

Flow Control – Sliding Window Protocol (1/4)

1

TCP regulates the flow over a connection with the help of the **sliding window protocol**

- Sliding window protocol works with adaptive, load-independent **windows**, whereby acknowledgement and window assignment are decoupled from each other

Example (1/3): Message with a length of 2,500 bytes is transmitted from A to B via a TCP connection with a window size of 1,500 bytes ($F = 1,500$)

- A sends first 1,000 bytes
- B takes these, writes them into input buffer, adjusts window size to $1,500 - 1,000 = 500$ ($F=500$), acknowledges receipt of the first 1,000 bytes ($ACK=1,000$), and window size $F=500$

Flow Control – Sliding Window Protocol (2/4)

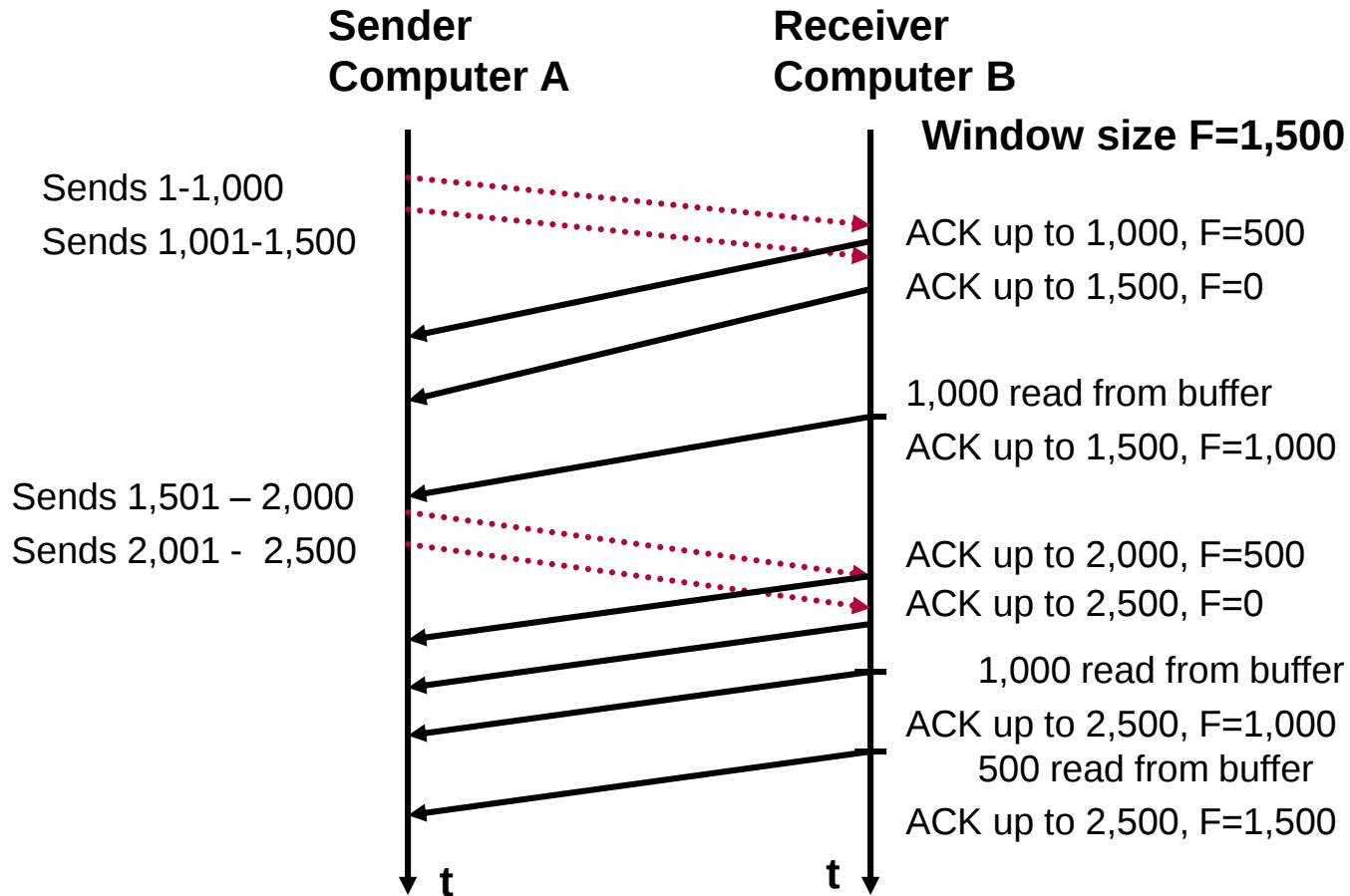
2

Example (2/3): Message with a length of 2,500 bytes is transmitted with a window size of 1,500 bytes from A to B via TCP connection

- ...
- A transmits the next 500 bytes according to the allowed window size
- B takes these, sets window size to 0 and acknowledges receipt and new window size
- A must now wait until the application at B has finished reading the 1,000 bytes from the output buffer, resets the window size to 1,000 bytes, and sends the corresponding message (ACK=1,500, F=1,000)
- A can now send the rest of the message (1,000 bytes)

Flow Control – Sliding Window Protocol (3/4)

3



Silly Window syndrome

- Without additional measures, window size would adjust to 0 in case of very fast connections - resulting in an inefficient use of the transmission path
- Receivers therefore give feedback on the available window size only first when at least 50% of the maximum window size is free again
- Sender ensures that the window size available is fully used