Transport Layer

Introduction

- Transport layer is the layer 4 of the OSI reference model.
- The transport layer uses the services provided by the network layer, such as best path selection and logical addressing, to provide end-to-end communication between source and destination.
- Its primary duties are to transport and regulate the flow of information from source to destination reliably and accurately.
- End-to-end control, provided by sliding windows and reliability in sequencing numbers and acknowledgments, is also.

Transport layer duties

Segmenting upper layer

- Transport layer functionality is accomplished segment by segment.
  - This means that different applications can send data segments on a first-come, first-served basis.
  - Such segments can be intended for the same destination or for many different destinations.

Establishing a connection

- Figure 1 below, show a typical connection between a client and server:-
  - Client connection request
  - Acknowledgement and server request.
  - Acknowledgment inform the server that both sides have agreed, and a connection has been established.
  - After the connection has been established, data transfer begins.

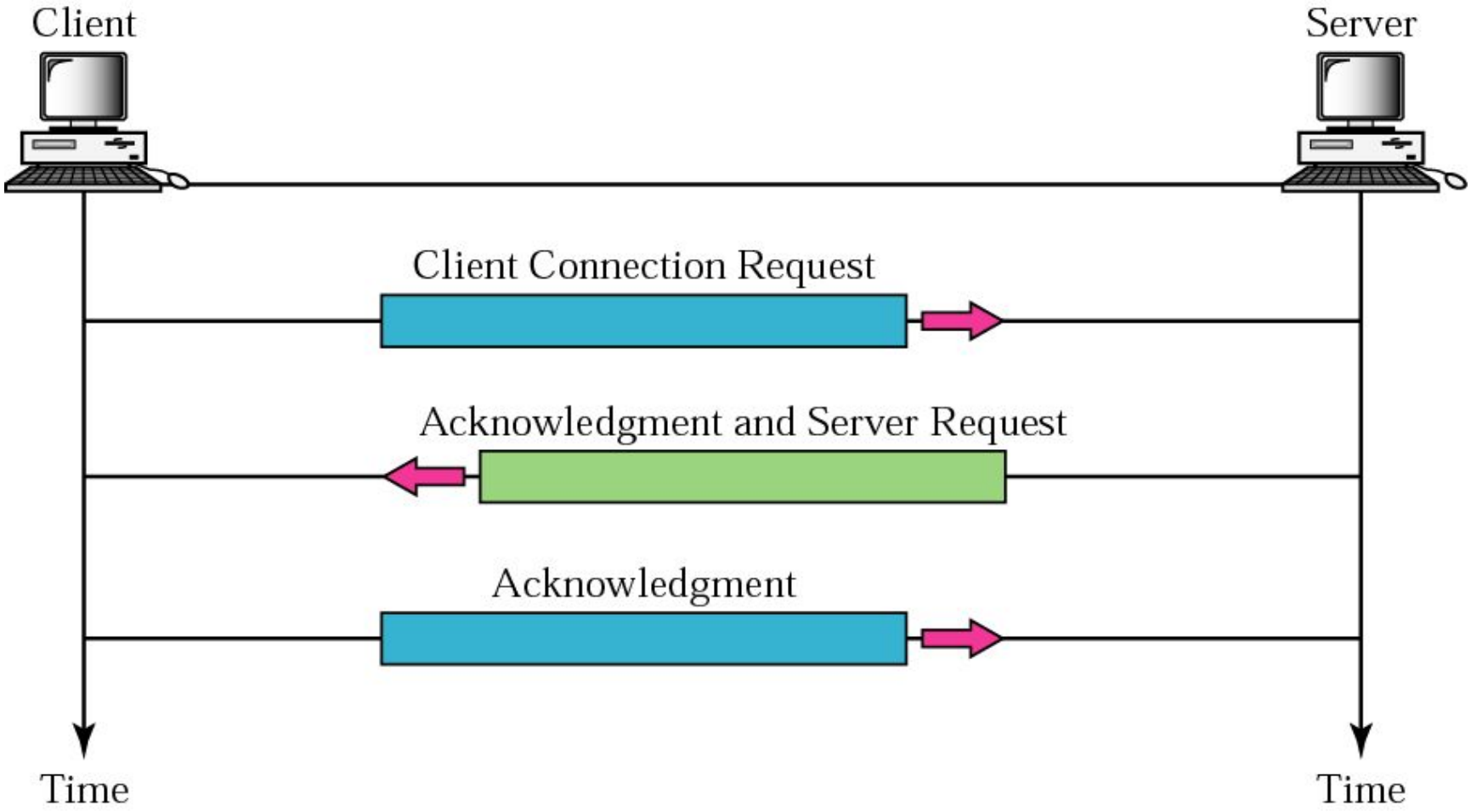

Figure 1:Establishing a connection between a client and server

Windowing

- Windowing is a method to control the amount of information transferred end-to-end.
  - For example, In the most basic form of reliable connection-oriented data transfer.
  - Data packets must be delivered to the recipient in the same order in which they were transmitted.
  - The protocol fails if any data packets are lost, damaged, duplicated or received in a different order.
  - The basic solution is to have a recipient acknowledge the receipt of each and every segment or data packets.
  - But if the sender waits for an acknowledge after sending each segment, throughput is low.

Windowing……

- Window size (see Figure 2) determines the amount of data that can be transmitted…
  - before stopping transmission and waiting for an acknowledgment from the destination.
  - The larger the window size number (in bytes, for TCP)
  - The greater the amount of data that the host can transmit before stopping.

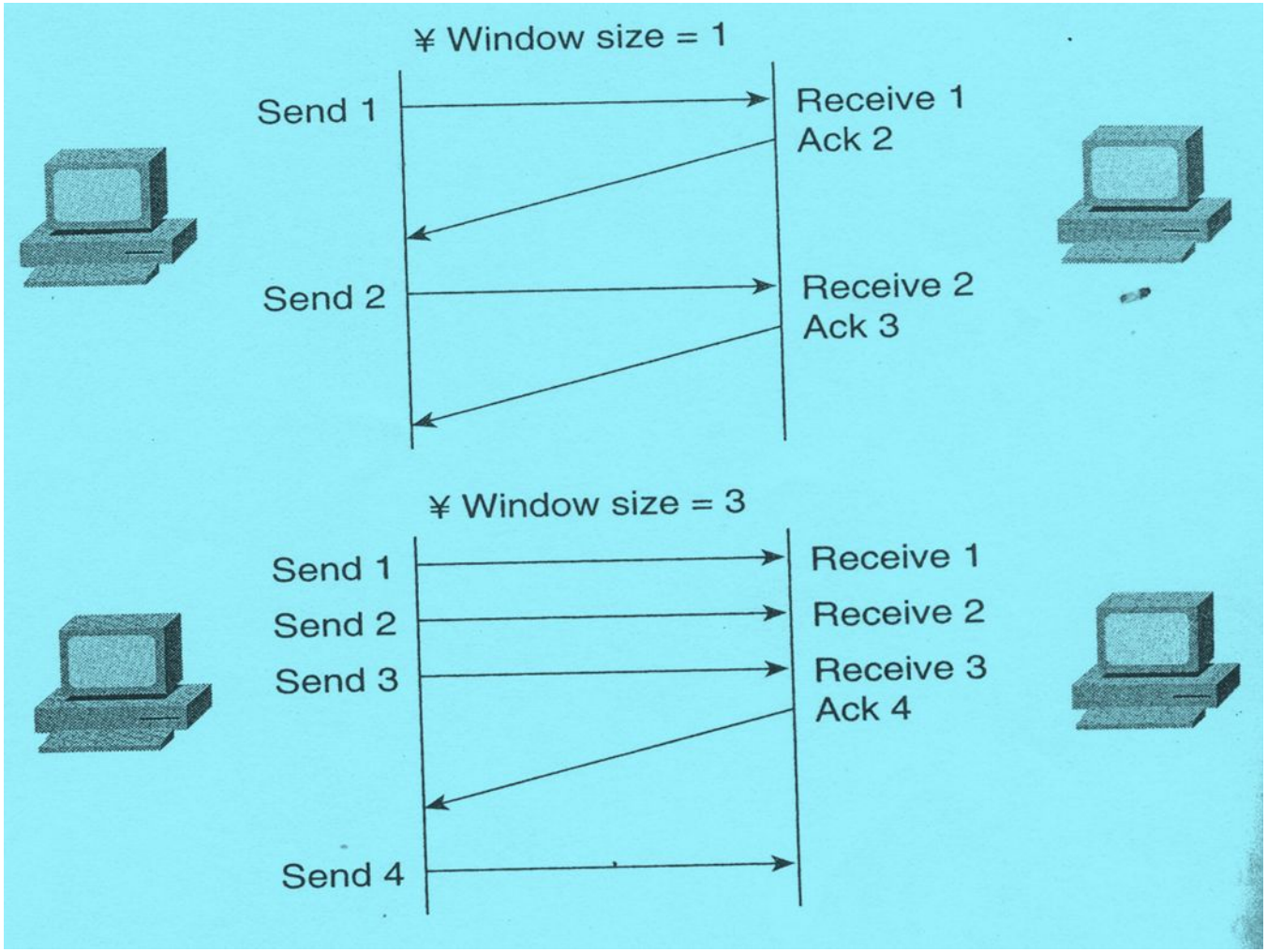

Figure 2 : Window size

Acknowledgment

- Positive acknowledgment with retransmission is one technique that guarantees reliable delivery of data streams.
- Positive acknowledgment requires a recipient to communicate with the source and..
  - sending back an acknowledgment message when it receives data.
  - For example in Figure 3, the sender keeps a record of each data packet it sends and waits for an acknowledgment before sending the next data packet.
  - The sender also starts a timer when it sends a segment, and…
  - Retransmits a segment if the timer expires before an acknowledgment arrives (see Figure 4).

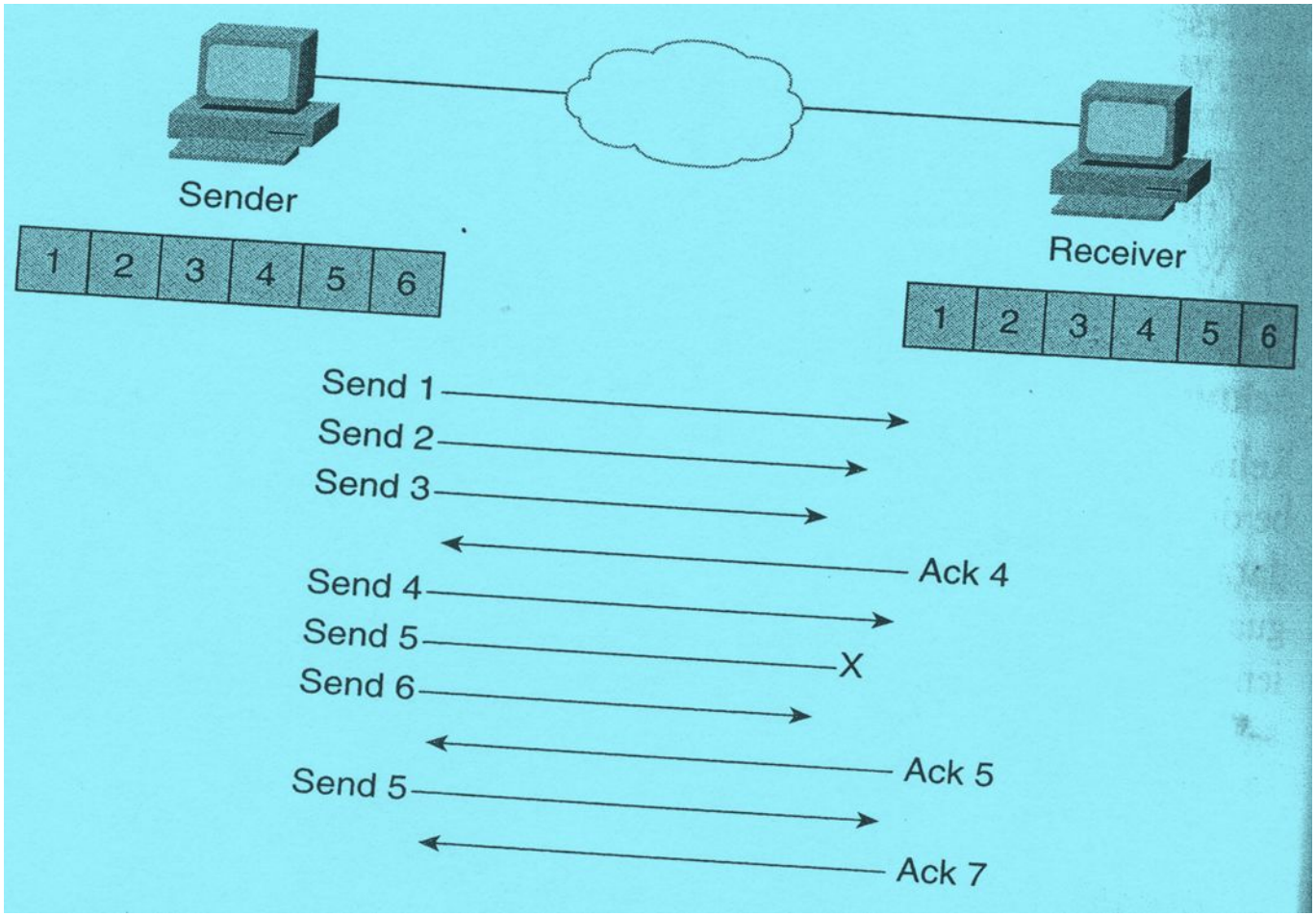

Figure 3, Sender keeps a record of each data packet it sends and waits for an acknowledgment before sending the next data packet

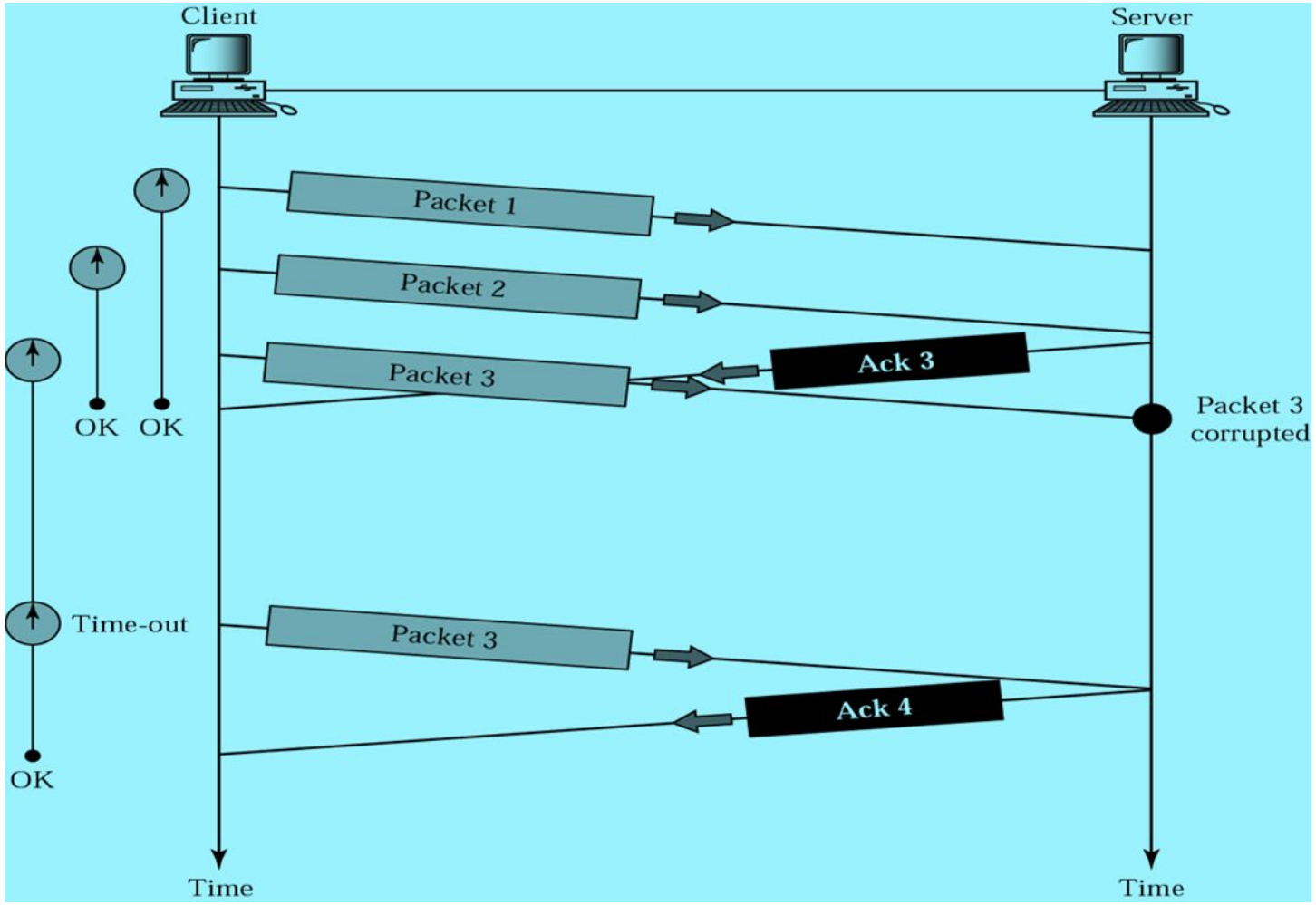

Figure 4: Sender starts a timer when it sends a segment, and retransmits a segment if the timer expires before an acknowledgment arrives

Flow control

- Ensure that segments delivered will be acknowledge.
- Provide retransmission of any segments that are not acknowledged and,
    - Put segments back into their correct sequence at the destination
    - Also provide congestion avoidance and control.

Transport protocols

- The following are transport protocols (see Figure 5):
  - **Transmission control protocol(TCP)** and
  - **User datagram protocol(UDP)**
- TCP provides reliable data transmission between hosts.
- UDP Transports data unreliably between hosts.
- Both TCP and UDP use port numbers to pass information to the upper layers (see Figure 6).
  - Port numbers are used to keep track of different conversations that cross the network at the same time.

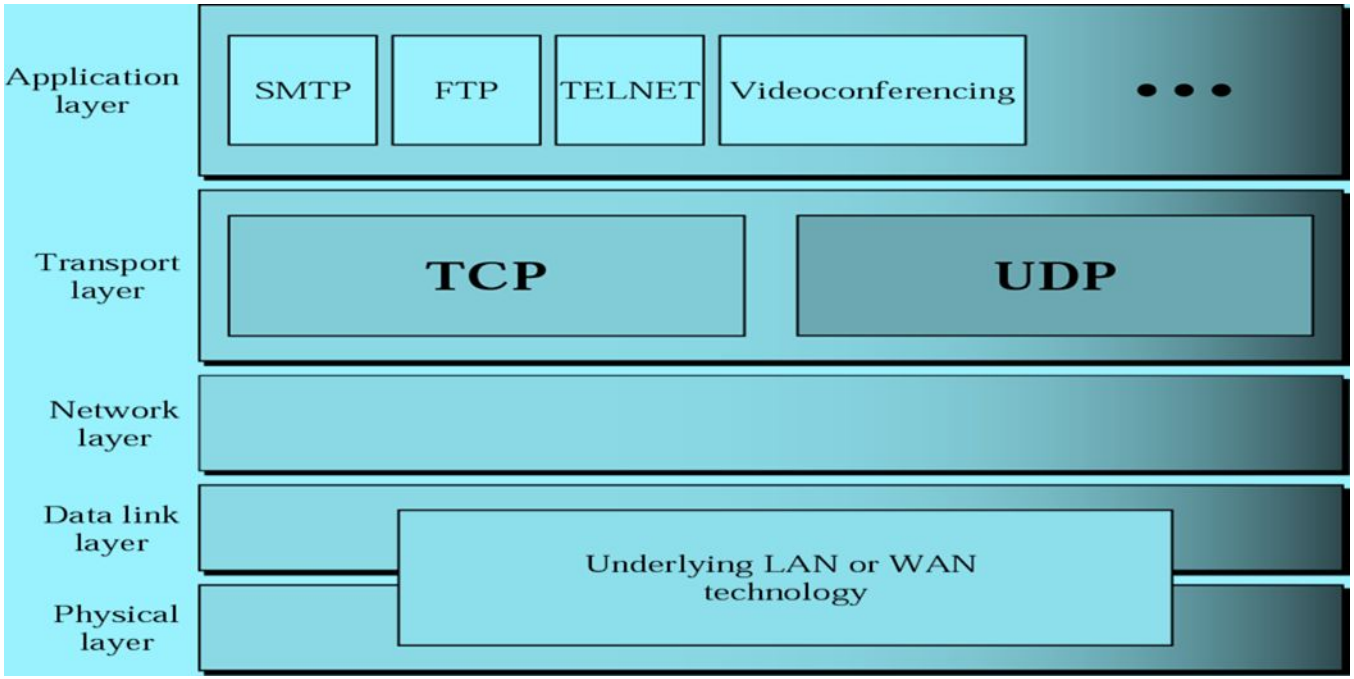

Figure 5:Transport layer protocols

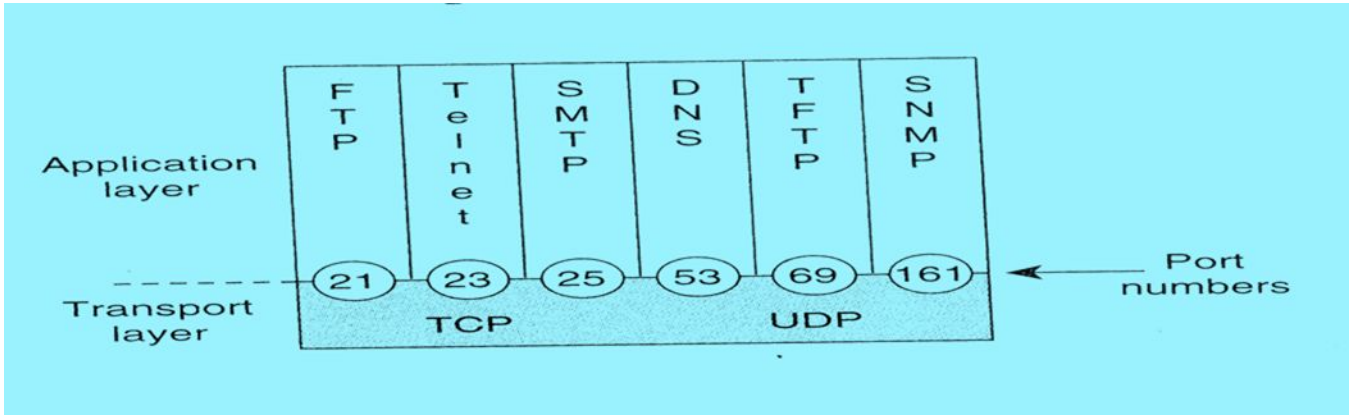

Figure 6: TCP and UDP ports

Transmission Control Protocol

- TCP sets up a connection sometimes called a virtual circuit between end-user applications.
- The following are TCP characteristics:
  - Is connection-oriented
  - Is reliable
  - Divides outgoing messages into segments
  - Reassembles message at the destination station from incoming segments
  - Resends anything not received

TCP specification

- TCP is made reliable with the following:
- **Sequence numbers-** Each TCP packet is sent with a sequence number.
- **Acknowledgment**- Packets contain acknowledgment number.
  - This is the sequence number of the next expected transmitted data byte in the reverse direction.
  - On sending, a host stores the transmitted data in a storage buffer, and starts a timer.
  - If the packet is acknowledged then this data is deleted
  - Else, if no acknowledgment is received before the timer runs out, the packet is retransmitted.
- **Window**- With a host sends a window value which specifies the number of bytes.

TCP Three-Way Handshake

- TCP hosts establish a connection-oriented session with one another using a three-way handshake (see Figure 7).
  - First, one host initiates a connection by sending a packet indicating its initial sequence number of x with a certain bit in the header to indicate a connection request.
  - Second, the other host receives the packet, records the sequence number of x, replies with an acknowledgment of x+1, and includes its own initial sequence number of y.
  - The acknowledgment number of x+1 means that the host has received all octets up to and including x and is expecting x+1 next.

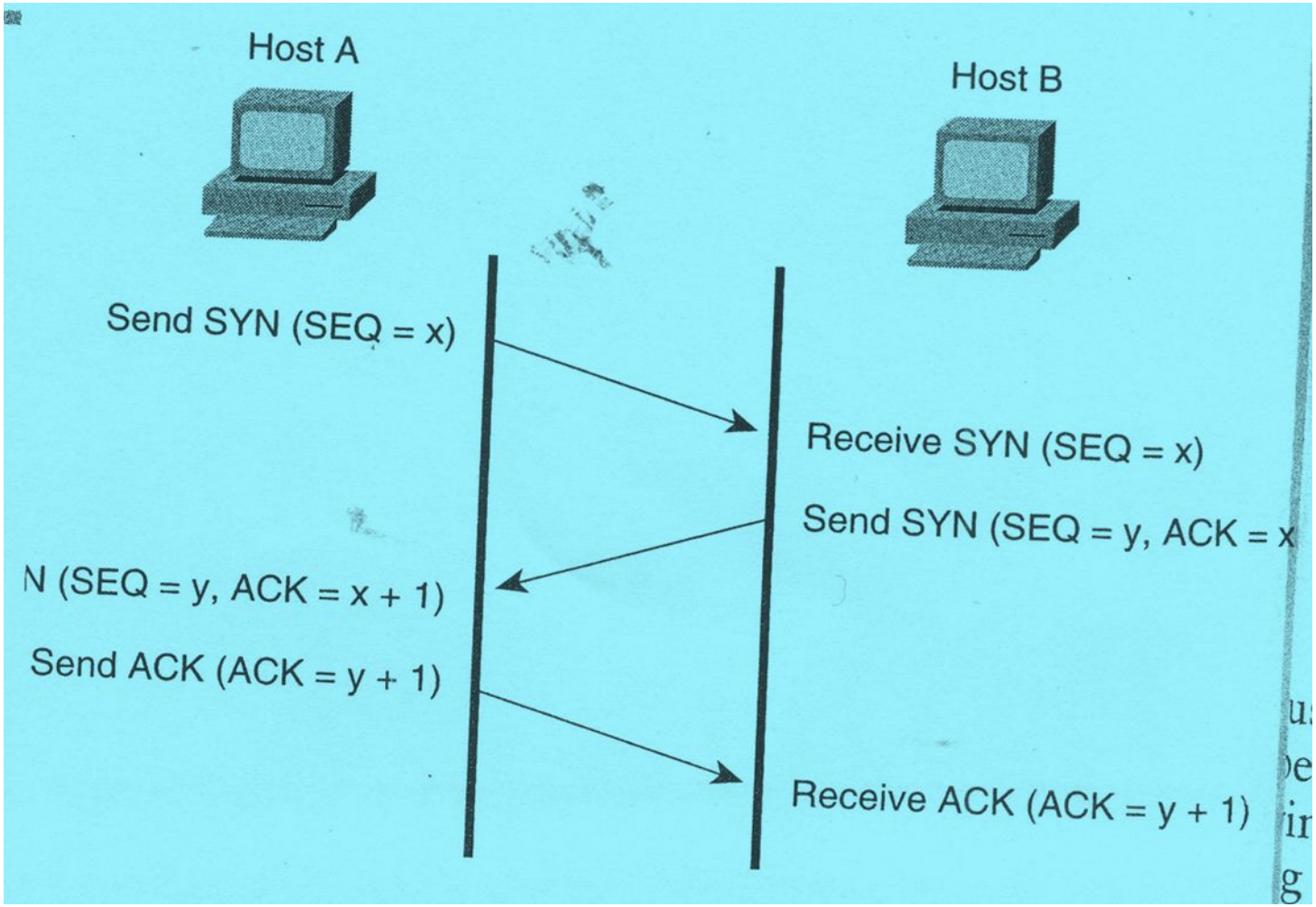

Figure 7: TCP three-way handshake

TCP segment format

- The following are the definitions of the fields in the TCP segment (see Figure 8).

- **Source port-** Number of the calling port. This identifies the upper-layer process at the source end of the segment.
- **Destination port**- Number of the called port. This is the port assigned at the destination end of the segment.
- **Sequence number**- This is the sequence number for the first octet in the user data field.
- **Acknowledgment number**- The next expected TCP octet.
- **HLEN**- Represent bit number in the header.
- **Reserved**- set to zero.

TCP segment format……..

- **Control bits**- Control functions (such as setup and termination of a session).
- **Window**- Number of octets that the device is willing to accept.
- **Checksum**- Calculated checksum of the header and data fields.
- **Urgent pointer**- Indicator for the end of the urgent data.
- **Options**- Maximum TCP segment size.
- **Data**- Upper-layer protocol data.

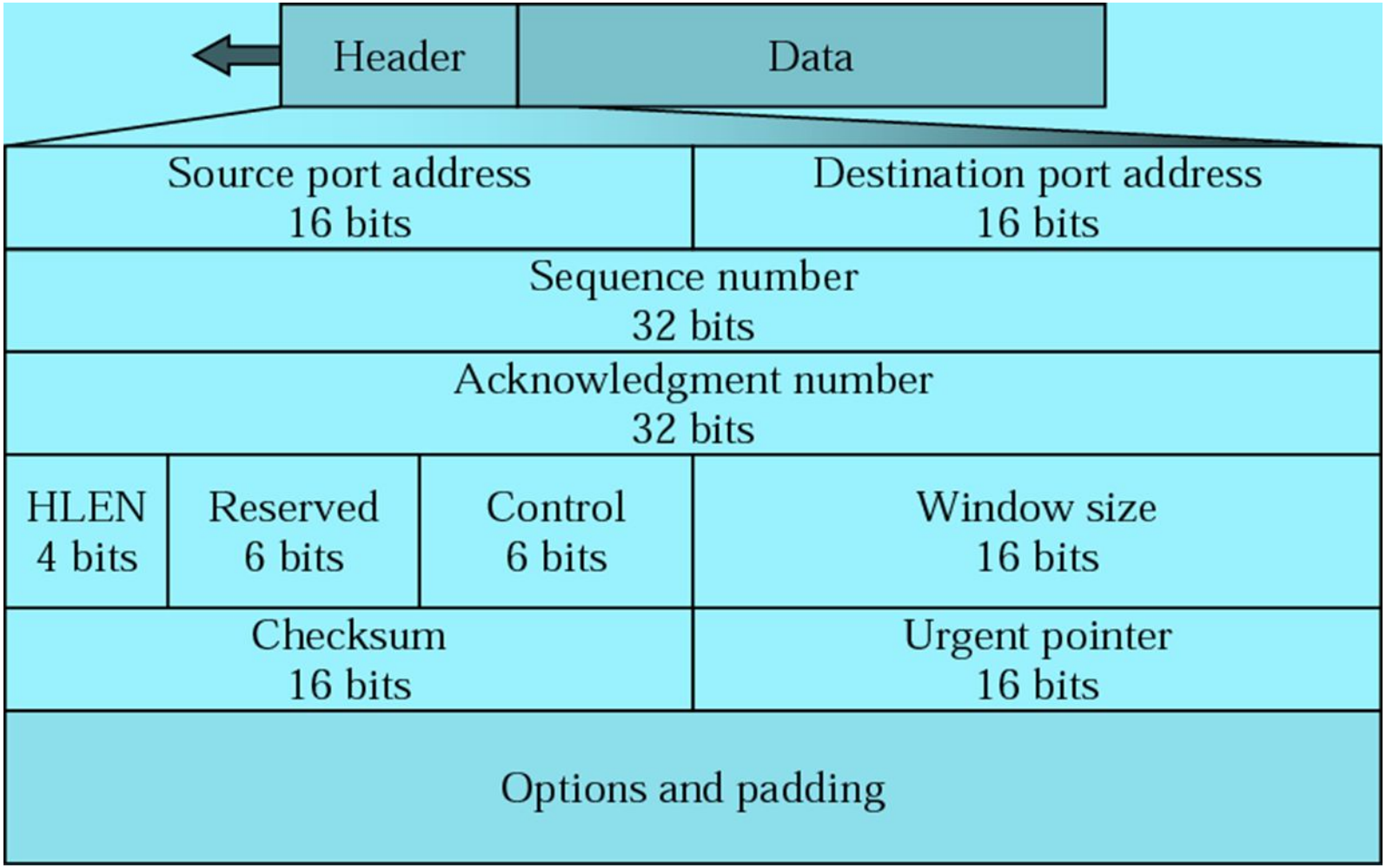

Figure 8:TCP Segment

User datagram protocol

- UDP is a simple protocol that exchanges datagrams, without acknowledgments or guaranteed delivery.
  - Error processing and retransmission must be handled by other protocols.
  - UDP uses no windowing or acknowledgments;
  - Application layer protocols provide the reliability.
  - UDP is designed for application that do not need to put sequences of segments together.

UDP……

- Protocols that use UDP include the following:
  - Trivial File Transfer Protocol (TFTP)
  - Simple Network Management Protocol (SNMP)
  - Dynamic Host Control Protocol (DHCP)
  - Domain Name System (DNS)
  - BOOTP

UDP header format

- The following are the definitions of the fields in the UDP header format (see Figure 9).
- **Source port**-:This identifies the local port number which should be used when the destination host requires to contact originator.
- **Destination port**-:This connects to the destination
- **Length**-: This is a bytes number, including the UDP header and the data.
- **Checksum**-:The 16-bit 1's complement of the 1's complement sum of the UDP header and the data.

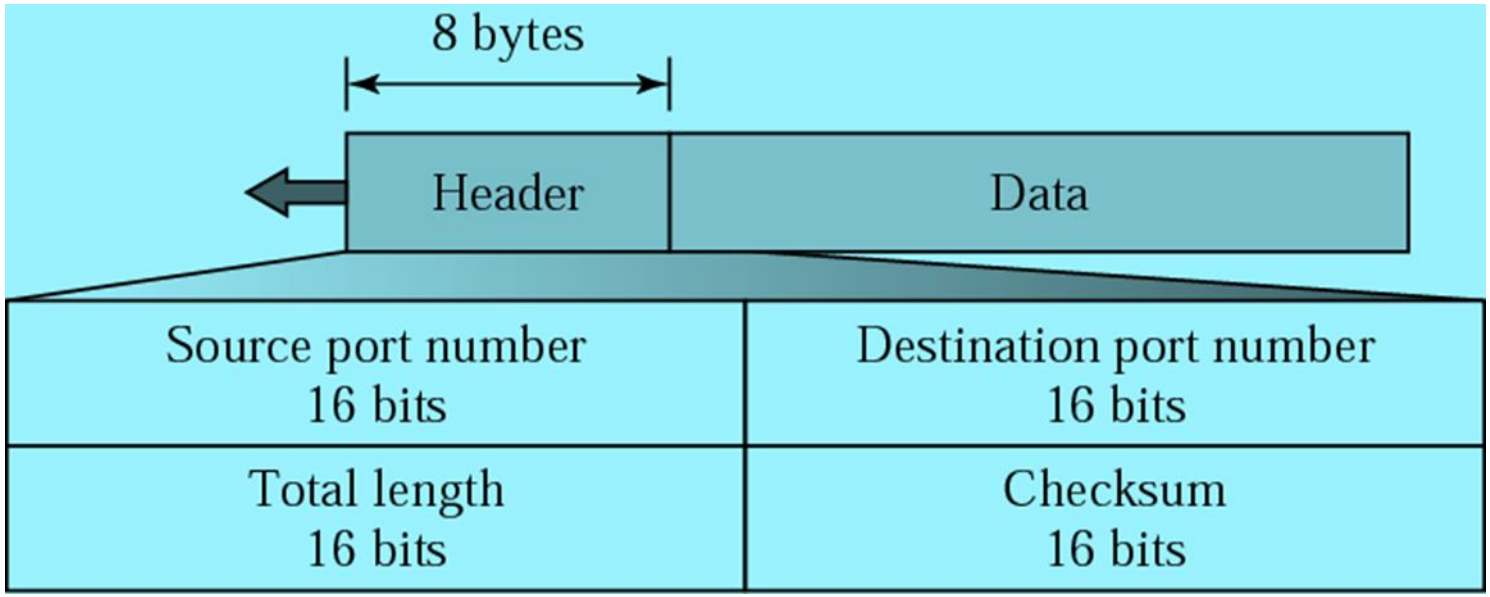

Figure 9: User datagram

END