COMPUTER COMMUNICATION NETWORKS

Department of Electronics and
Communication Engineering

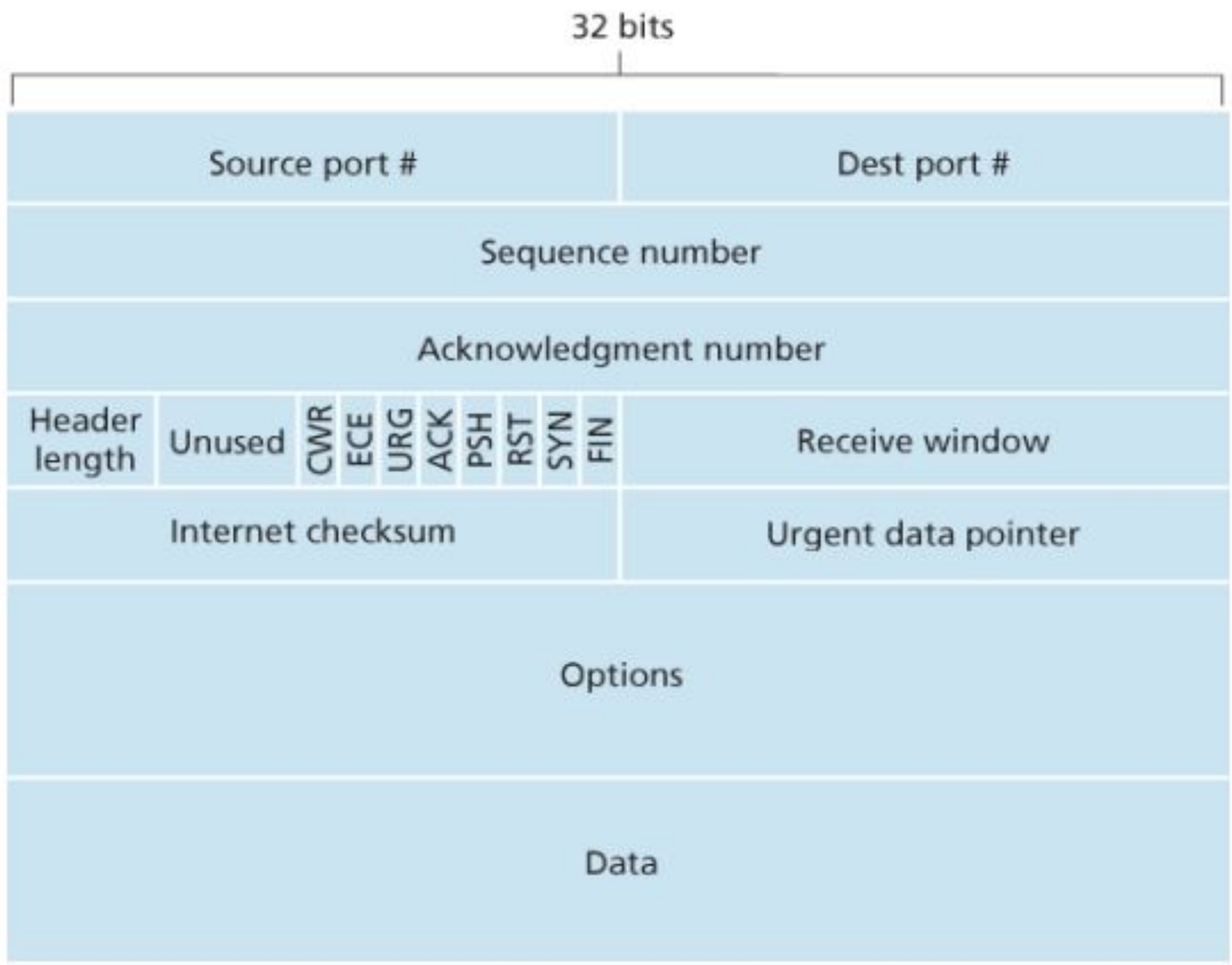
COMPUTER COMMUNICATION NETWORKS

TCP Segment Format - details

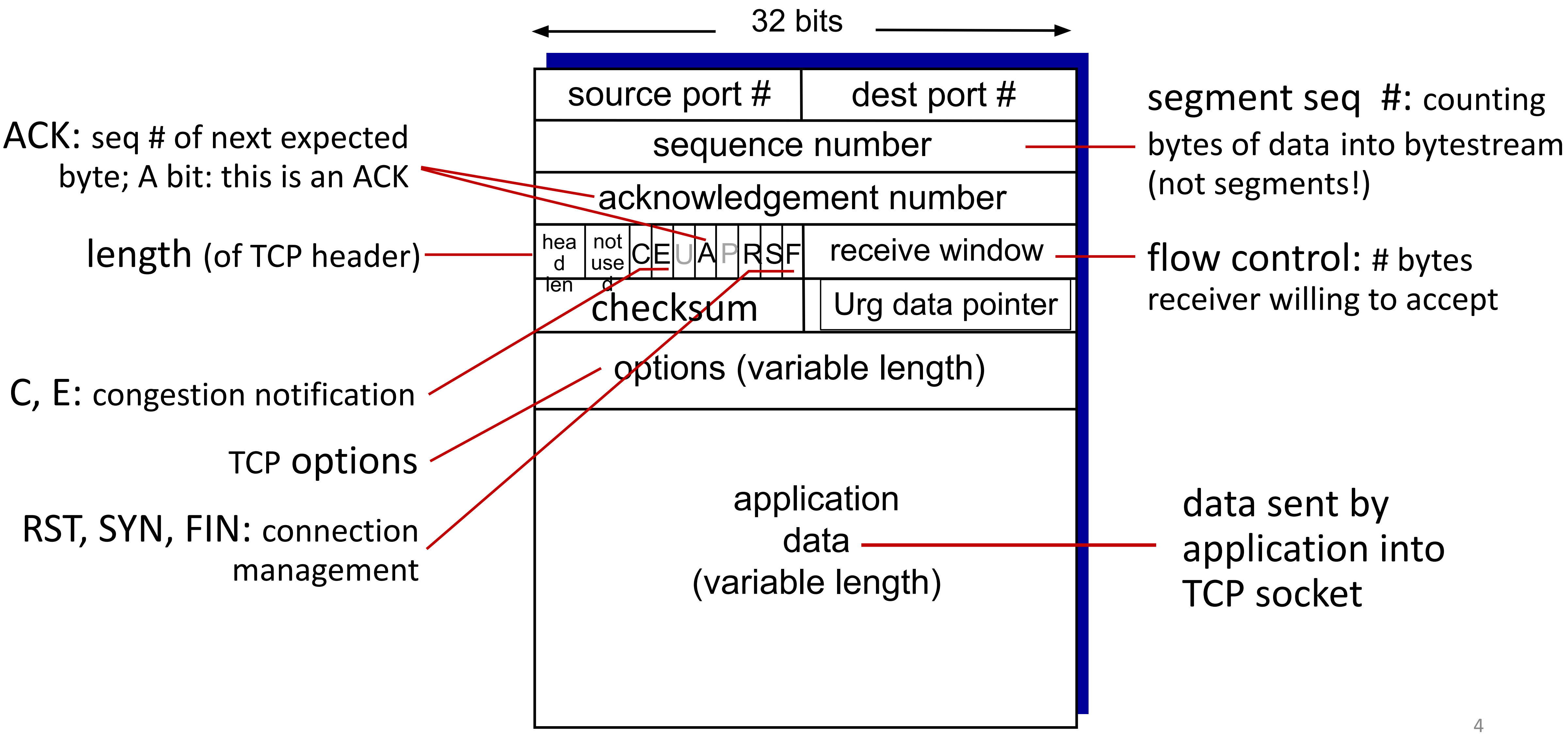
Dr. Arpita Thakre

Department of Electronics and Communication Engineering

TCP Segment Format - details



TCP Segment Format - details



TCP Segment Format - details

- The 32-bit **sequence number** field and the 32-bit **acknowledgment number** field are used by the TCP sender and receiver in implementing a reliable data transfer service.

- The **16-bit receive window** field is used for flow control.

- The **4-bit header length field** specifies the length of the TCP header in 32-bit words. The TCP header can be of variable length due to the TCP options field. (Typically, the options field is empty, so that the length of the typical TCP header is 20 bytes.)

TCP Segment Format - details

- The **optional and variable-length options** field is used when a sender and receiver negotiate the maximum segment size (MSS) or as a window scaling factor for use in high-speed networks.

- A **timestamping option** is also defined. (See RFC 854 and RFC 1323 for additional details)

TCP Segment Format - details

- The flag field contains 6 bits.
  - The **ACK bit** is used to indicate that the value carried in the acknowledgment field is valid; that is, the segment contains an acknowledgment for a segment that has been successfully received.
  - The **RST,SYN, and FIN** bits are used for connection setup and teardown.
  - The **CWR and ECE** bits are used in explicit congestion notification.
  - Setting the **PSH bit** indicates that the receiver should pass the data to the upper layer immediately.
  - The **URG bit** is used to indicate that there is data in this segment that the sending-side upper-layer entity has marked as “urgent.” The location of the last byte of this urgent data is indicated by the 16-bit urgent data pointer field. TCP must inform the receiving-side upper layer entity when urgent data exists and pass it a pointer to the end of the urgent data

TCP SEQUENCE Numbers and ACKs

Sequence numbers:

- A message may be divided based on the MSS into TCP segments.
- The 1st byte in each segment is assigned a unique sequence number before being transmitted.
- For the 1st segment a random 32-bit number is assigned.
- The received segments are reordered using these numbers

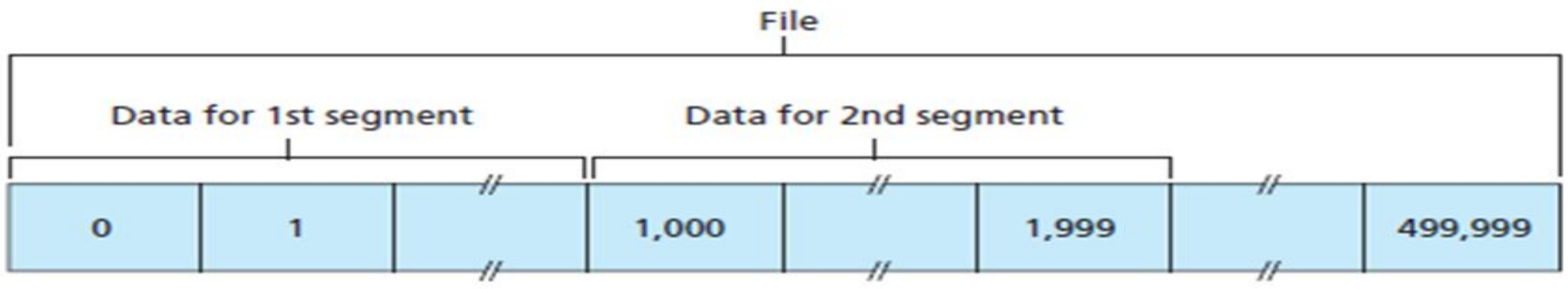


Figure 3.30 ♦ Dividing file data into TCP segments

TCP SEQUENCE Numbers and ACKs

Acknowledgement numbers

- Used by receiver to inform the sender of missing segments.
- Receiver acknowledges each received segment.
- Receiver stores out-of-order segments.
- Replies with acknowledgement of the last correctly received in-sequence segment.
- Cumulative acknowledgements are accepted by the sender in case out-of-order acknowledgements are received

TCP SEQUENCE Numbers and ACKs

Suppose that Host A has received one segment from Host B containing bytes 0 through 535 and another segment containing bytes 900 through 1,000. For some reason Host A has not yet received bytes 536 through 899. In this example, Host A is still waiting for byte 536 (and beyond) in order to re-create B's data stream. Thus, A's next segment to B will contain 536 in the acknowledgment number field. Because TCP only acknowledges bytes up to the first missing byte in the stream, TCP is said to provide **cumulative acknowledgments**

TCP SEQUENCE Numbers and ACKs

Host A received the third segment (bytes 900 through 1,000) before receiving the second segment (bytes 536 through 899). Thus, the third segment arrived out of order. **The subtle issue is**:

What does a host do when it receives out-of-order segments in a TCP connection?

Depends upon the people programming a TCP implementation.

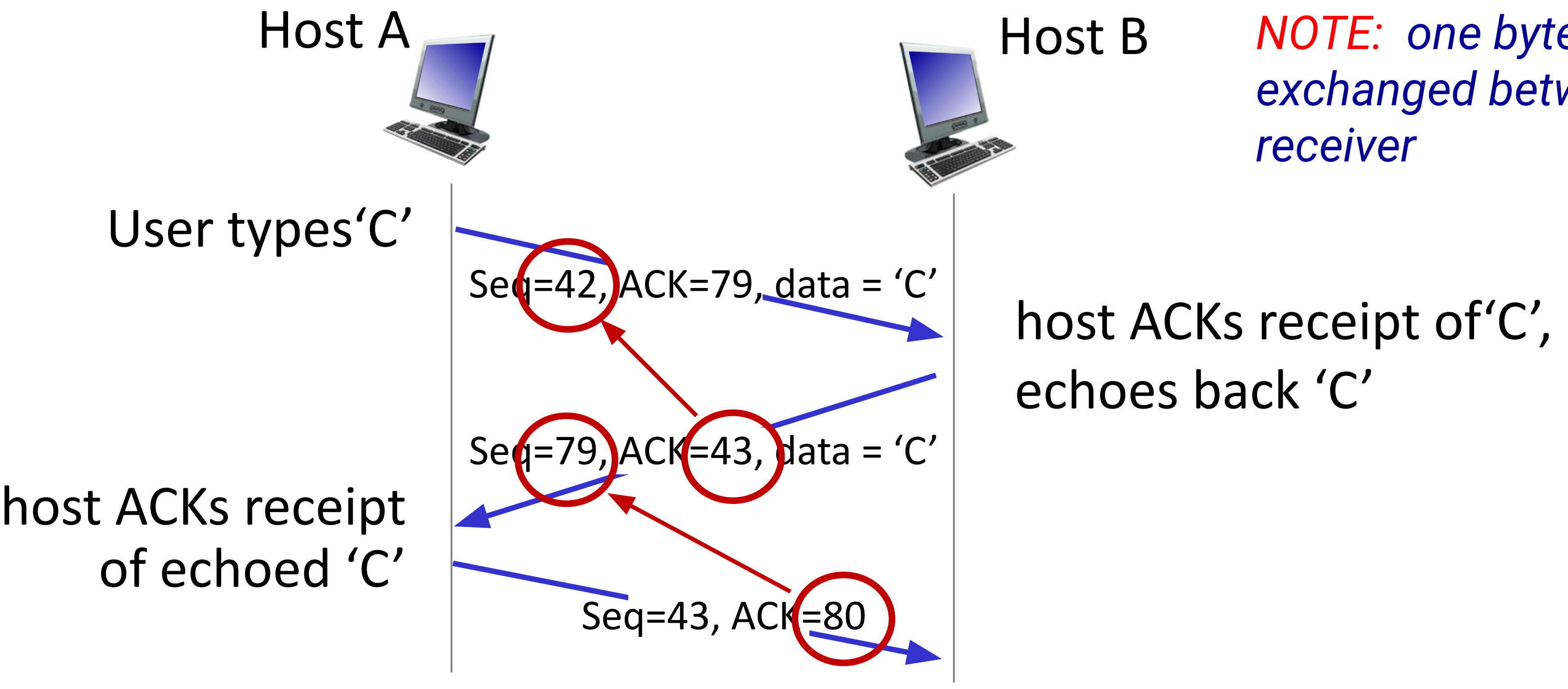
Two choices:

(1) the receiver immediately discards out-of-order segments (which, can simplify receiver design), or

(2) the receiver keeps the out-of-order bytes and waits for the missing bytes to fill in the gaps.

(which is more efficient in terms of network bandwidth, and is the approach taken in practice.

TCP SEQUENCE Numbers and ACKs



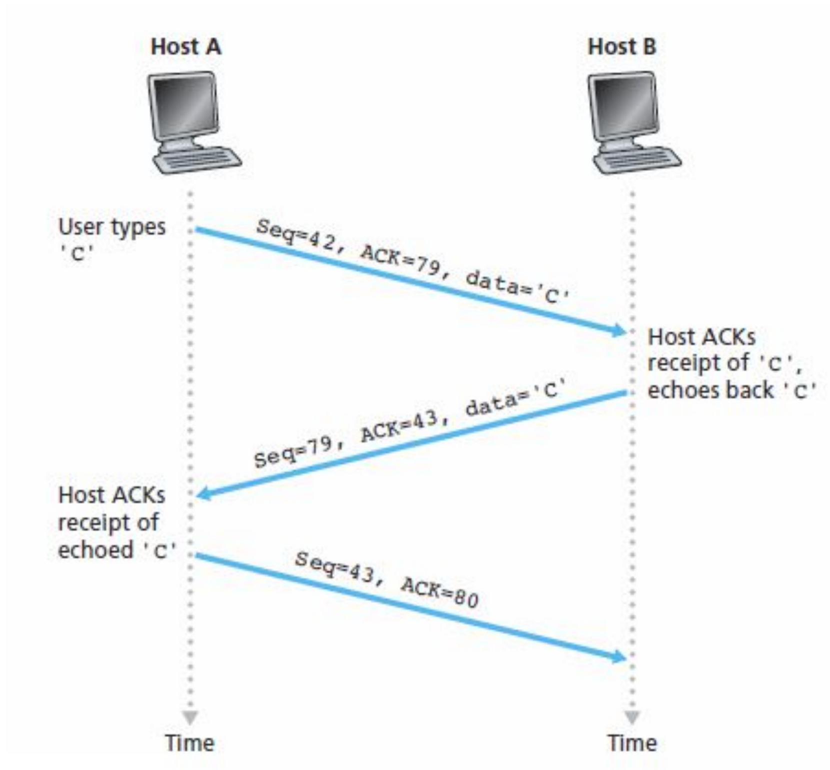
NOTE: one byte segment is being exchanged between sender and receiver

simple telnet scenario

TCP SEQUENCE Numbers and ACKs

The acknowledgment for client-to-server data is carried in a segment carrying server-to-client data; this acknowledgment is said to be **piggybacked** on the server-to-client data segment.

The segment has 80 in the acknowledgment number field because the client has received the stream of bytes up through byte sequence number 79 and it is now waiting for bytes 80 onward. Because TCP has a sequence number field, the segment needs to have some sequence number. This segment has an empty data field (that is, the acknowledgment is not being piggybacked with any client-to-server data).



Sequence and acknowledgment numbers for a simple Telnet application over TCP

THANK YOU

Dr. Arpita Thakre

Department of Electronics and Communication Engineering