TCP

By

Dr M. Senthilkumar
Assistant Professor
Department of Computer Science
Government Arts and Science College, Avinashi - 641654

TCP – Functions

- Reliable
- Connection-Oriented
- Stream Oriented
- Full Duplex
- End – to – End Communication

TCP – Functions

- Service Point Addressing
  - Ports Addressing
  - Socket Addressing
- Connection Establishment
  - Virtual Connection
  - Handshake/ Agreement
- Connection Release
  - Terminate/ Close Connection

TCP – Functions

- Segmentation
  - Breaks Message into Segments

- End-to-End Error Control
  - IP provides Unreliable Service

- End-to-End Flow Control
  - Avoid Buffer overflow

- Multiplexing and Demultiplexing sessions

TCP – Reliable Service

- Transmits Segment

- Dynamically Calculates Waiting Time for ACK
  - RTT = α * old RTT + (1- α)* new RTT
  - α usually 90%
  - RTT for LANs in Milli Seconds
  - RTT for WANs in Seconds
  - Most common, Retransmission time = 2* RTT

TCP – Reliable Service – Timer Mechanism

- Starts a Timer with RTT
- Waits for ACK
- If ACK not received within RTT, Retransmit Segment and wait for a Longer RTT for ACK
- After a number of Retransmissions, it will give up
- Piggy-backed Acknowledgments or sent as separate

TCP – Error Control - Sequence Numbers

- Sequence Number is attached with every byte
- Seq # is used to Detect Loss of Segment
- Seq # is used to Reordering Segments
- Seq # is used to Remove Duplicate Segments
- Seq # and ACK # refer to byte number

TCP – Error Control - Sequence Numbers

- Seq # for each Segment is # of the first byte in Segment
-
- ACK # denotes # of the next byte expected

- ACK # may be for cumulative/ bulky

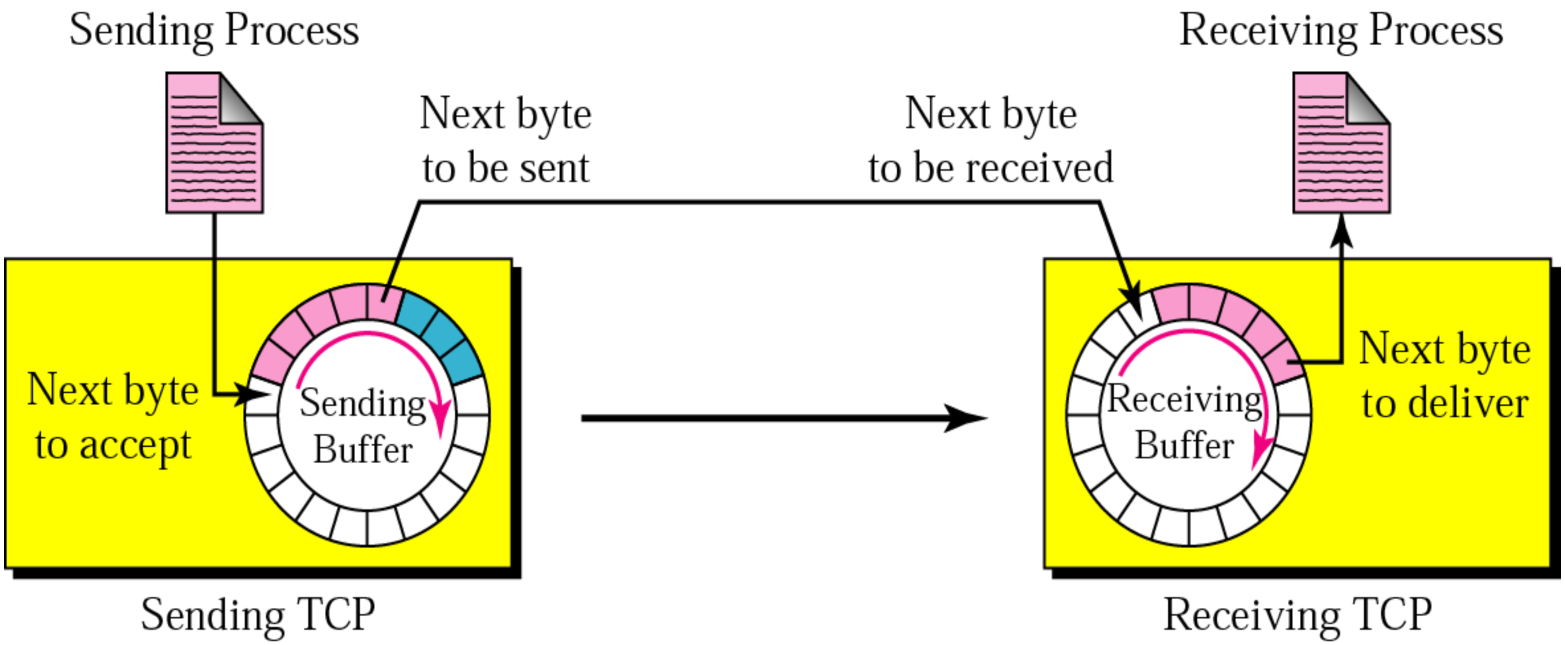
- ACK # 5643 refers to:

  - Receipt of all bytes from beginning up to 5642
  - Without any error

TCP – Flow Control - Buffering Mechanism

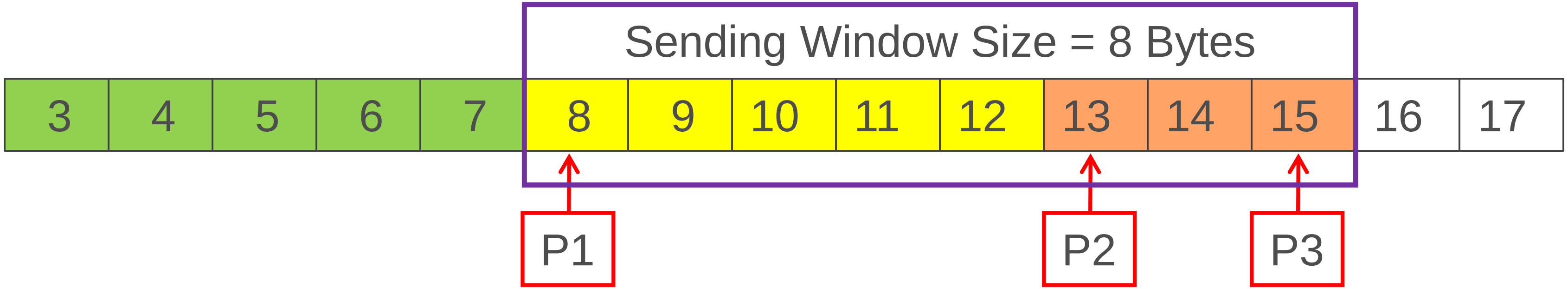
- Sending buffer and Receiving buffer
- Sender and Receiver may not Produce and Consume data at Same Speed
- 2 buffers for each direction
  (Sending and Receiving buffer)

TCP – Flow Control - Buffering Mechanism



TCP – Flow Control-Sliding Window Mechanism

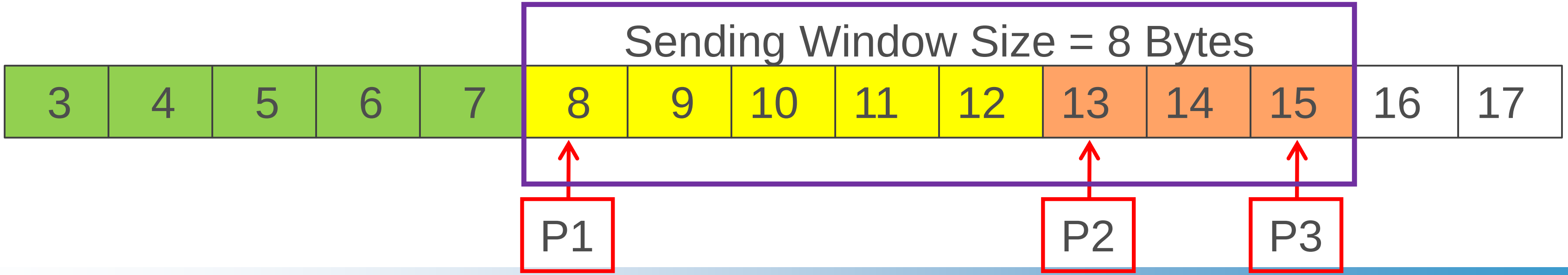
- ✓ Sender maintains 3 Pointers for each connection
- ✓ Pointer 1: Bytes sent and acknowledged
- ✓ Pointer 2: Bytes sent, but not yet acknowledged
- ✓ Pointer 3: Bytes that yet be sent
- ✓ Sender window includes Bytes sent but not acknowledged



TCP – Flow Control-Sliding Window Mechanism

- ✓ Sender's window is Slide on Buffer
  Without a change in Receiver's Advertised Window

- ✓ Expanding Sender's Window
  Receiving process consumes data faster than it receives, then Receiver Window Size increases

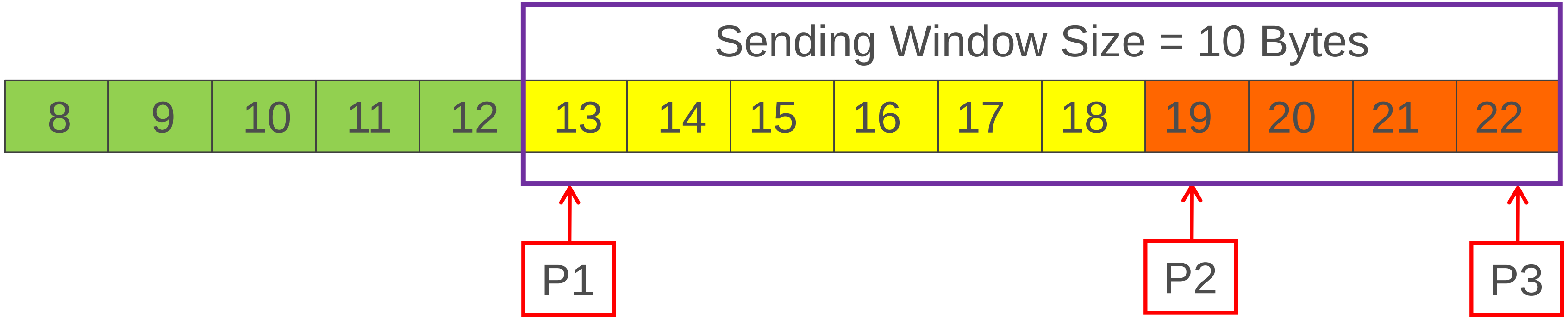


TCP – Flow Control-Sliding Window Mechanism

✓ Expanding Sender’s Window

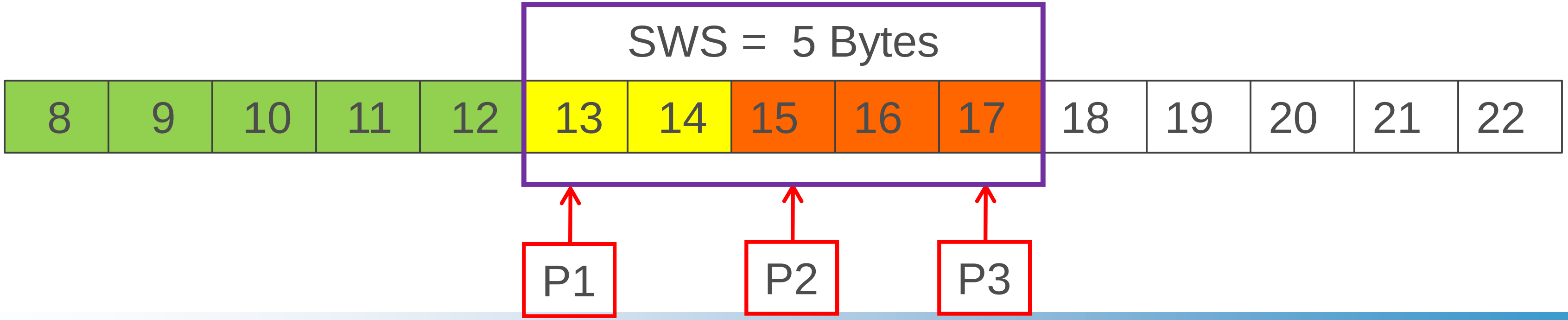
Receiving process consumes data faster than it receives, then Receiver Window Size increases

(If ACK = 13, WS = 10)



TCP – Flow Control-Sliding Window Mechanism

- ✓ Shrinking Sender's Window
  Receiving process consumes data more slowly than it receives, the Receiver Window Size reduces
  (If ACK = 13, WS = 5)
- ✓ Closing sender window
  Receiver advertises a Window of Zero



References

- ✓ Books: Data communication and Networking, Behrouz A Forouzan, Fourth edition
- ✓ Computer Networks, Andrew S. Tanenbaum, 4th edition, PHI
- ✓ Data Communication and Networks, Achyut Godbole, 2007, TMH.
- ✓ Computer Networks: Protocols, Standards, and Interfaces, Uyless Black, 2nd ed, PHI
- ✓ Various relevant websites

Thank You