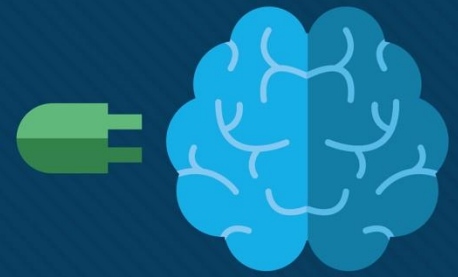


Module 6: Transport Layer

(UDP& TCP)

Transport Layer

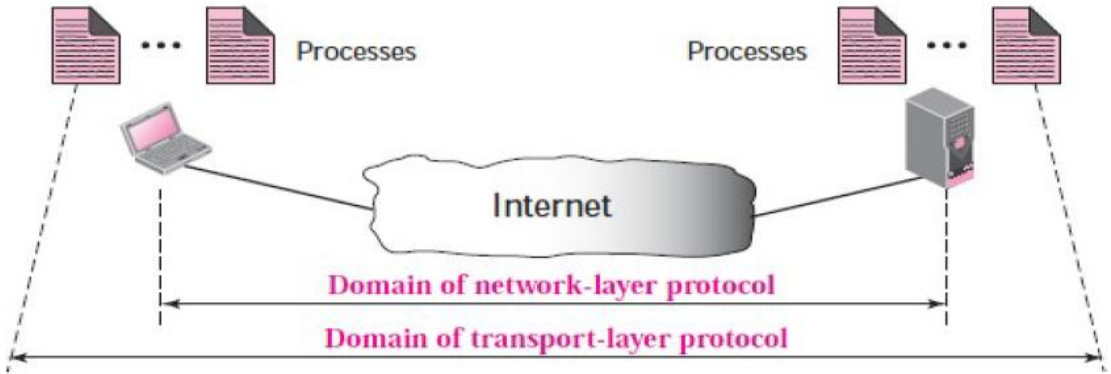
Transport layer provide ***logical communication*** **between application processes** running on different hosts, Transport protocols run in **end systems.**

Network layer: logical communication **between hosts**

Transport layer: logical communication **between processes**

The transport layer deals with:

- The quality-of-service (QoS) issues of reliability
- End-to-end flow control
- Error correction and recovery
- Connection oriented and connectionless.



6.1 UDP Protocol

UDP Protocol

The 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

Q: Why is there a UDP?

- ✓ No connection establishment (which can add delay)
- ✓ Simple: no connection state at the sender, receiver (stateless)
- ✓ Small header size
- ✓ No congestion controls.

UDP Protocol

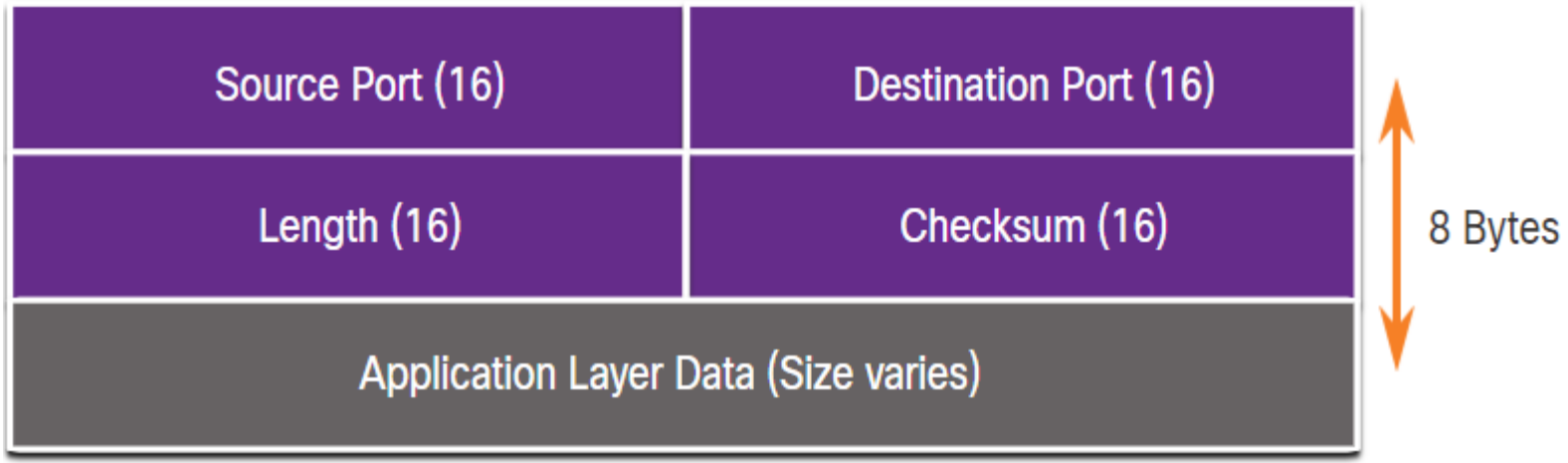
Use of UDP

The following lists some uses of the UDP protocol:

- Routing Information Protocol (**RIP**)
- Domain Name System (**DNS**)
- Streaming **multimedia apps** (loss tolerant, rate-sensitive)
- **SNMP**

UDP Header

The UDP header is far simpler than the TCP header because it only has four fields and requires 8 bytes (i.e. 64 bits).



UDP Header Fields

The table identifies and describes the four fields in a UDP header.

UDP Header Field	Description
Source Port	A 16-bit field used to identify the source application by port number.
Destination Port	A 16-bit field used to identify the destination application by port number.
Length	A 16-bit field that indicates the length of the UDP datagram header.
Checksum	A 16-bit field used for error checking of the datagram header and data.

UDP Example

Example 6-1

The following is a dump of a UDP header in hexadecimal format:

CB84000D001C001C

a. What is the source port number?

b. What is the destination port number?

c. What is the total length of the user datagram?

d. What is the length of the data?

UDP Example

Solution

Datagram	CB84000D001C001C															
Hexa	C	B	8	4	0	0	0	D	0	0	1	C	0	0	1	C
Binay	1100	1011	1000	0100	0000	0000	0000	1101	0000	0000	0001	1100	0000	0000	0001	1100
Decimal	52100				13				28				28			
	Source port number				Destination port number				Total length				Checksum			

a. The source port number is the first four hexadecimal digits (CB84), which means that the source port number is 52100.

b. The destination port number is the second four hexadecimal digits (000D), which means that the destination port number is 13.

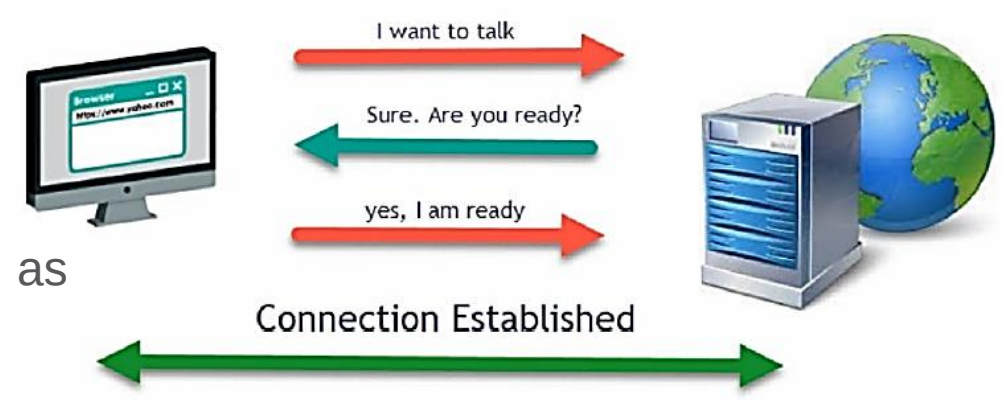
c. The third four hexadecimal digits (001C) define the length of the whole UDP packet as 28 bytes.

d. The length of the data is the length of the whole packet minus the length of the header, or 28 – 8 = 20 bytes.

6.2 TCP Protocol

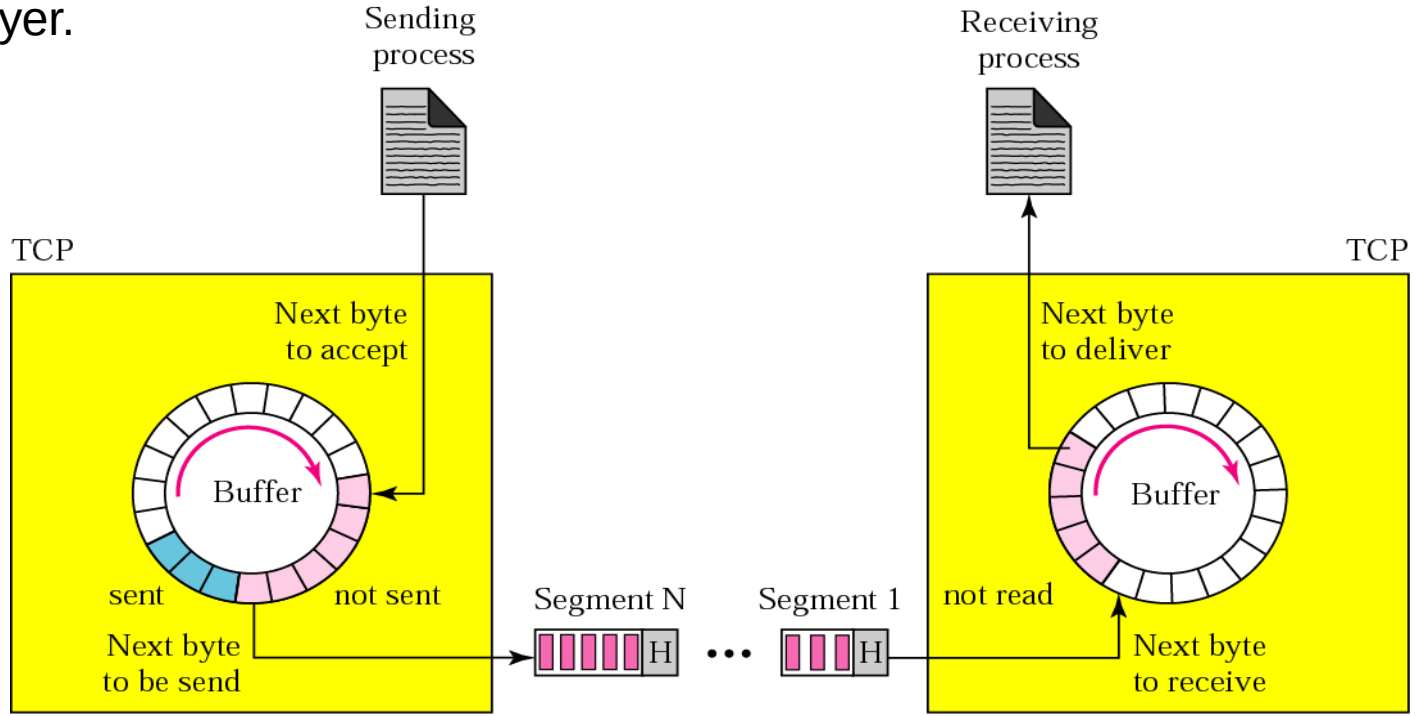
TCP Protocol

- **Transmission Control Protocol** (TCP) is a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.
- **TCP** is a connection-oriented protocol; it creates a virtual connection between two TCPs send data.
- **TCP** uses flow and error control mechanisms at the transport level
- **TCP** is a **Three-Way Handshake** as shown in figure:

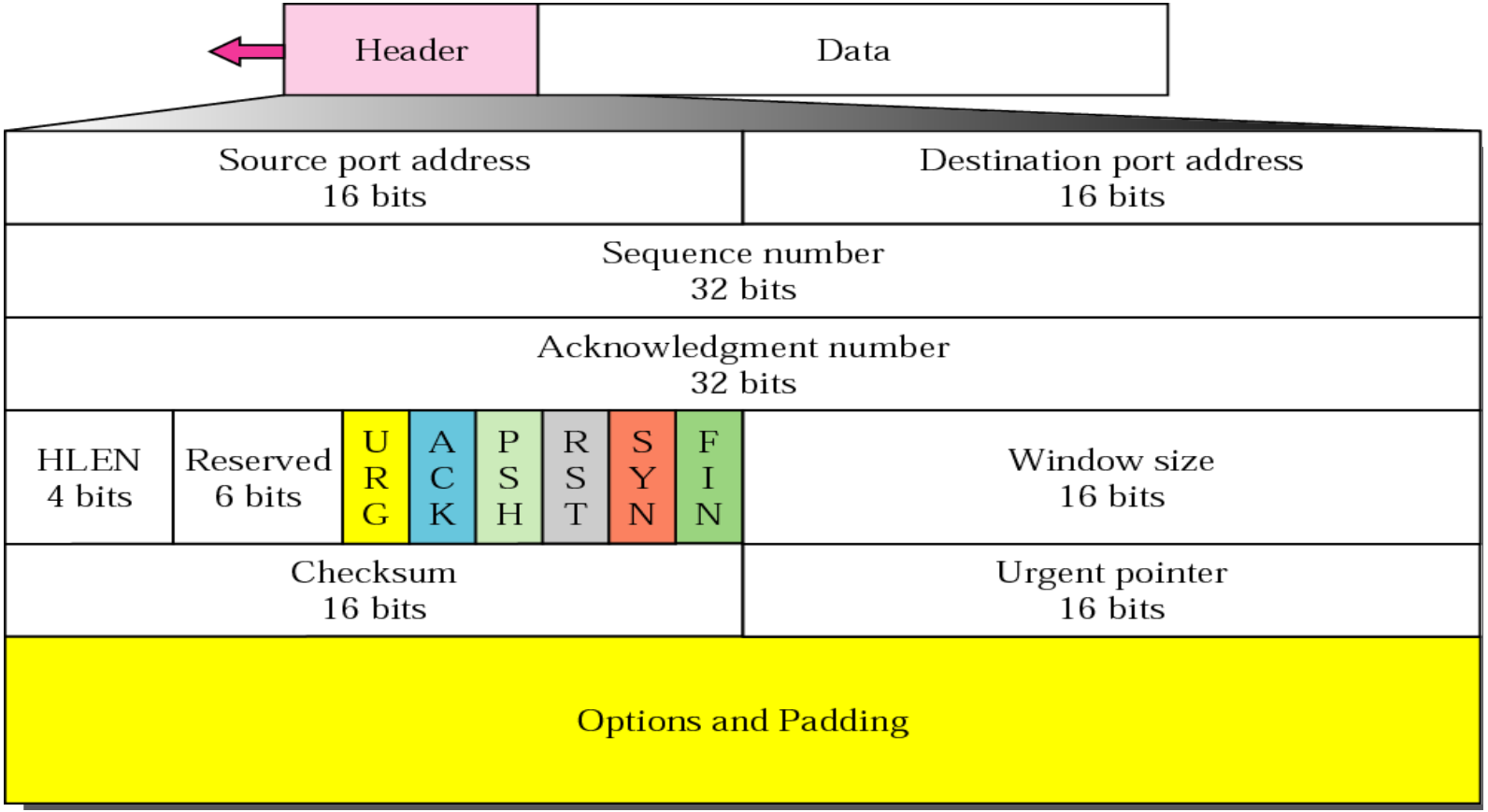


TCP segment

TCP provides a stream delivery service. It breaks up the data stream into segments of variable size. Each segment receives a header and is handed off to the IP layer.



TCP segment format:



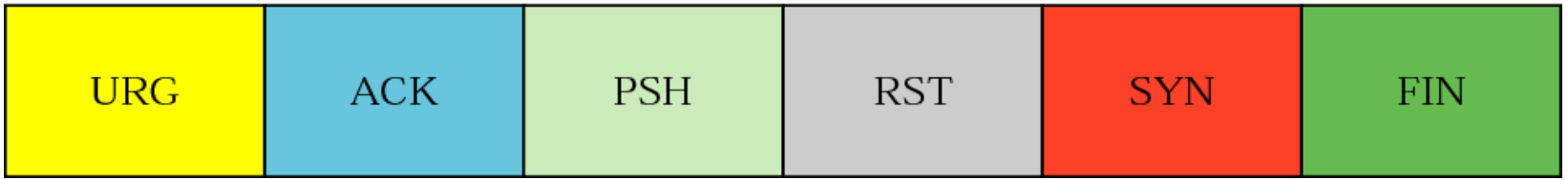
TCP segment format

TCP Header Field	Description
Source Port	A 16-bit field used to identify the source application by port number.
Destination Port	A 16-bit field used to identify the destination application by port number.
Sequence Number	A 32-bit field used for data reassembly purposes.
Acknowledgment Number	A 32-bit field used to indicate that data has been received and the next byte expected from the source.
Header Length (HLEN)	A 4-bit field known as "data offset" that indicates the length of the TCP segment header.
Reserved	A 6-bit field that is reserved for future use.
Control bits	A 6-bit field used that includes bit codes, or flags, which indicate the purpose and function of the TCP segment (shown in figure in next slide) .
Window size	A 16-bit field used to indicate the number of bytes that can be accepted at one time.
Checksum	A 16-bit field used for error checking of the segment header and data.
Urgent	A 16-bit field used to indicate if the contained data is urgent.

TCP segment format

Description of flags in the control field

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



TCP Example

Example 6-2

Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data is sent in five segments, each carrying 1000 bytes?

Solution:

The following shows the sequence number for each segment:

Segment 1 ➡ Sequence Number: 10,001 (range: 10,001 to 11,000)

Segment 2 ➡ Sequence Number: 11,001 (range: 11,001 to 12,000)

Segment 3 ➡ Sequence Number: 12,001 (range: 12,001 to 13,000)

Segment 4 ➡ Sequence Number: 13,001 (range: 13,001 to 14,000)

Segment 5 ➡ Sequence Number: 14,001 (range: 14,001 to 15,000)

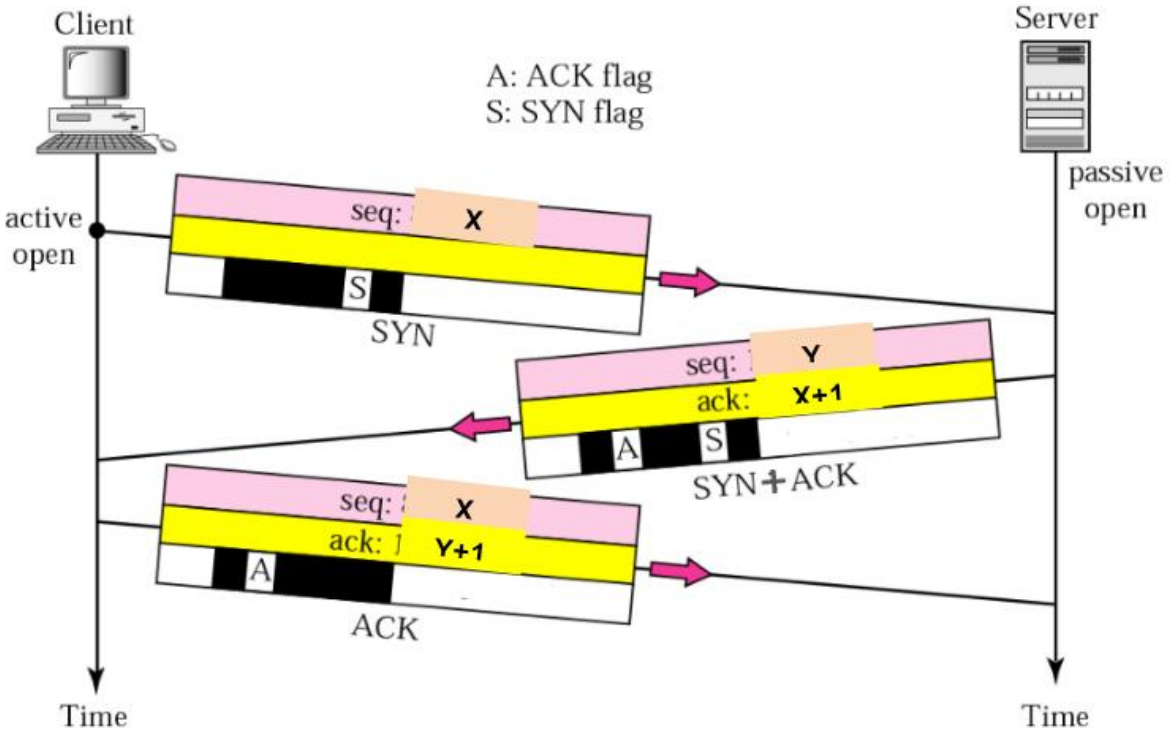
TCP connection

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All of the segments belonging to a message are then sent over this virtual path. A connection-oriented transmission requires three phases:

1. connection establishment
2. data transfer
3. connection termination.

TCP connection

1. Connection establishment using three-way handshaking



A server tells its TCP that it is ready to make a connection - this is called a passive open.

Note:
SYN bit is set in first packet.
X=Seq number
Y=Ack number

TCP connection

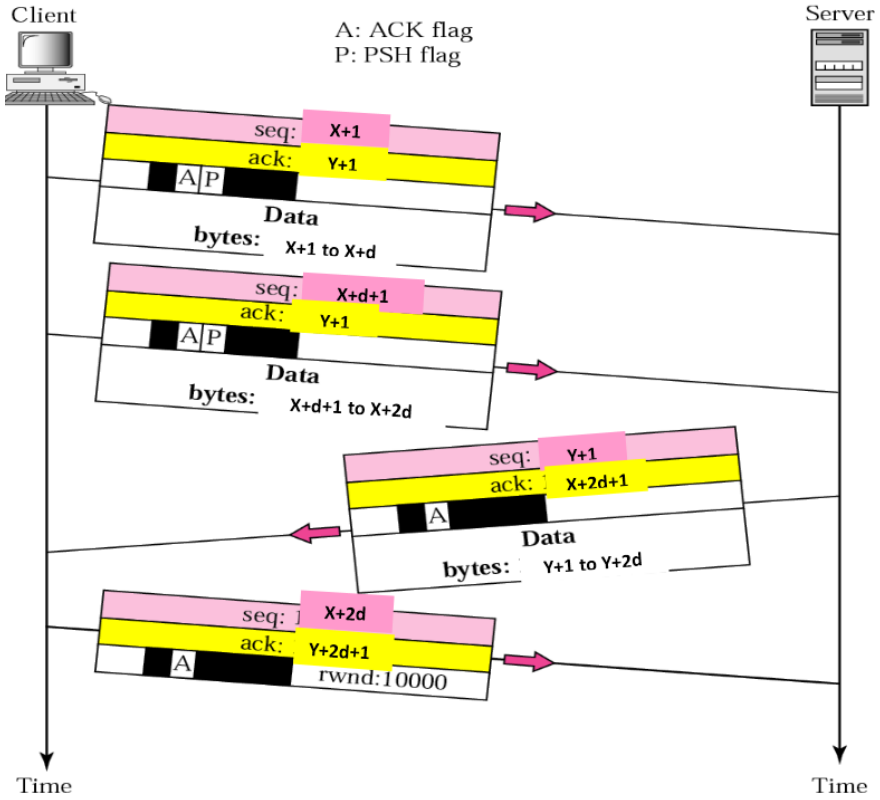
A SYN segment cannot carry data, but it consumes one sequence number.

A SYN + ACK segment cannot carry data, but does consume one sequence number.

An ACK segment, if carrying no data, consumes no sequence number.

TCP: Data transfer

2. Data transfer using three-way handshaking

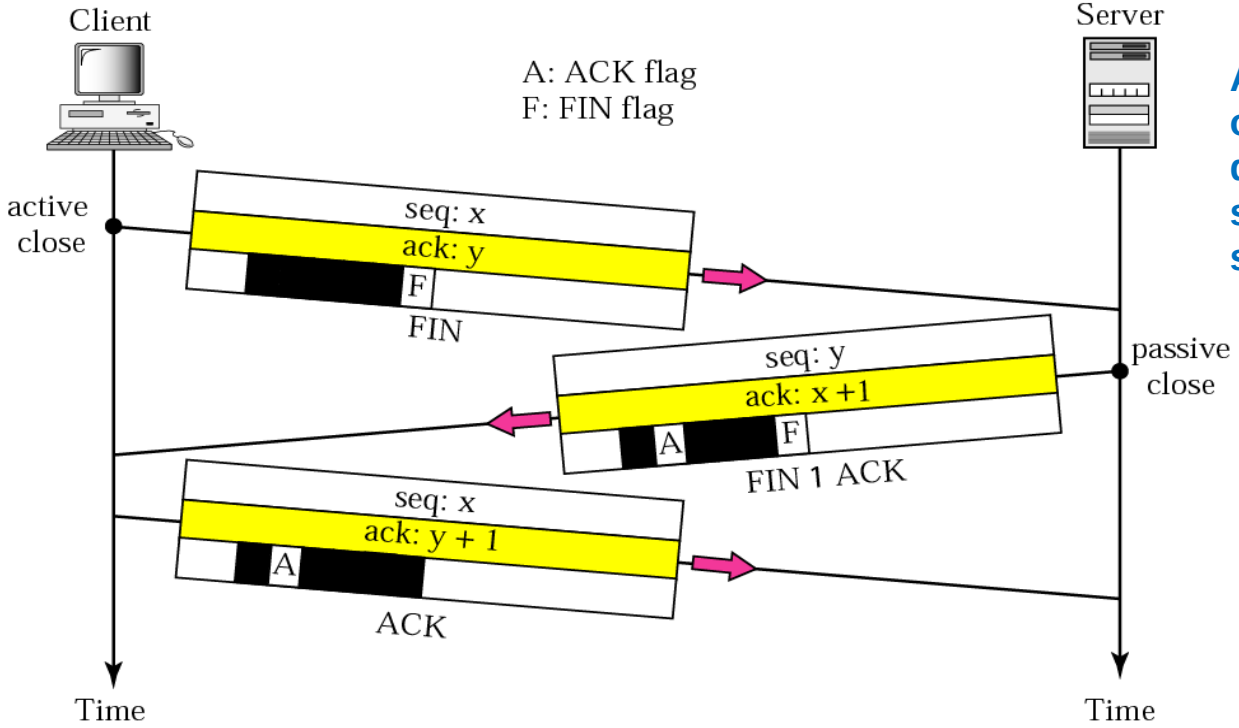


Push flag means deliver the data to the receiver as soon as it is received (don't put it in a buffer and hold until you have enough bytes for a complete segment).
This feature is usually ignored.

Note: d is the segments of data transferred

TCP: Connection terminated

3. Connection termination using three-way handshaking



A FIN segment consumes one sequence number if it does not carry data. So should third segment be seq: x+1?

TCP Example

Example 6-3

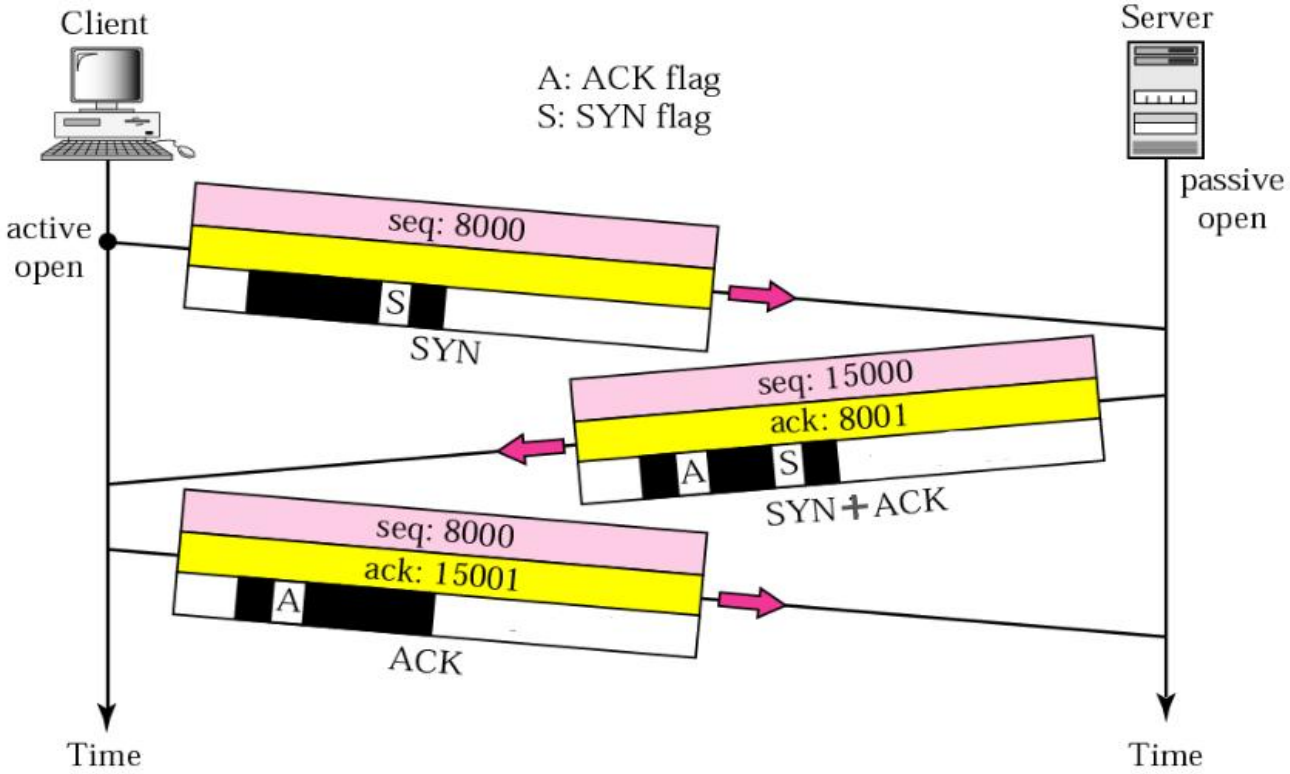
Draw the mechanism of three-way handshaking of the TCP protocol to transfer 2 MB of data in two segments each carrying 1 MB (1000 byte) of data, if you know that the sequence number (8000), the acknowledgment number (15000). According to the following:

1. Connection establishment
2. Data transfer
3. Connection termination

TCP Example

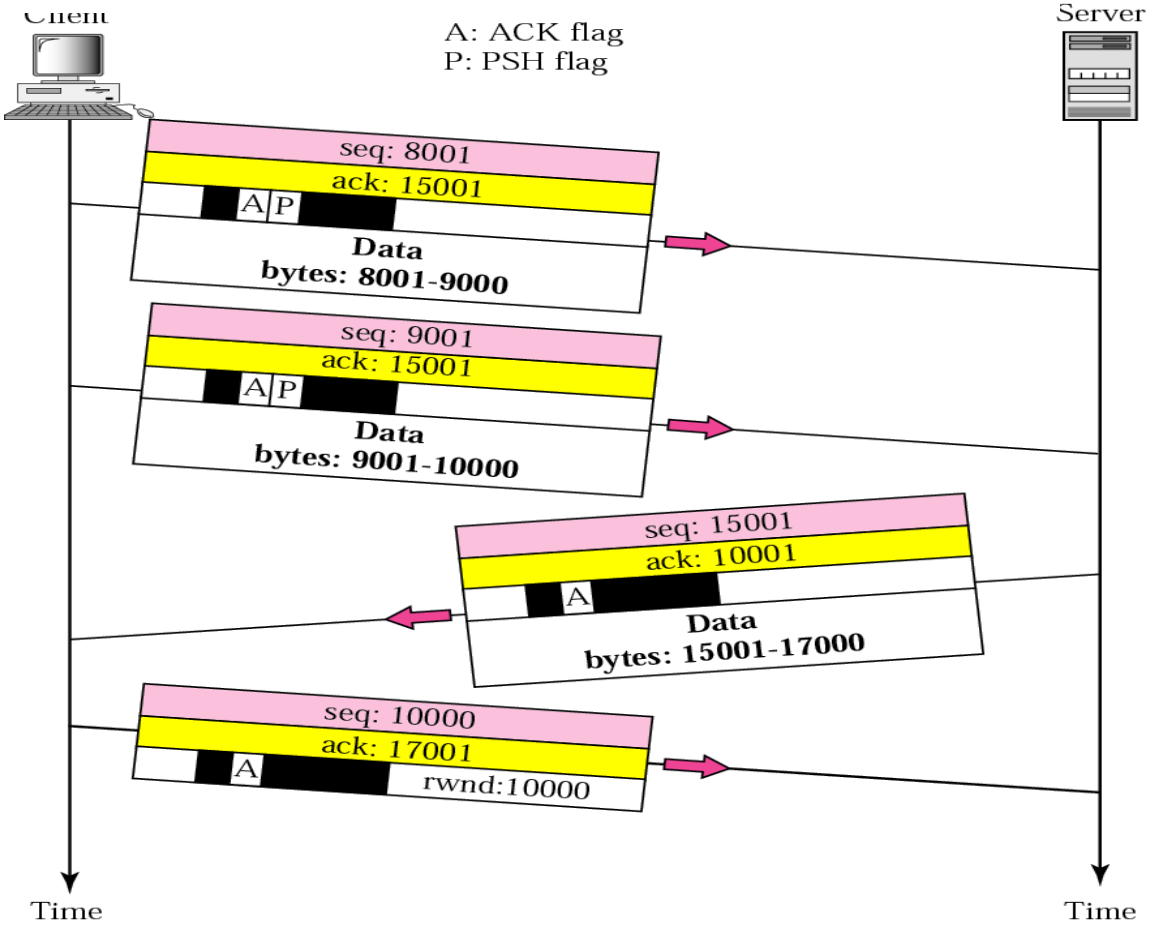
Solution

1. Connection establishment



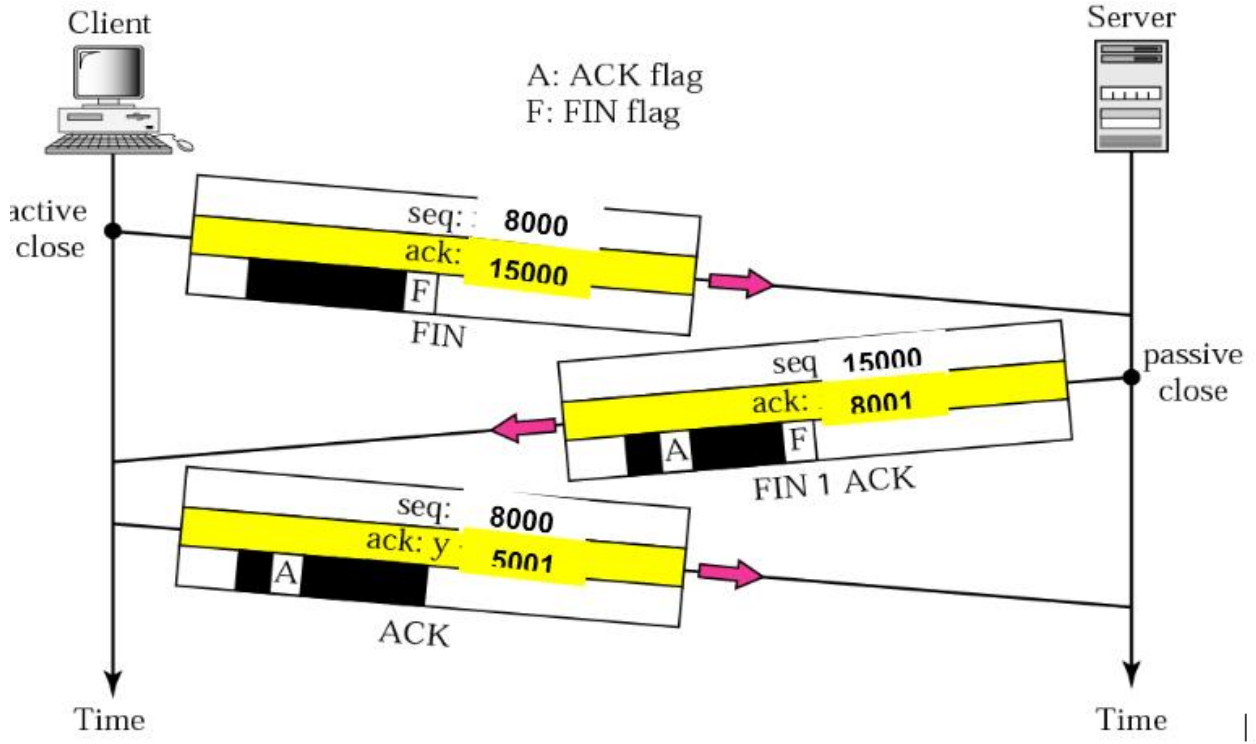
TCP Example

2. Data transfer

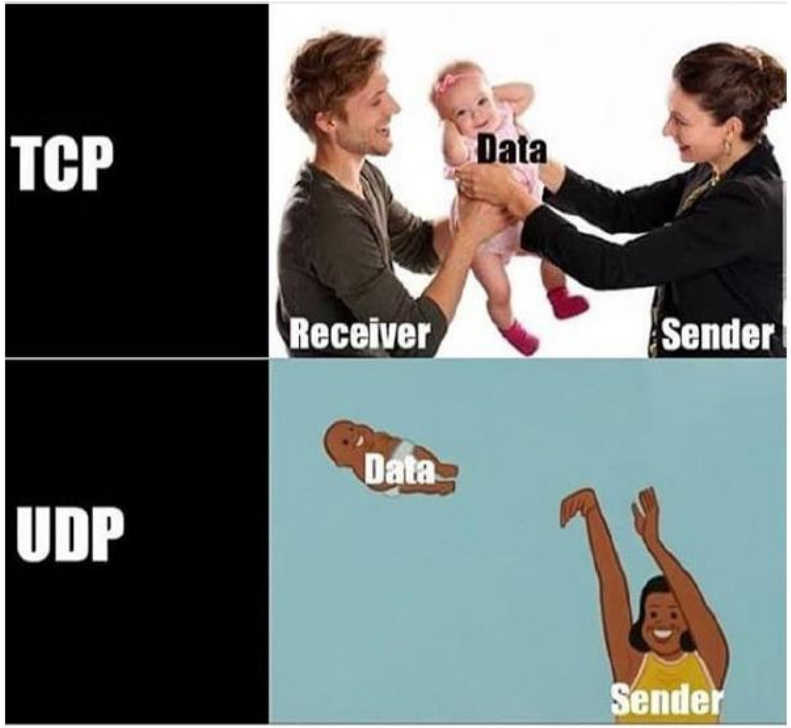


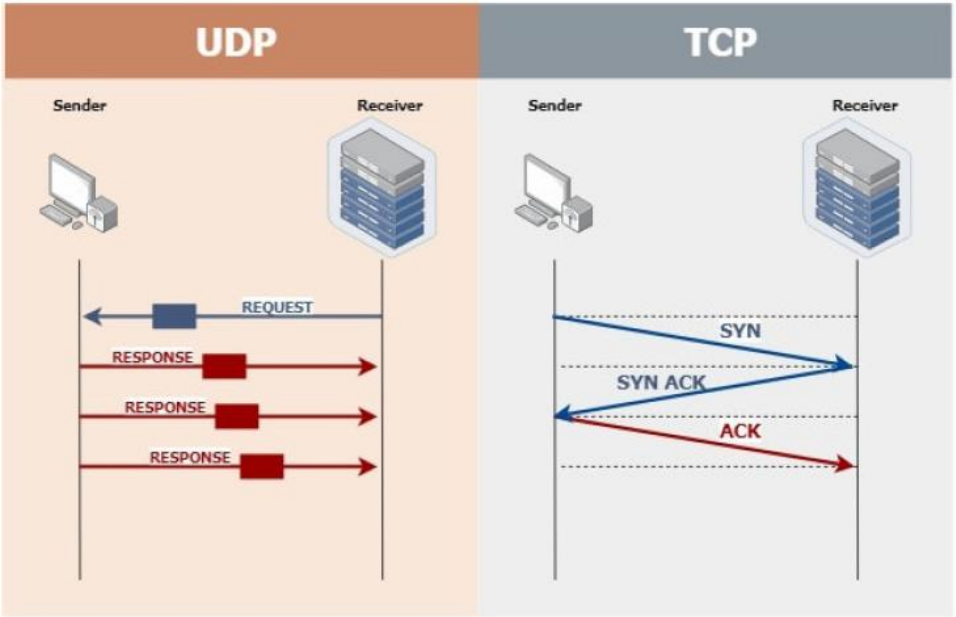
TCP Example

3. Data termination



TCP vs. UDP





TCP vs. UDP

	TCP	UDP
Connection	TCP is a connection-oriented protocol.	UDP is a connectionless protocol.
Ordering of data packets	TCP rearranges data packets in the order specified.	UDP has no inherent order as all packets are independent of each other.
Speed of transfer	The speed for TCP is slower than UDP.	UDP is faster because there is no error-checking for packets.
Reliability	There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent.	There is no guarantee that the messages or packets sent would reach at all.
Header Size	TCP header size is 20 bytes	UDP Header size is 8 bytes.

TCP vs. UDP (contd.)

	TCP	UDP
Streaming of data	Data is read as a byte stream,	Packets are sent individually
Data Flow Control	TCP does Flow Control. TCP requires three packets to set up a socket connection,	UDP does not have an option for flow control
Acknowledgement	Acknowledgement segments	No Acknowledgment