1. **List out Transport Layer Protocols.**

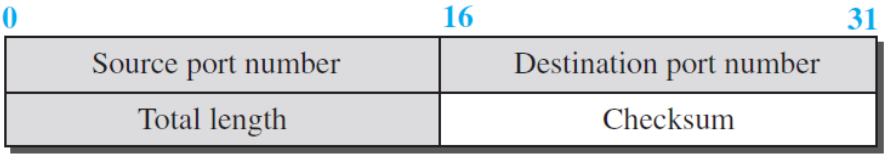
- There are three transport layer protocols.
    - UDP – User Datagram Protocol.
    - TCP – Transport Layer Protocol.
    - SCTP – Stream Control Transport Protocol.

2. **Differentiate connection-oriented and connectionless services.**

- Connection-oriented service.
    - Connection-oriented services establish a connection between the sender and receiver before any data is transmitted.
    - This connection is like a telephone call: the sender dials the receiver's number, and a circuit is established between the two parties.
    - Once the connection is established, data can be transmitted in a reliable way. The sender can be sure that all of the data it sends will be received by the receiver in the correct order.
- Connectionless service
    - Connectionless services, on the other hand, do not establish a connection before data is transmitted.
    - Instead, data is sent in packets, and each packet is addressed to the receiver.
    - The packets may or may not arrive in the correct order, and there is no guarantee that all of the packets will be received.
    - Connectionless services are more efficient than connection-oriented services, but they are also less reliable.

3. **Explain UDP user datagram header format.**

- The UDP user datagram header format is given below.



UDP user datagram header format

- UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).
- The first two fields define the source and destination port numbers.
- The third field defines the total length of the user datagram, header plus data.
  Total Length = Header Length + Data Length
- The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- The last field can carry the optional checksum.

4. **The following is the content of a UDP header in hexadecimal format.**

   CB84000D001C001C

   What is the source port number?

   What is the destination port number?

   What is the total length of the user datagram?

   What is the length of the data?

Is the packet directed from a client to a server or vice versa?

What is the client process?

- The source port number is the first four hexadecimal digits $(CB84)_{16}$, which means that the source port number is 52100.
- The destination port number is the second four hexadecimal digits $(000D)_{16}$, which means that the destination port number is 13.
- The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.
- The length of the data is the length of the whole packet minus the length of the header, or 28 − 8 = 20 bytes.
- Since the destination port number is 13 (well-known port), the packet is from the client to the server.
- The client process is the Daytime.

5. Explain UDP services.

- Process-to-Process Communication
  - UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.
- Connectionless service
  - It means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- Error Control
  - There is no error control mechanism in UDP except for the checksum.
- Encapsulation and Decapsulation
  - To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages.
- Queuing
  - In UDP, queues are associated with ports. At the client site, when a process starts, it requests a port number from the operating system.
  - Some implementations create both an incoming and an outgoing queue associated with each process.
  - Other implementations create only an incoming queue associated with each process.
- Multiplexing and Demultiplexing
  - UDP provides multiplexing at the sender side and demultiplexing at the receiver side.

6. List out protocols that use the UDP as the transport layer protocol.

- DNS – Domain Name System
- SNMP – Simple Network Management Protocol
- RIP – Routing Information Protocol.
- DHCP – Dynamic Host Configuration Protocol
- TFTP – Trivial File Transfer Protocol

7. Explain byte number, sequence number and acknowledgement number in TCP?

- Byte Number
  - It is used for flow and error control.
  - TCP numbers all data bytes (octets) that are transmitted in a connection. The numbering does not necessarily start from 0. Instead, TCP chooses an arbitrary number between 0 and $2^{32} - 1$ for the number of the first byte.

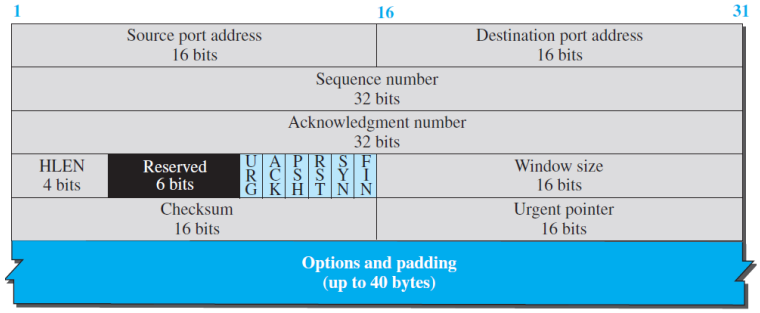
  - For example, if the number happens to be 1057 and the total data to be sent is 6000 bytes, the bytes are numbered from 1057 to 7056.
- Sequence number
  - TCP assigns a sequence number to each segment that is being sent. The sequence number, in each direction, is defined as follows:
    - The sequence number of the first segment is the ISN (initial sequence number), which is a random number.
    - The sequence number of any other segment is the sequence number of the previous segment plus the number of bytes (real or imaginary) carried by the previous segment.
  - The value in the sequence number field of a segment defines the number assigned to the first data byte contained in that segment.
- Acknowledgement number
  - Client and Server use an acknowledgment number to confirm the bytes it has received.
  - The acknowledgment number defines the number of the next byte that the client or server expects to receive.
  - The acknowledgment number is cumulative, which means that the party (Client or Server) takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the acknowledgment number.
  - The term cumulative here means that if a party (Client or Server) uses 5643 as an acknowledgment number, it has received all bytes from the beginning up to 5642.

8. Explain TCP services.

- Process-to-Process Communication
  - TCP provides process-to-process communication using port numbers.
- Stream Delivery Service
  - TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
  - TCP uses stream buffers on the sender as well as receiver side.
- Full-Duplex Communication
  - TCP offers full-duplex service, where data can flow in both directions at the same time.
- Multiplexing and Demultiplexing
  - TCP performs multiplexing at the sender and demultiplexing at the receiver.
- Connection-oriented Service
  - TCP established the connection before data transfer.
  - TCP terminates the connection as the data transfer completed.
- Reliable Service
  - TCP is a reliable transport protocol.
  - It uses an acknowledgment mechanism to check the safe and sound arrival of data.

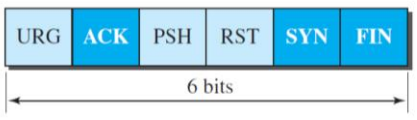
9. Explain TCP segment header format.

- The TCP segment header format is given below.



TCP Segment Header Format

- 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.
- Acknowledgment number.
  - This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.
- 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, as shown below.



URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH : Request for push
RST : Reset the connection
SYN: Synchronize sequence numbers
FIN : Terminate the connection

  - These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.
- Window size.
  - This field defines the window size of the sending TCP in bytes.
- Checksum
  - This 16-bit field contains the checksum.
- Urgent pointer.
  - This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.
- Options.
  - There can be up to 40 bytes of optional information in the TCP header.

10. The following is part of a TCP header dump (contents) in hexadecimal format.

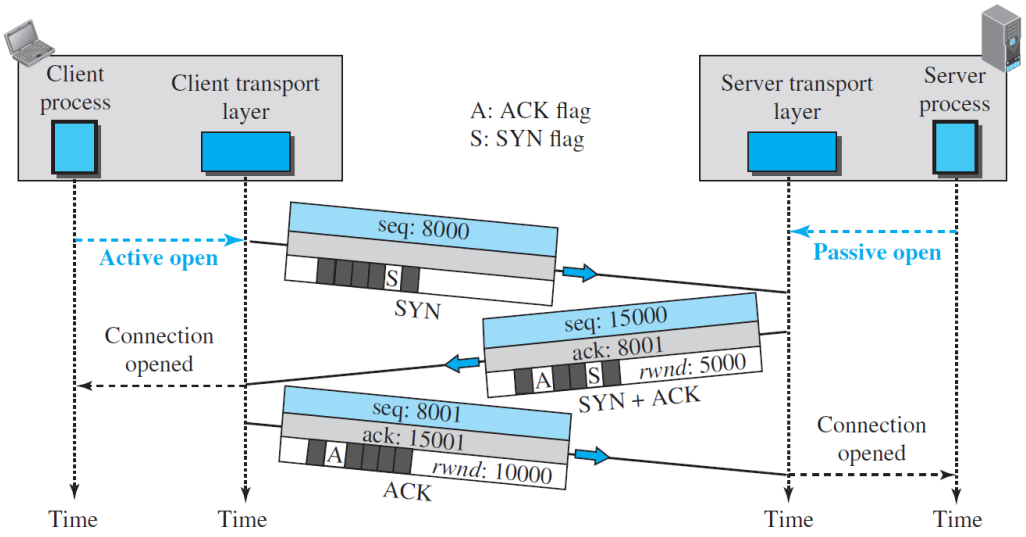
E293 0017 00000001 0000000 5002 07FF..
a. What is the source port number?

b. What is the destination port number?
c. What is the sequence number?
d. What is the acknowledgment number?
e. What is the length of the header?
f. What is the type of the segment?
g. What is the window size?

a. Source Port number = $(E293)_{16}$ = 58003
b. Destination port number = $(0017)_{16}$ = 23
c. Sequence number = $(00000001)_{16}$ = 1
d. Acknowledgement number = $(00000000)_{16}$ = 0
e. Length of the header = 5 * 4 = 20 Bytes
f. Type of segment = SYN Segment
g. Window size = 2047 Bytes.

11. Explain TCP connections establishment with diagram.

- The client program which wants to transfer data to the server program and vice versa, first establish the connection using TCP connection establishment.
- The connection establishment in TCP is called three-way handshaking.
- In the first handshake, the client transfers the TCP segment with the SYN flag set to the server.
- In the second handshake, the server generates the TCP segment with the SYN and ACK flag, replies to the client.
- In the third handshake, Client generates the TCP segment with the ACK flag, replies to the server.
- Consider the following diagram to understand the TCP connection establishment.



TCP Connection Establishment

- Active open is the connection between client process and client TCP program.
- Passive open is the connection between server process and server TCP program.
- The explanation of the TCP connection establishment diagram is as follows.
  1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN).
  2. The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the

other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client. The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.

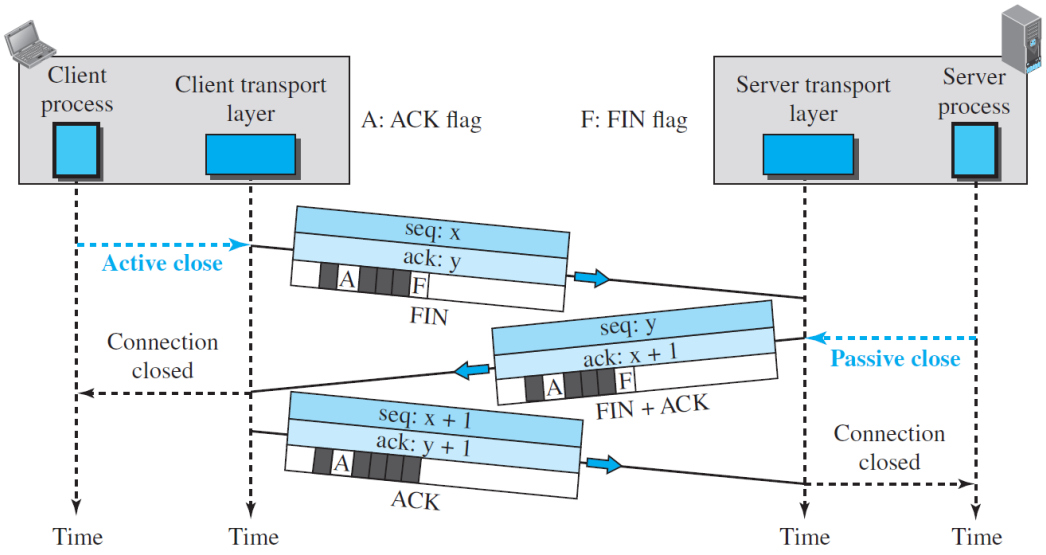
3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.

12. Explain TCP connection termination with diagram.

- Either the client or the server can initiate the termination of a TCP connection, though it is typically done by the client.
- Modern implementations provide two methods for connection termination:
    - Three-way handshaking
    - Four-way handshaking with a half-close option.

Three-way handshaking

- Consider the following diagram to understand the Connection termination using Three-way handshaking.



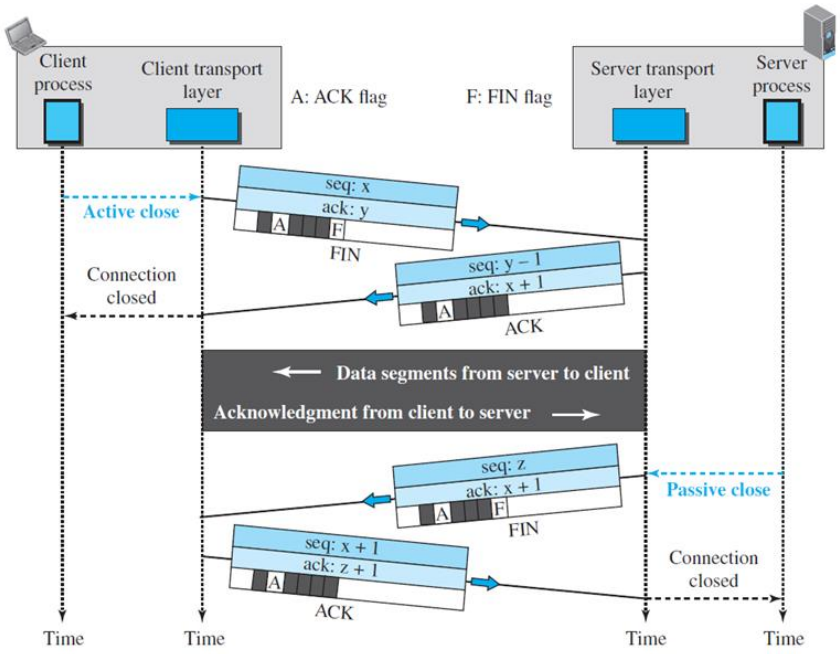
TCP Connection termination with three handshakes.

- The explanation of the above diagram is as follows.
    1. The client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.
    2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.
    3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.

Four-way handshaking with a half-close option.

- In TCP, one end can stop sending data while still receiving data. This is called a half-close.
- Either the server or the client can issue a half-close request.

- Consider the following diagram to understand the connection termination using a half-close option.



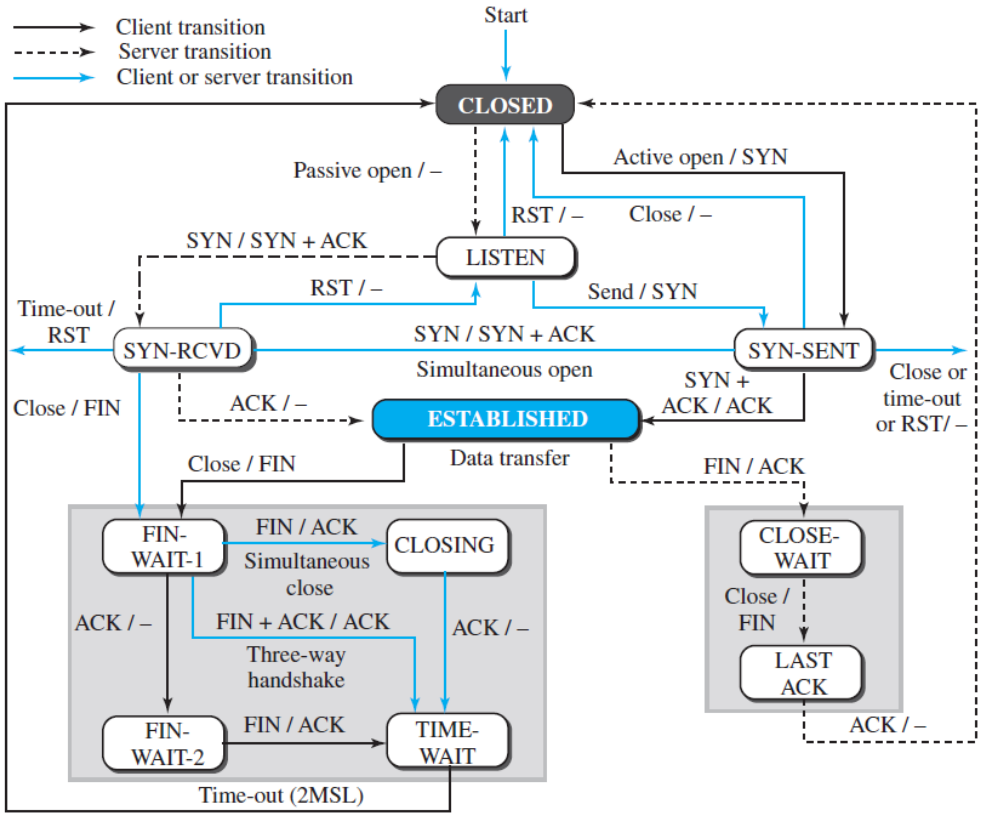
TCP Connection termination with half-close option

- The data transfer from the client to the server stops.
    - The client half-closes the connection by sending a FIN segment.
    - The server accepts the half-close by sending the ACK segment.
- The server, however, can still send data.
    - When the server has sent all of the processed data, it sends a FIN segment, which is acknowledged by an ACK from the client.
- After half-closing the connection, data can travel from the server to the client and acknowledgments can travel from the client to the server. The client cannot send any more data to the server.

13. List out protocols that use the TCP as the transport layer protocol.

- HTTP – Hyper Text Transport Protocol
- FTP – File Transport Protocol
- SMTP – Simple Mail Transfer Protocol
- SSH – Secure Shell
- Telnet – Teletype Network

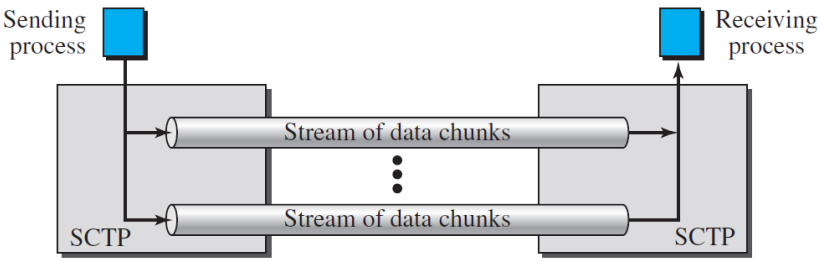
14. Draw state transition diagram of TCP.



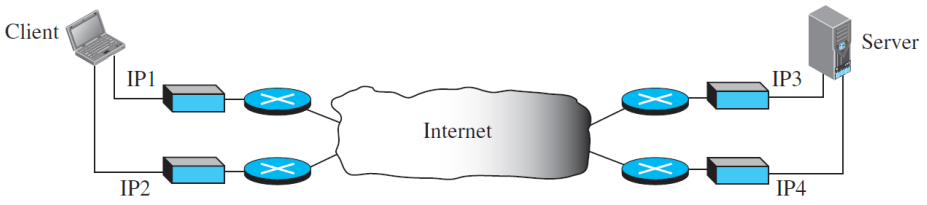
TCP State transition diagram

15. Explain SCTP services.

- Process-to-Process Communication
  - SCTP provides process-to-process communication between sending and receiving process.
- Multiple Streams
  - SCTP allows multistream service in each connection, which is called association in SCTP terminology.
  - If one of the streams is blocked, the other streams can still deliver their data.



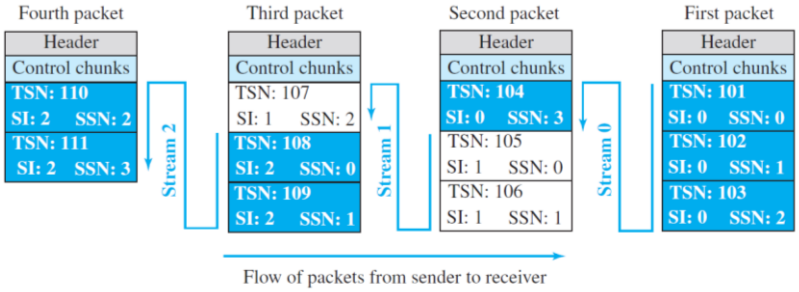
- Multihoming
  - An SCTP association supports multihoming service.
  - The sending and receiving host can define multiple IP addresses in each end for an association. In this fault-tolerant approach, when one path fails, another interface can be used for data delivery without interruption.



- Full-Duplex Communication
  - SCTP offers full-duplex service, where data can flow in both directions at the same time.
- Connection-Oriented Service
  - SCTP provides connection-oriented services between sender and receiver.
- Reliable Service
  - An SCTP is a reliable transport protocol.
  - It uses an acknowledgment mechanism to check the safe and sound arrival of data.

16. Explain Transmission Sequence Number (TSN), Stream Identifier (SI), Stream Sequence Number (SSN) with example.

- Transmission Sequence Numbers (TSN)
  - The unit of data in SCTP is a data chunk.
  - SCTP uses a transmission sequence number (TSN) to number the data chunks.
  - TSNs are 32 bits long and randomly initialized between 0 and $2^{32} - 1$.
  - Each data chunk must carry the corresponding TSN in its header.
- Stream Identifier (SI)
  - Each stream in SCTP needs to be identified using a stream identifier (SI).
  - Each data chunk must carry the SI in its header.
  - The SI is a 16-bit number starting from 0.
- Stream Sequence Number (SSN)
  - When a data chunk arrives at the destination SCTP, it is delivered to the appropriate stream and in the proper order and for that SCTP uses a stream sequence number (SSN).
- Consider the following example to understand the TSN, SI and SSN.
- The process A needs to send 11 messages to process B in three streams. The first four messages are in the first stream, the second three messages are in the second stream, and the last four messages are in the third stream. Although a message, if long, can be carried by several data chunks, we assume that each message fits into one data chunk. Therefore, we have 11 data chunks in three streams.
- The application process delivers 11 messages to SCTP, where each message is earmarked for the appropriate stream. Here we assume that it delivers all messages belonging to the first stream first, all messages belonging to the second stream next, and finally, all messages belonging to the last stream. We also assume that the network allows only three data chunks per packet, which means that we need four packets, as shown in Figure below.



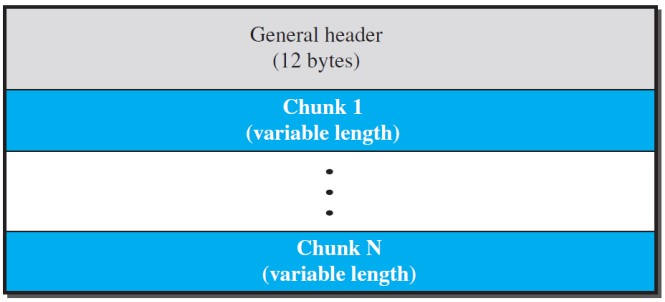
- Data chunks in stream 0 are carried in the first and part of the second packet; those in stream 1 are carried in the second and the third packet; those in stream 2 are carried in the third and fourth packet.
- Each data chunk needs three identifiers: TSN, SI, and SSN.
- TSN is a cumulative number and is used, for flow control and error control. In our example, the first message is divided into four chunk and the first chunk TSN is 101 and fourth chunk TSN is 104.
- SI defines the stream to which the chunk belongs. In our example three different SI numbers are used 0, 1 and 2.
- SSN defines the chunk's order in a particular stream. In our example, SSN starts from 0 for each stream.

17. Explain SCTP packet format.

- An SCTP packet has a mandatory general header and a set of blocks called chunks.
- There are two types of chunks: control chunks and data chunks.
- A control chunk controls and maintains the association.
- a data chunk carries user data.
- In a packet, the control chunks come before the data chunks.
- The SCTP packet format is shown below.

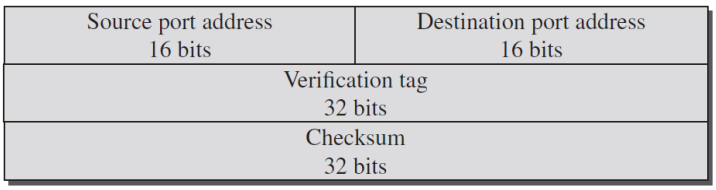
General Header

- The general header (packet header) defines the end points of each association to which the packet belongs, guarantees that the packet belongs to a particular association, and preserves the integrity of the contents of the packet including the header itself.
- The format of the SCTP packet is shown below.



SCTP packet format

- There are four fields in the general header.

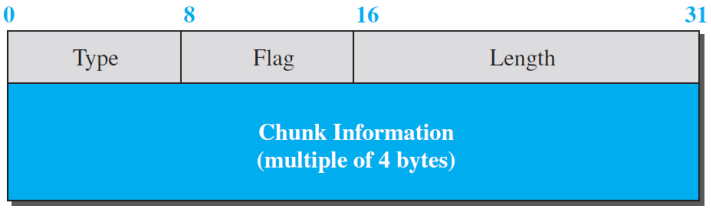


General Header of the SCTP Packet

- 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 SCTP packet.
- 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 SCTP packet.

- The verification tag
  - It is a 32-bit field that matches a packet to an association.
  - It serves as an identifier for the association; it is repeated in every packet during the association.
- Checksum
  - It is used for error control.
- Control information or user data are carried in chunks. Chunks have a common layout, as shown in Figure below.



Control Chunk Format

- Type
  - It can define up to 256 types of chunks.
- Flag
  - It defines special flags that a particular chunk may need. Each bit has a different meaning depending on the type of chunk.
- Length
  - The length field defines the total size of the chunk, in bytes, including the type, flag, and length fields.
  - If a chunk carries no information, the value of the length field is 4
- Information
  - It depends on the type of chunk.

18. The following is a dump of an SCTP general header in hexadecimal format.

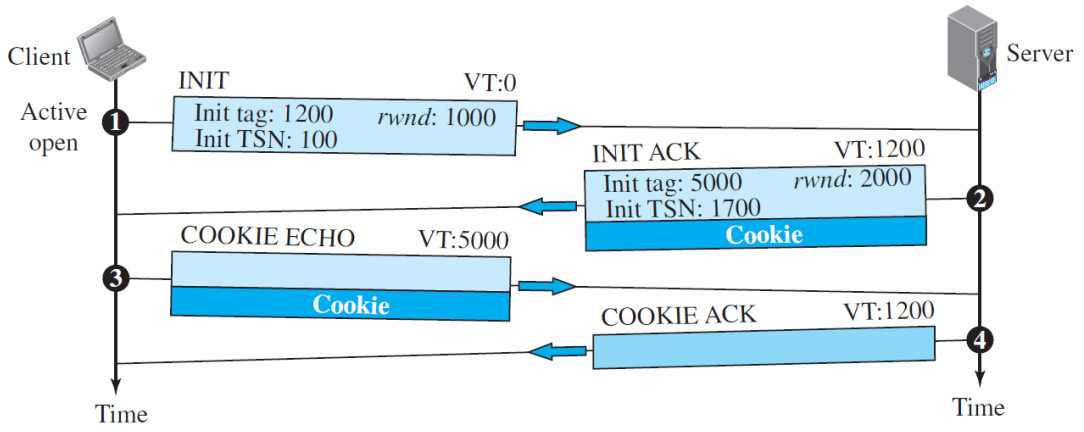
04320017 00000001 00000000
a. What is the source port number?
b. What is the destination port number?
c. What is the value of the verification tag?
d. What is the value of the checksum?

a. Source port number = $(0432)_{16}$ = 1074
b. Destination port number = $(0017)_{16}$ = 23
c. Value of the verification tag = $(00000001)_{16}$ = 1
d. Value of the checksum = $(000000000)_{16}$ = 1

19. Explain SCTP Association establishment with diagram.

- A connection in SCTP is called an association.
- Association establishment in SCTP requires a four-way handshake.
- In this procedure, a process, normally a client, wants to establish an association with another process, normally a server, using SCTP as the transport-layer protocol.

- The SCTP server needs to be prepared to receive any association (passive open). Association establishment, however, is initiated by the client (active open).
- SCTP association establishment is shown in the following figure.

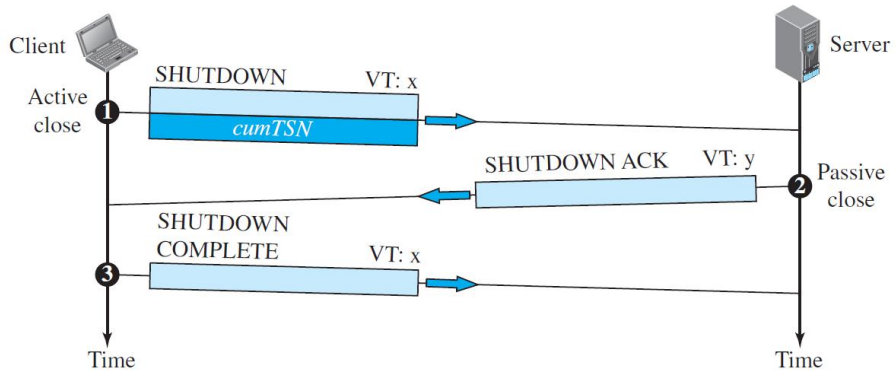


SCTP association establishment.

- These four handshakes are explained below.
    1. The client sends the first packet, which contains an INIT chunk. The verification tag (VT) of this packet (defined in the general header) is 0 because no verification tag has yet been defined for this direction (client to server). The INIT tag includes an initiation tag to be used for packets from the other direction (server to client). The chunk also defines the initial TSN for this direction and advertises a value for rwnd.
    2. The server sends the second packet, which contains an INIT ACK chunk. The verification tag is the value of the initial tag field in the INIT chunk. This chunk initiates the tag to be used in the other direction, defines the initial TSN, for data flow from server to client, and sets the server's rwnd. The INIT ACK also sends a cookie that defines the state of the server at this moment.
    3. The client sends the third packet, which includes a COOKIE ECHO chunk.
    4. The server sends the fourth packet, which includes the COOKIE ACK chunk that acknowledges the receipt of the COOKIE ECHO chunk.

20. Explain SCTP Association termination with diagram.

- In SCTP, either of the two parties involved in exchanging data (client or server) can close the connection. SCTP does not allow a "half-closed" association.
- Association termination uses three packets, as shown in Figure below.



SCTP Association Termination

- These three handshakes are explained below.
    1. In the first handshake, client initiates the connection closing request by sending the packet which contains the SHUTDOWN chunk to the server.
    2. In the second handshake, server sends SHUTDOWN ACK chunk in response to the connection closing request from the client and start connection closing from the server side also.
    3. In the third handshake, client sends SHUTDOWN COMPLETE chunk to the server and close the connection.
- The above figure shows the case in which termination is initiated by the client, it can also be initiated by the server.

21. Differentiate UDP and TCP.

UDP	TCP
It is a connectionless protocol.	It is a connection-oriented protocol.
It is unreliable protocol.	It is reliable protocol.
UDP datagram header size is 8 bytes.	TCP segment header size is 20-60 bytes.
Checksum is optional.	Checksum is mandatory.
There is no need to establish connection before data transfer. Hence it is faster than TCP.	There is a need to establish connection before data transfer. Hence it is slower than UDP.
It generates low overhead compared to TCP.	It generates more overhead compared to UDP.
It does not provide congestion control.	It provides congestion control.
An application layer protocol DHCP uses the services of UDP.	An application layer protocol HTTP uses the services of TCP.

22. Differentiate TCP and SCTP.

TCP	SCTP
Connection establishment done using three-way handshake.	Association establishment done using four-way handshake.
Connection termination can be done using three handshake or half-close with four handshakes.	Connection termination done using only three handshakes. It does not support half-close.
It does not support multihoming.	It supports multihoming.
It does not support multi streaming	It supports multi streaming.
It is byte oriented.	It is message oriented.
It does not support unordered delivery of data.	It supports both ordered and unordered delivery of data.
Application layer protocols such as HTTP uses the service of TCP.	Application layer protocols such as VoIP uses the service of SCTP.

23. Differentiate UDP and SCTP.

UDP	SCTP
It is a connection less protocol.	It is a connection-oriented protocol.
It is unreliable protocol.	It is reliable protocol.
UDP datagram header size is 8 bytes.	SCTP packet header size is
It does not provide built-in error handling.	It provides built-in error handling.

There is no need to establish connection before data transfer. Hence it is faster than SCTP.	There is a need to establish association before the data transfer. Hence it is slower than UDP.
It is datagram oriented.	It is message oriented.
It does not provide congestion control.	It provides congestion control.
An application layer protocol DHCP uses the services of UDP.	An application layer protocol HTTP uses the services of TCP.