TRANSPORT LAYER

TCP segment structure

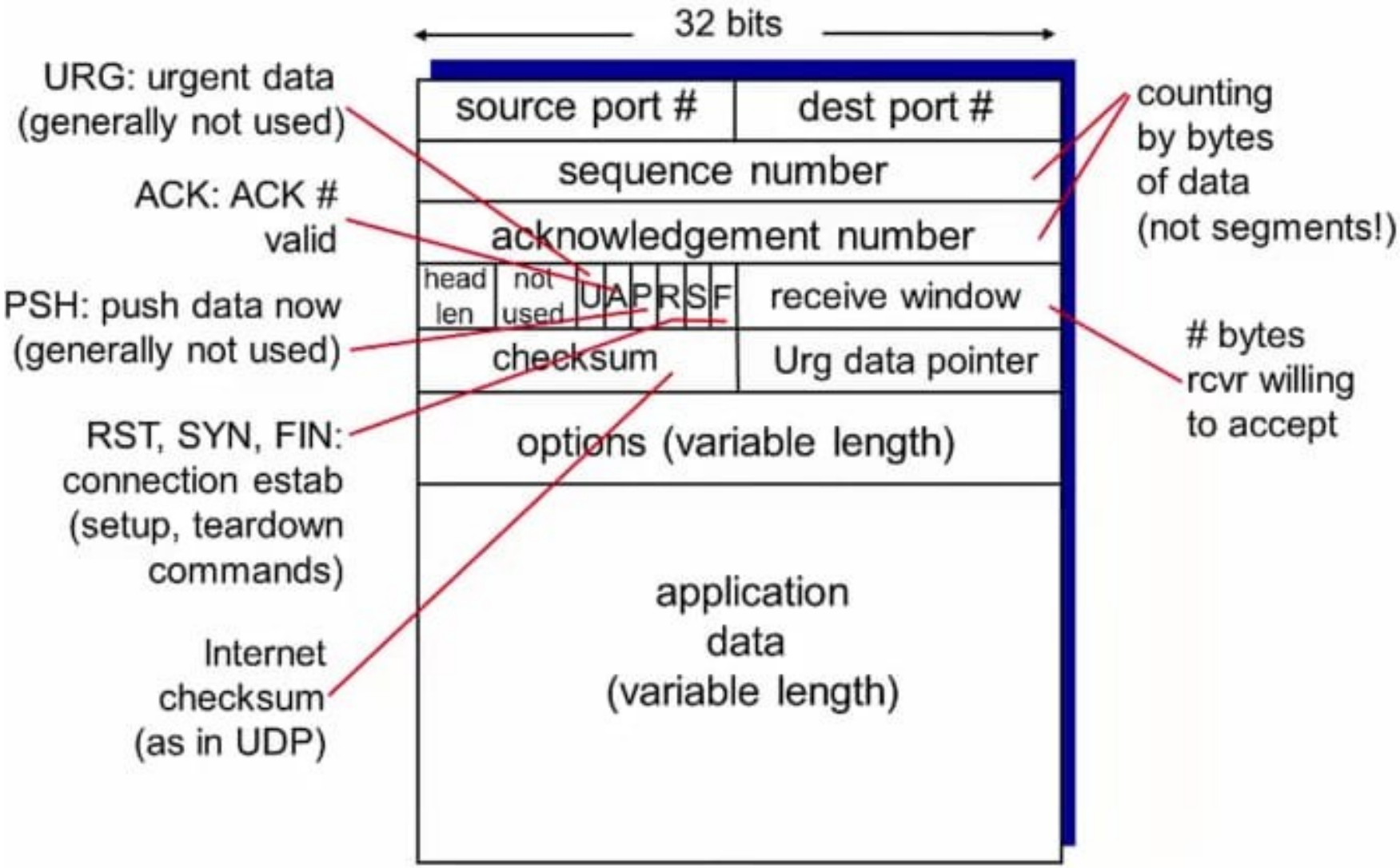
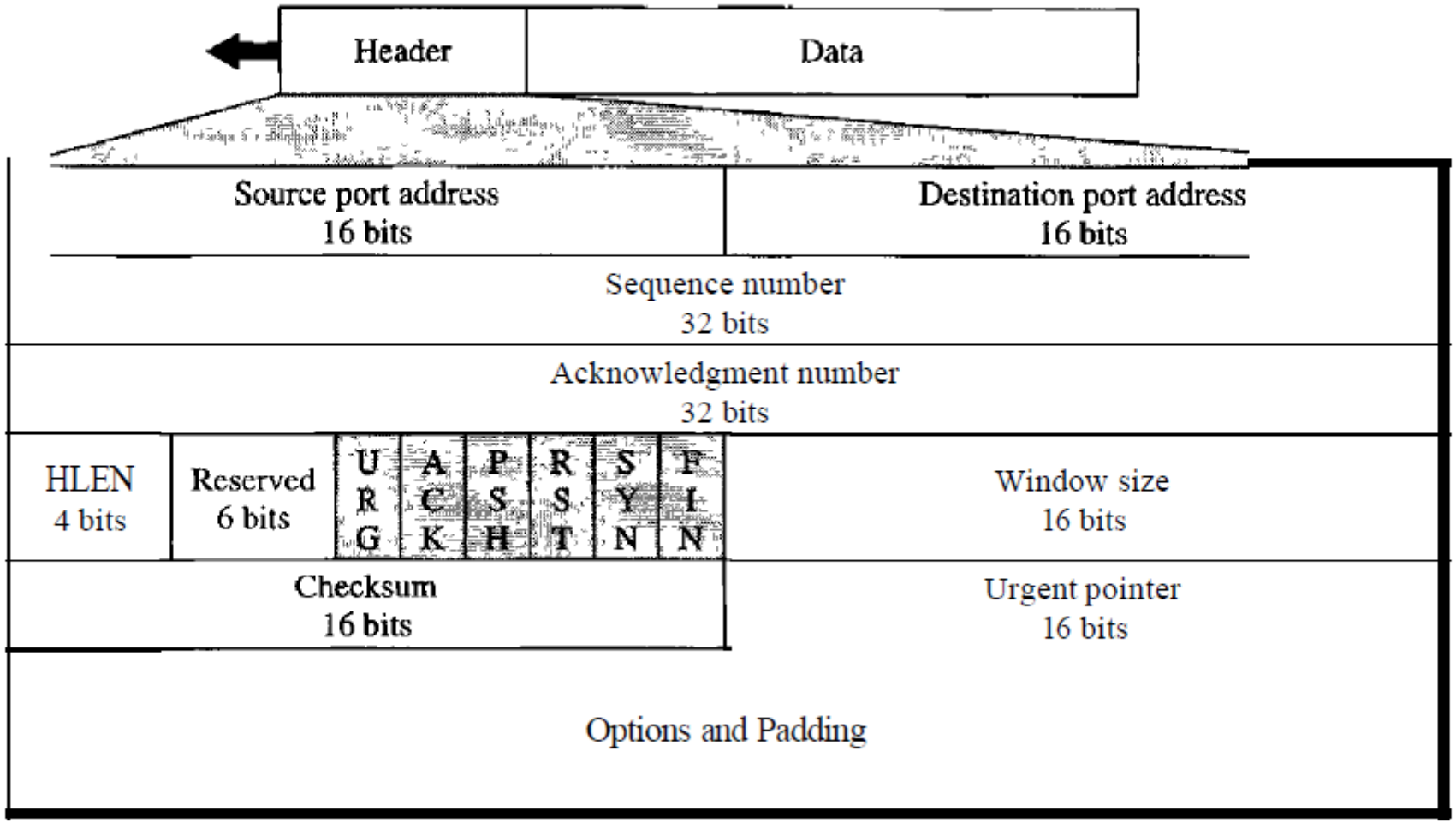


Figure 23.16 *TCP segmentformat*



The following is the header dump of TCP transmission Header in Hexadecimal

E2930017000000010000000500207FF....

Find the Source Port Number?

Find the Destination Port N

Sequence Number?

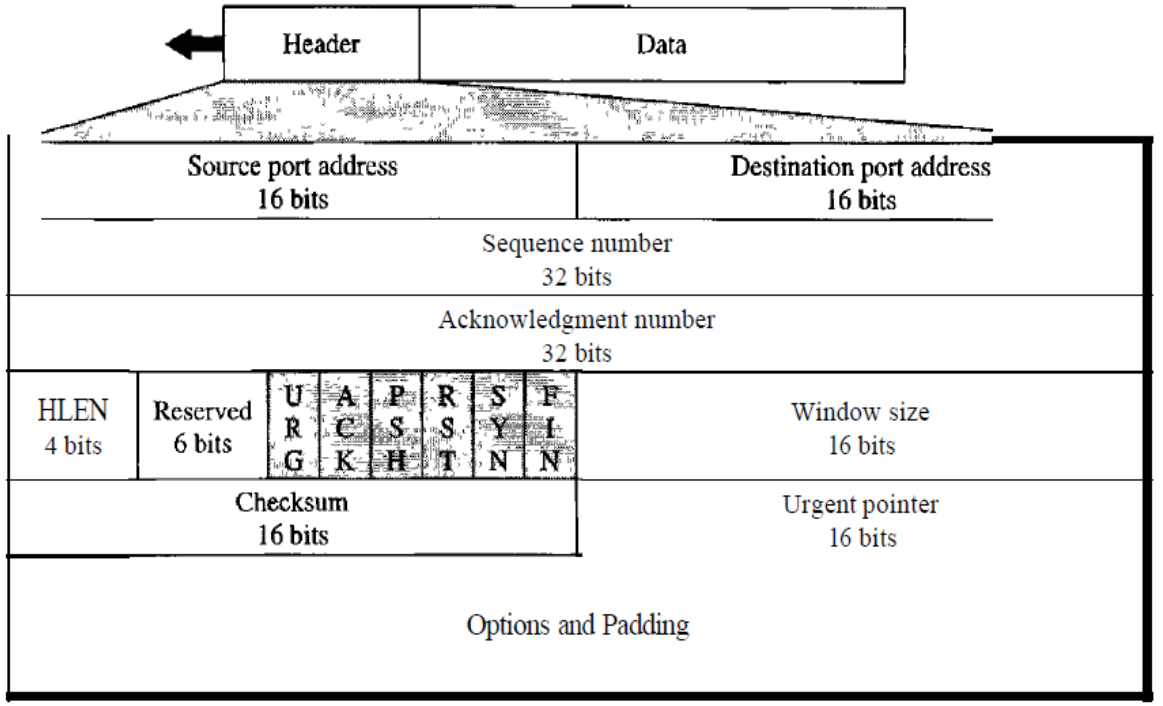
Acknowledgement Number

Length of the Header

Type of the Segment

What is the Window size

Figure 23.16 *TCP segmentformat*



E2930017000000010000000500207FF....

Find the Source Port Number?: E293- 58002

Find the Destination Port Number? 0017- 23

Sequence Number?00000001-1

Acknowledgement Number-00000000- 0

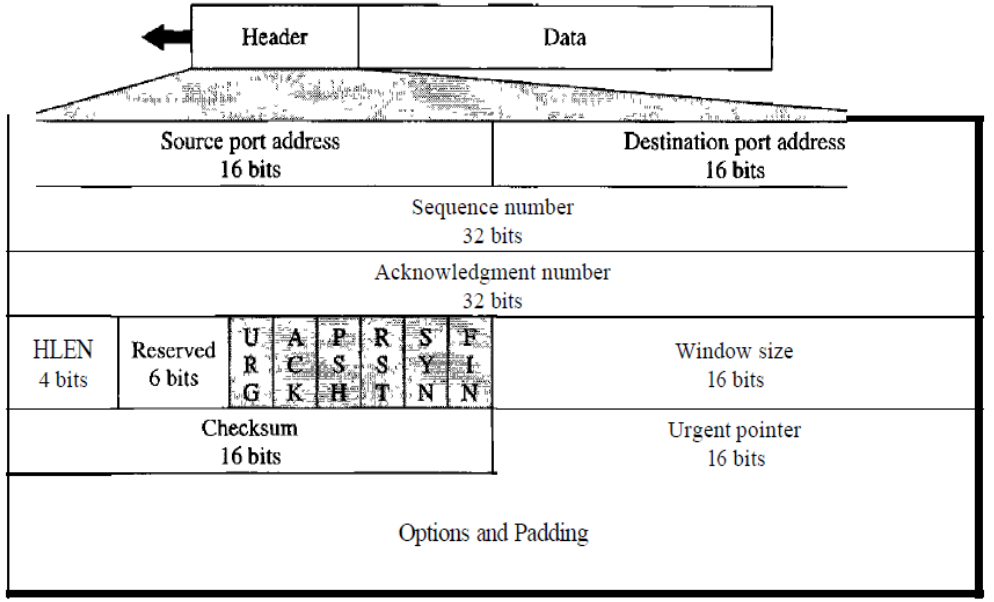
Length of the Header- 5 (5*4=20bytes)

Type of Segment- 002

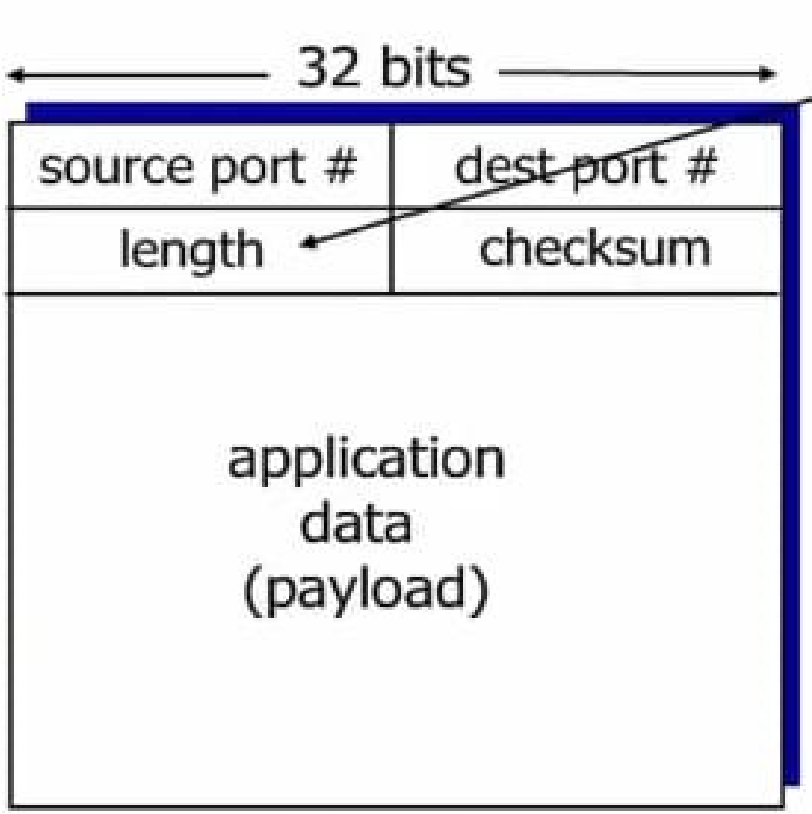
What is the Window size- 07FF- 2047

Type of the Segment-002-
in binary 000000
000010
1- SYN

Figure 23.16 *TCP segmentformat*



UDP: segment header



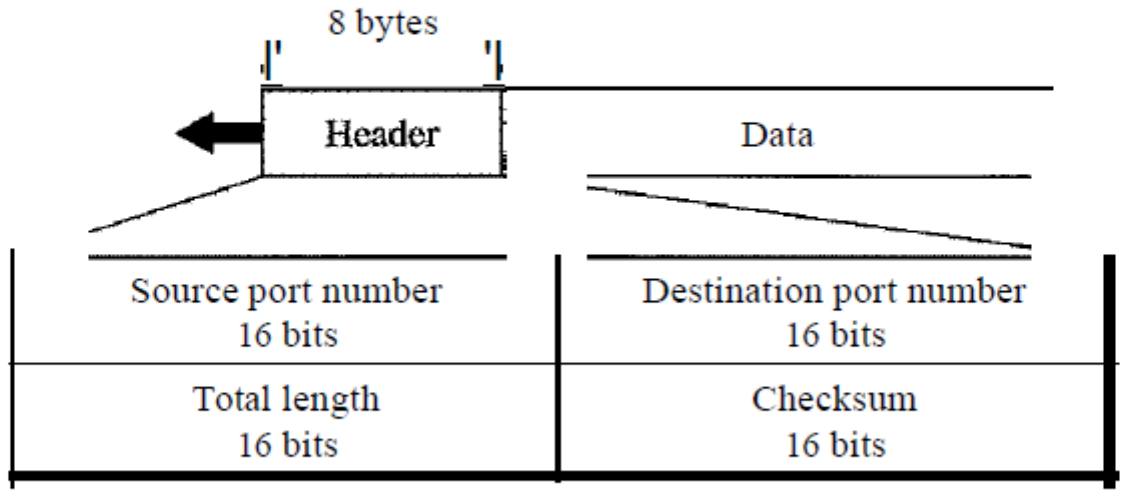
UDP segment format

length, in bytes of UDP segment, including header

why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired

Figure 23.9 *User datagram format*



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

0632000DOO lCE217

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?

The follov

The following is the header dump of UDP transmission Header in Hexadecimal

CB84000D001C001C

Find the Source Port Number?

Find the Destination Port Number?

Total Length of the User Datagram?

Total Length of the Data?

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

What is the client Process?

CB84000D001C001C

Find the Source Port Number?- CB84- 52100

Find the Destination Port Number?- 000D - 13

Total Length of the User Datagram?-001C- 28 BYTES

Total Length of the Data?

Is the packet directed from client to server or vice versa?(0-1023)-13(destination)

CLIENT TO SERVER

What is the client Proces

Day Time (refer next slid

Figure 23.9 *User datagram format*

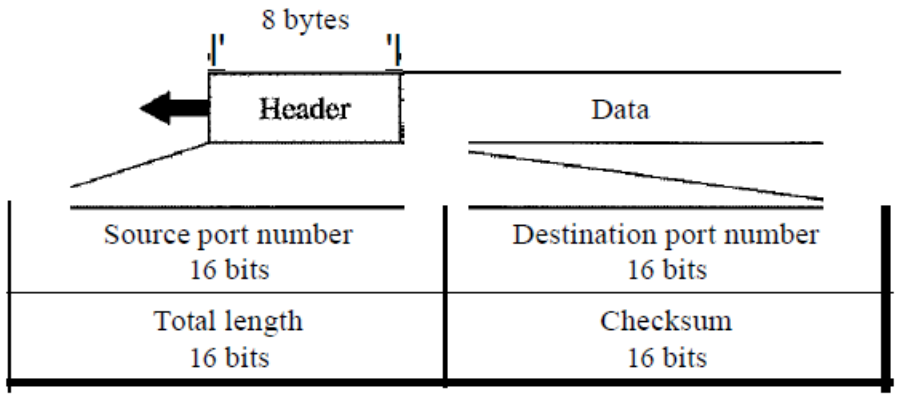


Table 23.1 *Well-known ports used with UDP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users

Port	*Protocol*	*Description*
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

The following is the contents of a UDP header in hexadecimal format:

0045DF0000580000

(GIVE ALL NUMERIC ANSWERS IN DECIMAL)

a. What is the source port number?
b. What is the destination port number?
c. What is the total length of the user datagram, in bytes?
d. What is the length of the data, in bytes?
e. Is the packet directed from a client to a server or vice versa?
f. What is the Application Layer protocol (based on the port numbers)?
g. Has the sender calculated a checksum for this datagram?

- UDP is a message-oriented protocol. A process delivers a message to UDP, which is encapsulated in a user datagram and sent over the network. UDP *conserves the message boundaries;* each message is independent of any other message.
- This is a desirable feature when we are dealing with applications such as IP telephony and transmission of real-time data. However, UDP is unreliable; the sender cannot know the destiny of messages sent. A message can be lost, duplicated, or received out of order. UDP also lacks some other features, such as congestion control and flow control, needed for a friendly transport layer protocol.

- TCP is a byte-oriented protocol. It receives a message or messages from a process stores them as a stream of bytes, and sends them in segments. There is no preservation of the message boundaries. However, TCP is a reliable protocol.
- The duplicate segments are detected, the lost segments are resent, and the bytes are delivered to the end process in order. TCP also has congestion control and flow control mechanisms.