Transport Layer

Outline

- Goal
  - Understand the Connection oriented and connectionless protocol
  - Frame Formats (TCP)
- Transport Layer Responsibilities
- UDP
- TCP
- Summary

Transport Layer

Other Responsibilities

- **Service-point addressing:** Network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process (using Port Address)
- **Segmentation and reassembly:** A message is divided into transmittable segments,with each segment containing a sequence number.
- **Connection control**
- **Flow control**
- **Error control**

Transport Layer

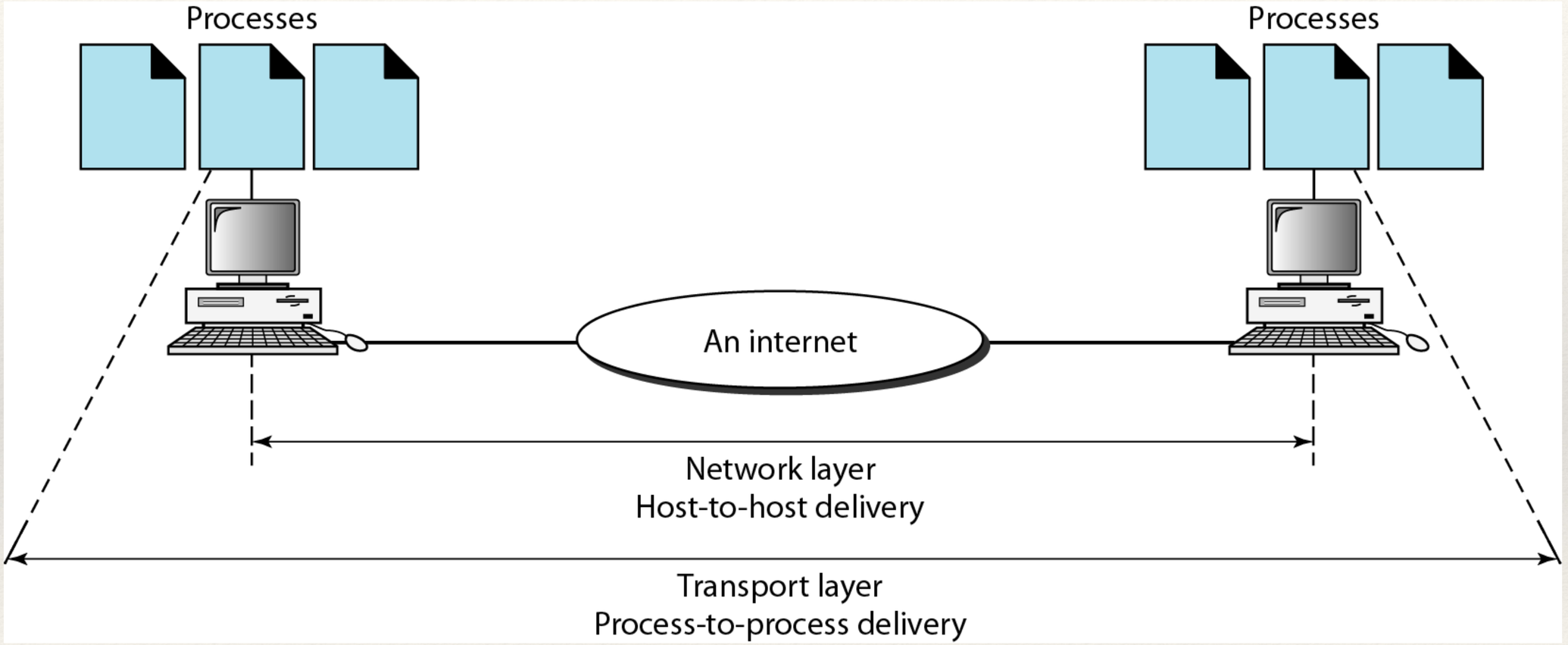

Fig 17: Reliable process-to-process delivery of a message

UDP

- User Datagram Protocol (UDP) is called a connectionless, unreliable transport
- protocol.
- It does not add anything to the services of IP except to provide process-to process communication instead of host-to-host communication.
- If a process wants to send a small message and does not care much about reliability, it can
- use UDP.
- Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP

UDP Header

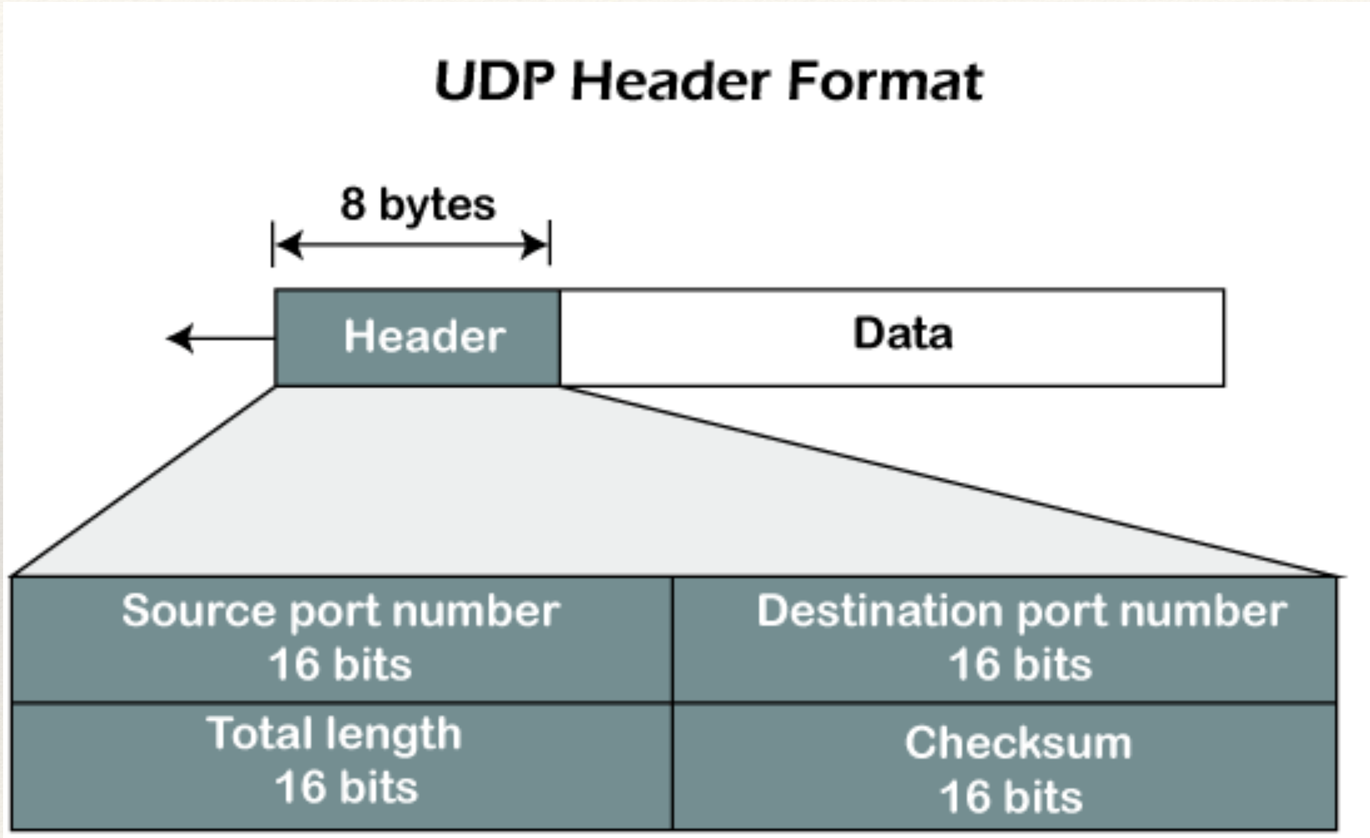

Fig: UDP Header

UDP: Pros and Cons

- Limitations
  - It provides an unreliable connection delivery service. It does not provide any services of IP except that it provides process-to-process communication.
  - The UDP message can be lost, delayed, duplicated, or can be out of order.
  - It does not provide a reliable transport delivery service. It does not provide any acknowledgment or flow control mechanism.
  - However, it does provide error control to some extent.
- Advantages
  - It produces a minimal number of overheads.

TCP

- Features
  - **Reliable** :TCP is a reliable protocol as it follows the flow and error control mechanism. It also supports the acknowledgment mechanism, which checks the state and sound arrival of the data.
  - **Order of the data:** This protocol ensures that the data reaches the intended receiver in the same order in which it is sent.
  - **Connection-oriented:** It is a connection-oriented service that means the data exchange occurs only after the connection establishment.
  - When the data transfer is completed, then the connection will get terminated.
  - **Full duplex:** It is a full-duplex means that the data can transfer in both directions at the same time.
  - **Stream-oriented:** TCP is a stream-oriented protocol as it allows the sender to send the data in the form of a stream of bytes and also allows the receiver to accept the data in the form of a stream of bytes.

TCP: Three-way Handshake

- In TCP, the connection is established by using three-way handshaking.
  - The client sends the segment with its sequence number.
  - The server, in return, sends its segment with its own sequence number as well as the acknowledgement sequence, which is one more than the client sequence number.
  - When the client receives the acknowledgment of its segment, then it sends the acknowledgment to the server.

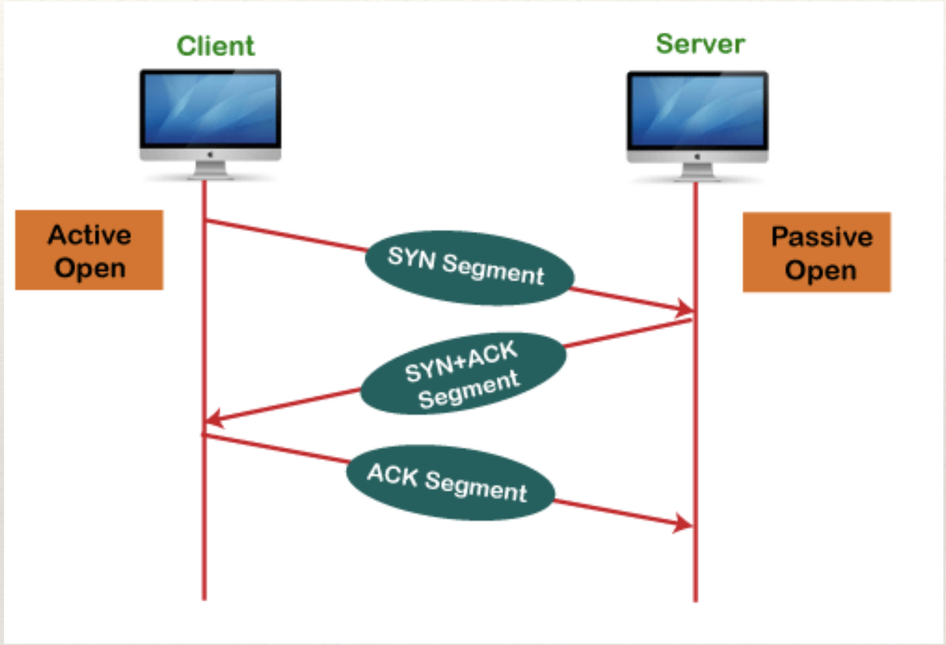

Fig: Three-way Handshake

TCP Header Format

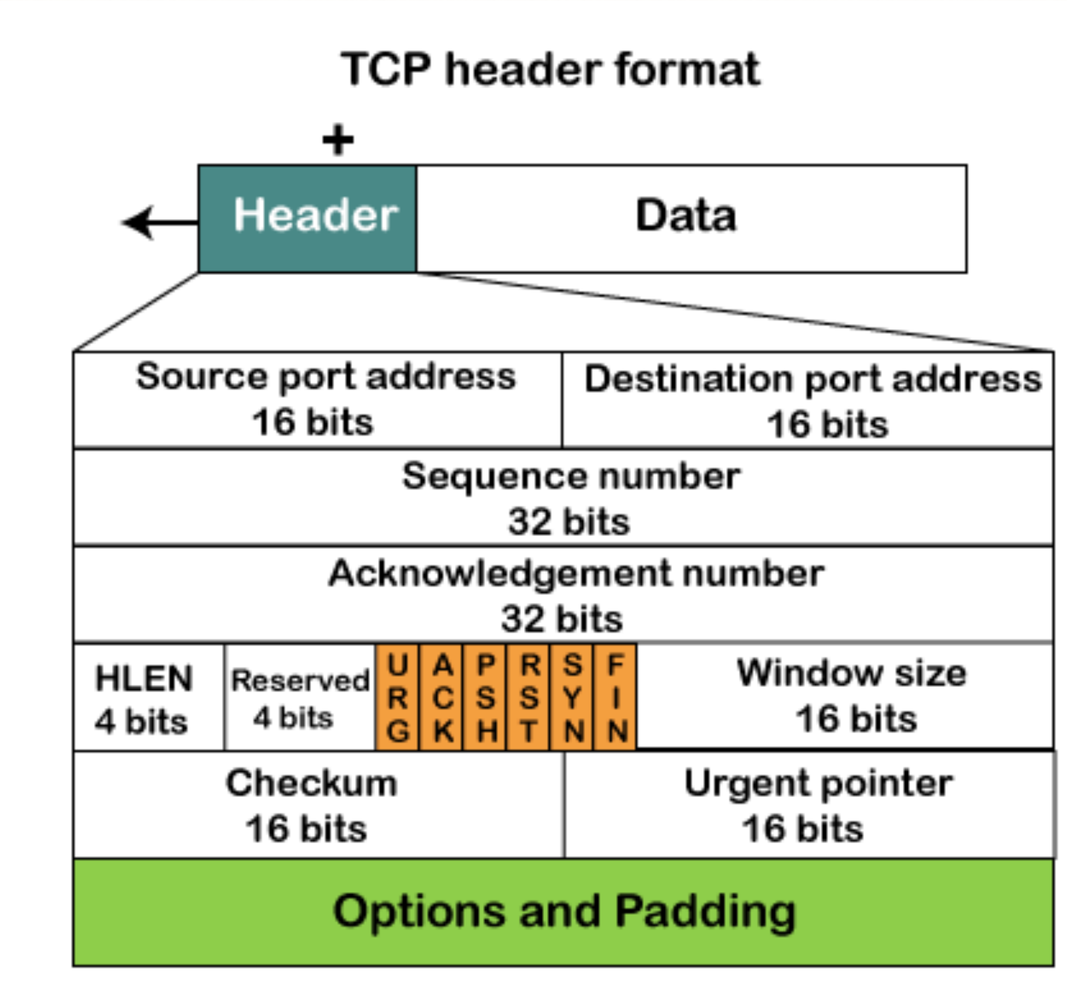

Fig: TCP Header Format

TCP Header Format

- **Source port:** It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.
- **Destination port:** It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.
- **Sequence number:** This field contains the sequence number of data bytes in a particular session.
- **Acknowledgment number:** When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.
- **HLEN:** It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.
- **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.
- Flags: There are six control bits or flags:
  - **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
  - **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.

TCP Header Format

- **PSH**: If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
- **RST**: If it is set, then it requests to restart a connection.
- **SYN**: It is used to establish a connection between the hosts.
- **FIN**: It is used to release a connection, and no further data exchange will happen.

- **Window size:** It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.
- **Checksum:** It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.
- **Urgent pointer:** It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.
- **Options:** It provides additional options. The optional field is represented in 32-bits. If this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

Summary

- Transport Layer Responsibilities
- TCP and UDP

Note: **Chapter 23 (23.2 and 23.3)**