receiver successfully. This is typically achieved through acknowledgment mechanisms, where the receiver sends back an acknowledgment to confirm receipt of the data.	Unreliable communication does not ensure that all data reaches the receiver, nor does it guarantee the order of delivery. Data packets may be lost, duplicated, or delivered out of order.
Sequencing	Data is delivered in the same order it was sent, ensuring that the receiver can reconstruct the original message accurately.	Unreliable communication protocols typically have lower overhead compared to reliable protocols since they do not need to implement acknowledgment, retransmission, and sequencing mechanisms.
Error Detection and Correction	Reliable communication protocols often include mechanisms for detecting and correcting errors that may occur during transmission, such as checksums or cyclic redundancy checks (CRC).	Due to the lack of acknowledgment and retransmission mechanisms, data is transmitted quickly without waiting for confirmation from the receiver.
Example	Transmission Control Protocol (TCP) is a widely used reliable communication protocol in computer networks	Datagram Protocol (UDP) is a common example of an unreliable communication protocol

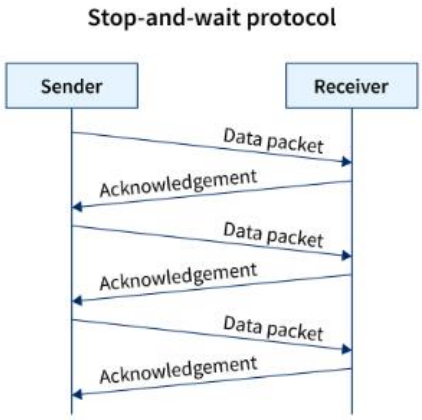
4. **Differentiate between connection oriented and connectionless communication with example**

Connection-oriented Service	**Connection-less Service**
Connection-oriented service is related to the telephone system.	Connection-less service is related to the postal system.
Connection-oriented service is preferred by long and steady communication.	Connection-less Service is preferred by bursty communication.
Connection-oriented Service is necessary.	Connection-less Service is not compulsory.
Connection-oriented Service is feasible.	Connection-less Service is not feasible.
In connection-oriented Service, Congestion is not possible.	In connection-less Service, Congestion is possible.
Connection-oriented Service gives the guarantee of reliability.	Connection-less Service does not give a guarantee of reliability.
In connection-oriented Service, Packets follow the same route.	In connection-less Service, Packets do not follow the same route.
Connection-oriented services require a bandwidth of a high range.	Connection-less Service requires a bandwidth of low range.
Ex: TCP (Transmission Control Protocol)	Ex: UDP (User Datagram Protocol)
Connection-oriented requires authentication.	Connection-less Service does not require authentication.

5. **Explain reliability protocols used in transport layer**.

a. Simple Protocol:

The Simple Protocol is a basic communication protocol that involves sending a single packet of data from a sender to a receiver. It does not include mechanisms for error detection, acknowledgment, or retransmission of lost packets. The sender simply sends the packet, and the receiver attempts to receive it. If the packet is received successfully, the receiver processes it. If not, the packet is lost, and no further action is taken. This protocol is straightforward but lacks reliability and robustness, making it suitable only for very simple communication scenarios.



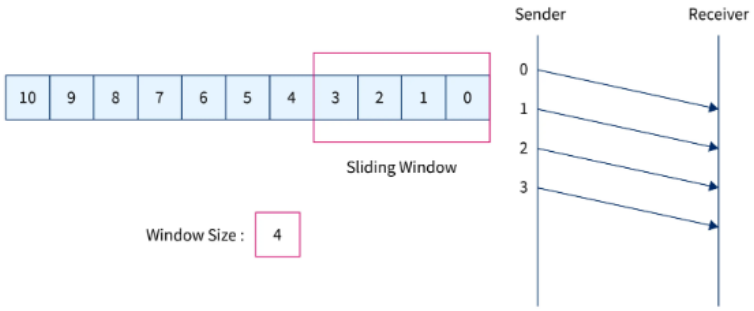
b. Stop-and-Wait Protocol:

The Stop-and-Wait Protocol is a simple automatic repeat request (ARQ) protocol used for reliable communication over an unreliable channel. In this protocol, the sender sends a single packet and waits for an acknowledgment (ACK) from the receiver before sending the next packet. If the sender does not receive an ACK within a specified timeout period, it assumes that the packet was lost and retransmits it. The receiver sends an ACK for each correctly received packet, and it discards duplicate packets. While simple and easy to implement, the Stop-and-Wait Protocol can suffer from low efficiency, especially in high-latency networks, as the sender must wait for acknowledgment before sending the next packet.

c. Go-Back-N Protocol:

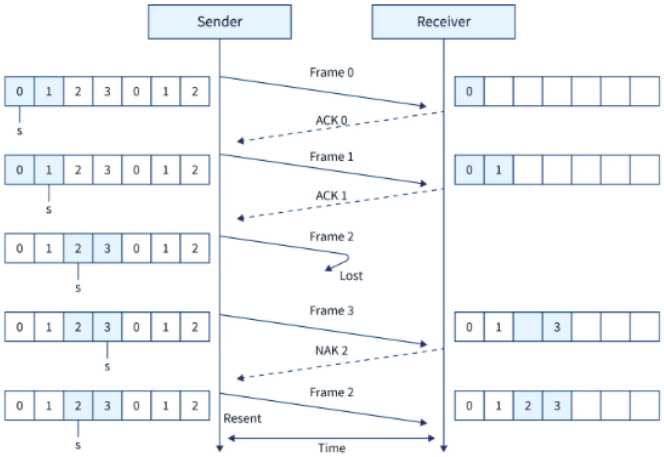
The Go-Back-N Protocol is a sliding window ARQ protocol that allows the sender to transmit multiple packets before receiving acknowledgments from the receiver. It maintains a window of packets that have been sent but not yet acknowledged. The sender can transmit packets within this window without waiting for individual acknowledgments. If the sender does not receive an acknowledgment for a particular packet within a timeout period, it retransmits all unacknowledged

packets starting from the earliest unacknowledged packet (hence the name "Go-Back-N"). The receiver acknowledges correctly received packets and discards out-of-order or duplicate packets. Go-Back-N is more efficient than Stop-and-Wait, but it requires more complex buffering and error handling mechanisms.



d. Selective Repeat Protocol:

The Selective Repeat Protocol is another sliding window ARQ protocol that allows the sender to transmit multiple packets before receiving acknowledgments. Unlike Go-Back-N, Selective Repeat only retransmits individual packets that are lost or damaged, rather than retransmitting all unacknowledged packets. Both the sender and receiver maintain a window of packets, and the receiver sends acknowledgments for each correctly received packet, regardless of order. If the sender does not receive an acknowledgment for a particular packet within a timeout period, it retransmits only that specific packet, rather than retransmitting all unacknowledged packets. Selective Repeat is more efficient than Go-Back-N in terms of bandwidth utilization but requires more complex buffering and acknowledgment management.



6. Define UDP protocol. Write the services and application of UDP.

UDP (User Datagram Protocol) is a connectionless transport layer protocol used for sending data packets over a network. Unlike TCP (Transmission Control Protocol), UDP does not establish a connection before transmitting data and does not provide mechanisms for ensuring reliable delivery, acknowledgment, or sequencing of packets. Instead, UDP operates in a best-effort manner, where data packets, known as datagrams, are sent independently, without prior negotiation or handshaking between sender and receiver.

Services of UDP:

- **Connectionless Communication**: UDP does not require a connection setup before transmitting data, allowing for quick and efficient transmission of packets.
- **Low Overhead**: UDP incurs lower overhead compared to TCP since it avoids the computational overhead associated with connection setup, acknowledgment, and flow control mechanisms.
- **Low Latency**: UDP is often preferred in applications where low latency and real-time communication are crucial, as it minimizes the delay introduced by connection establishment and acknowledgment mechanisms.
- **Broadcast and Multicast Support**: UDP supports broadcasting and multicasting, allowing a single packet to be sent to multiple recipients simultaneously.
- **Simple Implementation**: UDP's simplicity makes it easier to implement and use compared to TCP, making it suitable for applications where reliability is not a primary concern.

Applications of UDP:

Real-Time Communication: UDP is commonly used in applications requiring real-time communication, such as online gaming, multimedia streaming (e.g., video conferencing, VoIP), and live video/audio broadcasting.

DNS (Domain Name System) Resolution: UDP is used for DNS queries and responses, where low latency and quick responses are important.

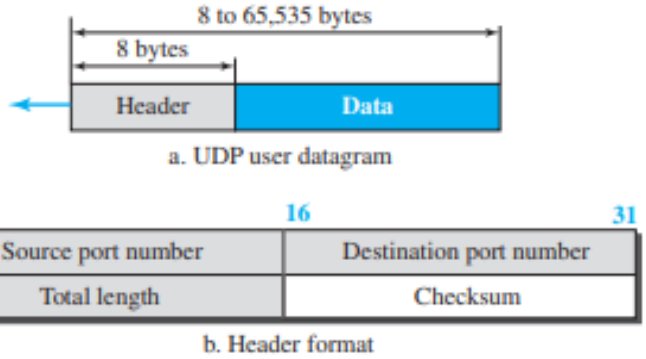
DHCP (Dynamic Host Configuration Protocol): DHCP uses UDP for client-server communication during the process of assigning IP addresses and network configuration parameters to devices on a network.

Network Monitoring and Management: UDP is utilized in network monitoring tools, such as SNMP (Simple Network Management Protocol), for sending management information between network devices and management stations.

IoT (Internet of Things) Applications: UDP is suitable for lightweight IoT applications, where minimizing overhead and conserving resources are priorities.

Broadcasting and Multicasting: UDP is used for broadcasting data packets to multiple recipients on a network simultaneously, such as in multimedia streaming or network discovery protocols.

7. Explain the datagram format of UDP protocol.



- **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.

8. **Define TCP. Write its services and application**.

TCP (Transmission Control Protocol) is a connection-oriented transport layer protocol in the Internet Protocol Suite (TCP/IP). It provides reliable, ordered, and error-checked delivery of data packets between communicating applications running on different hosts within a network. TCP ensures that data is delivered

accurately and in the correct order, even in the presence of network congestion, packet loss, or errors.

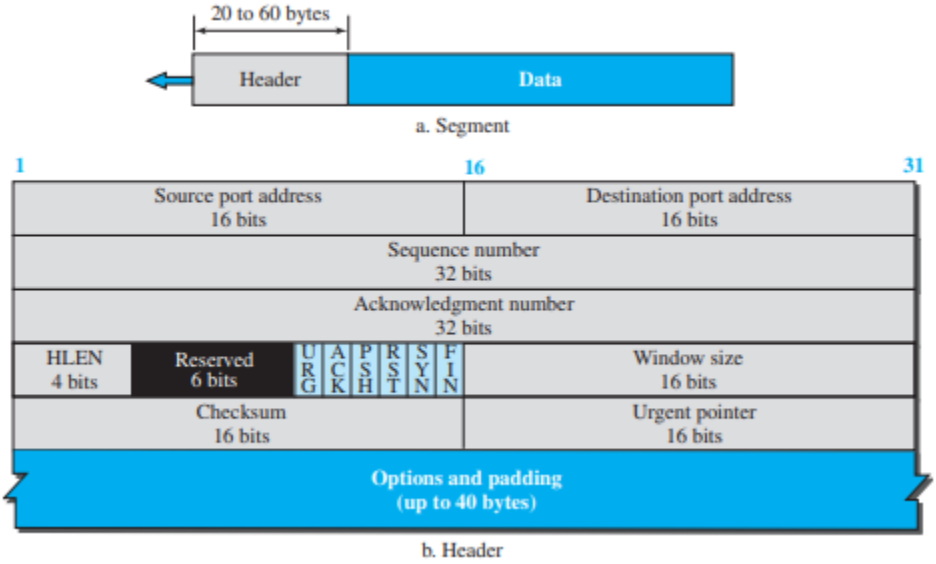
Services of TCP:

- **Reliable Delivery**: TCP ensures reliable delivery of data by using acknowledgment, retransmission, and sequencing mechanisms. The receiver sends acknowledgments (ACKs) for successfully received packets, prompting the sender to resend any unacknowledged packets.
- **Ordered Delivery**: TCP delivers data packets to the receiver in the same order they were sent by the sender, preserving the sequence of transmitted messages.
- **Flow Control**: TCP implements flow control mechanisms to prevent the sender from overwhelming the receiver with data. This includes the use of sliding window protocols to regulate the rate of data transmission based on available buffer space at the receiver.
- **Congestion Control**: TCP monitors network conditions and adjusts transmission rates to alleviate congestion and prevent network overload, ensuring fair resource allocation and stable performance.

Applications of TCP:

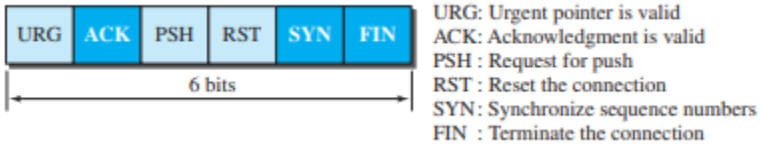
- **We**b Browsing: TCP is the foundation of HTTP (Hypertext Transfer Protocol), the protocol used for accessing websites and transferring web pages between web servers and clients (web browsers).
- **Email**: TCP is used for sending and receiving emails via protocols such as SMTP (Simple Mail Transfer Protocol) for sending emails and POP3 (Post Office Protocol version 3) or IMAP (Internet Message Access Protocol) for receiving emails.
- **File Transfer**: TCP is commonly used for file transfer protocols such as FTP (File Transfer Protocol) and SFTP (SSH File Transfer Protocol) for transferring files between hosts.
- **Remote Login**: TCP is used for remote login protocols such as SSH (Secure Shell) and Telnet for securely accessing and managing remote systems.
- **VoIP (Voice over IP):** TCP is used for VoIP applications, such as Skype, Zoom, and other voice and video conferencing services, for transmitting real-time audio and video data over the internet.
- **Database Access**: TCP is used for accessing databases over the network, such as MySQL, PostgreSQL, and Oracle, allowing clients to connect to and query remote databases.

9. Explain datagram format of TCP.



Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.
Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence is the first byte in the segment. During connection establishment (discussed later) each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.
Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from the other party, it returns x + 1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.
Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 × 4 = 20) and 15 (15 × 4 = 60).

Control. This field defines 6 different control bits or flags. One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP. A brief description of each bit is shown in the figure. We will discuss them further when we study the detailed operation of TCP later in the chapter.



Window size. This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Checksum. This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the use of the checksum in the UDP datagram is optional, whereas the use of the checksum for TCP is mandatory

10. **Differentiate between TCP and UDP with service and application**.

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.	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.
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 communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.