TCP: Sliding Window Implementation

Kameswari Chebrolu

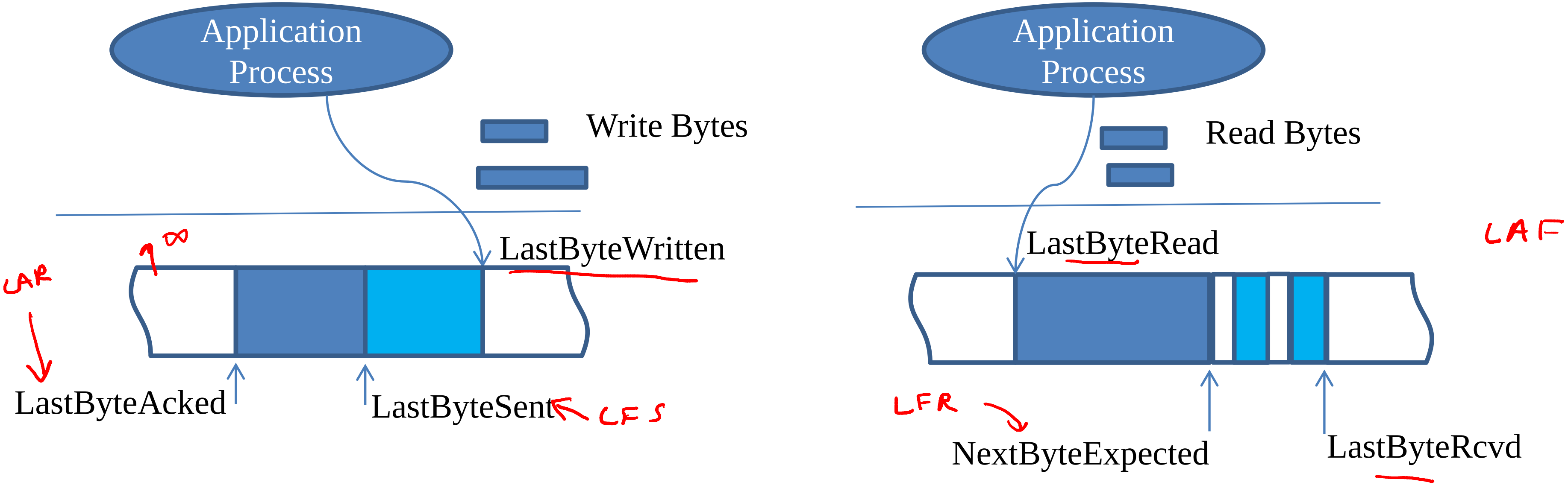
Recommendation: Refresh sliding window concept

Recap

- TCP services and header format
- TCP connection management
- TCP congestion control
- In this video
  - Flow Control
  - Some miscellaneous aspects

Flow Control: TCP Buffers

- Connection oriented "byte-stream" protocol



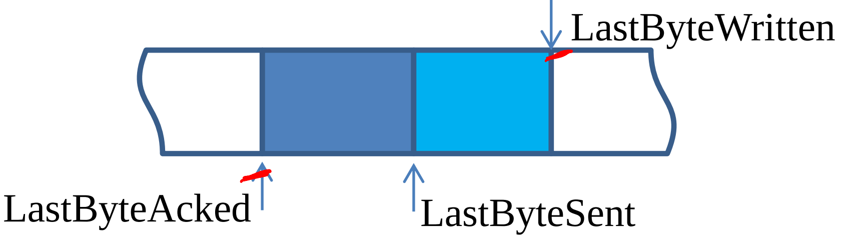
LastByteAcked <= LastByteSent <= LastByteWritten

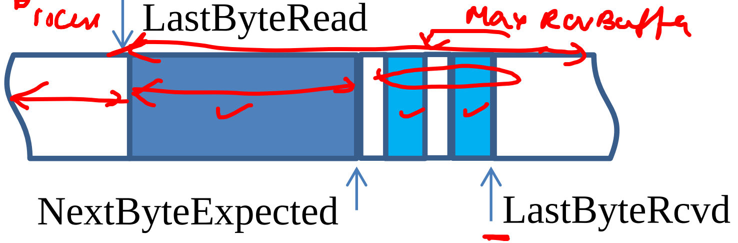
LastByteRead < NextByteExpected <= LastByteRcvd +1

↑ SWS → BDP

RWS = SWS

Both buffers are of finite size: MaxSendBuffer and MaxRcvBuffer



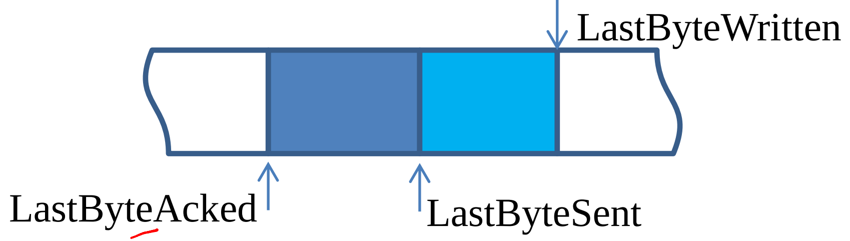


LastByteWritten – LastByteAcked <= MaxSendBuffer

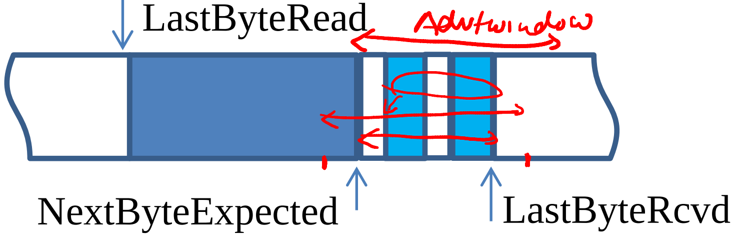
LastByteRcvd-LastByteRead <= MaxRcvBuffer

- Two cases that can cause receive buffer overflow
  - MaxRcvBuffer < BDP ↗ receive resource constrained
  - MaxRcvBuffer > BDP but application is slow to read
- controls Receiver Side: To avoid overflowing of the receive buffer: Advertised window
  - AdvertisedWindow = MaxRcvBuffer – [(NextByteExpected-1) – LastByteRead]

    (free space available at the receive buffer)



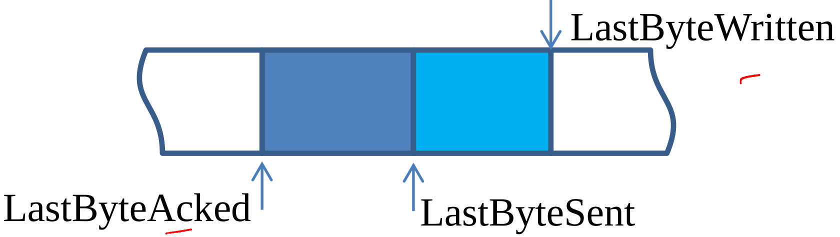
LastByteWritten – LastByteAcked <= MaxSendBuffer



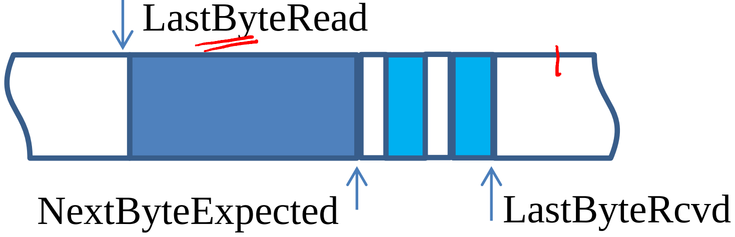
LastByteRcvd-LastByteRead <= MaxRcvBuffer

- Sender Side: Has to avoid overflowing receiver buffer and also not congest the network
  - LastByteSent-LastByteAcked <= AdvertisedWindow
  - LastByteSent-LastByteAcked <= cwnd
  - MaxWindow = min (cwnd, AdvertisedWindow)
  - EffectiveWindow = MaxWindow – (LastByteSent – LastByteAcked)

  (how much data it can send; can send only if EffectiveWindow > 0)



LastByteWritten – LastByteAcked <= MaxSendBuffer



LastByteRcvd-LastByteRead <= MaxRcvBuffer

- Sending side also needs to ensure that

send(-)msg

  - LastByteWritten – LastByteAcked <= MaxSendBuffer
  - Blocks the sending process and does not allow it to write

MaxRcvBufr > BDP

- Slow receiver stops the sender
  - First receiver buffer fills up →advertised window of 0 → sender can't send data → sending buffer fills up → TCP blocks sender

Zero Advertised Window

- How does the sender know when the advertised window is not zero?

  smart sender / dumb receiver

  - Receiver does not spontaneously send acks (dumb receiver)

Cum ACK → Advt window

- Solution: Sender persists in sending 1 byte segments periodically till it sees non-zero window