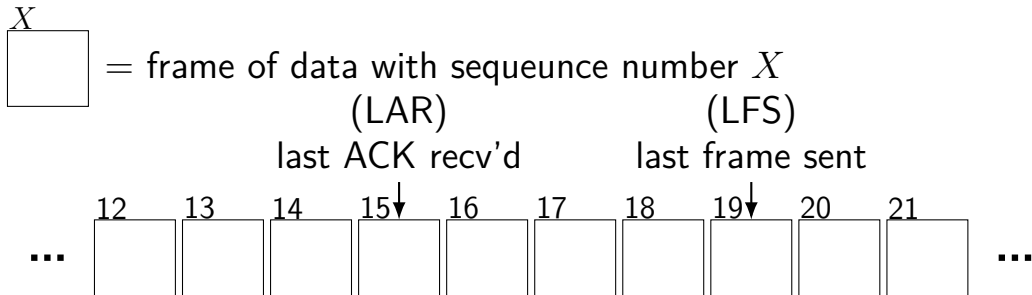
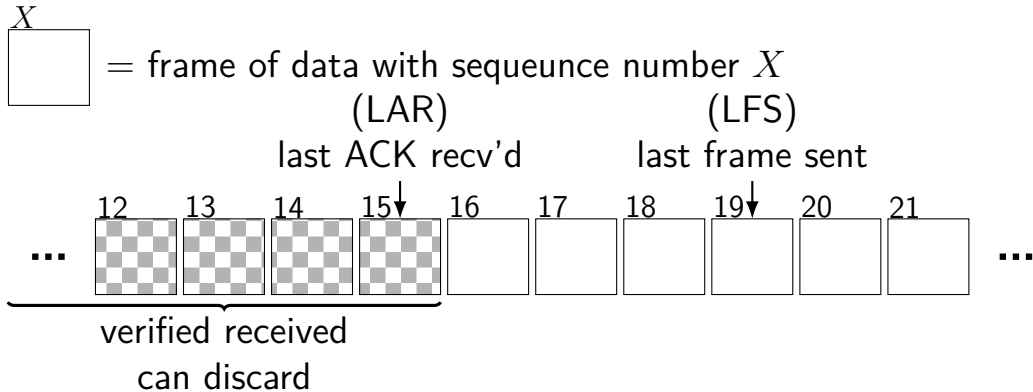


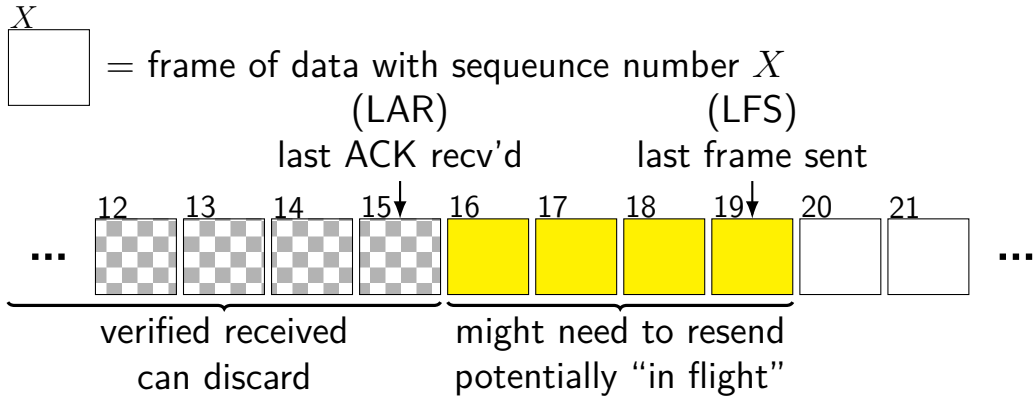
sender window tracking



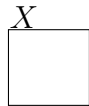
sender window tracking



sender window tracking



sender window tracking



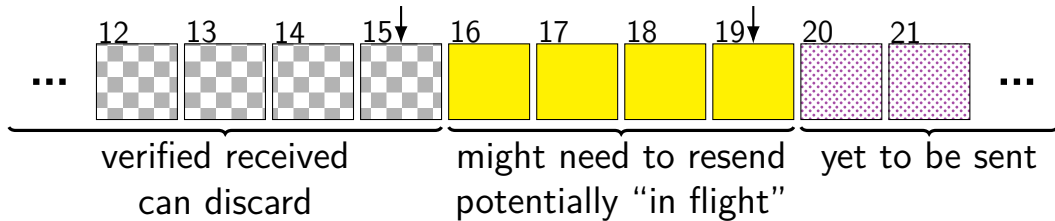
= frame of data with sequence number X

(LAR)

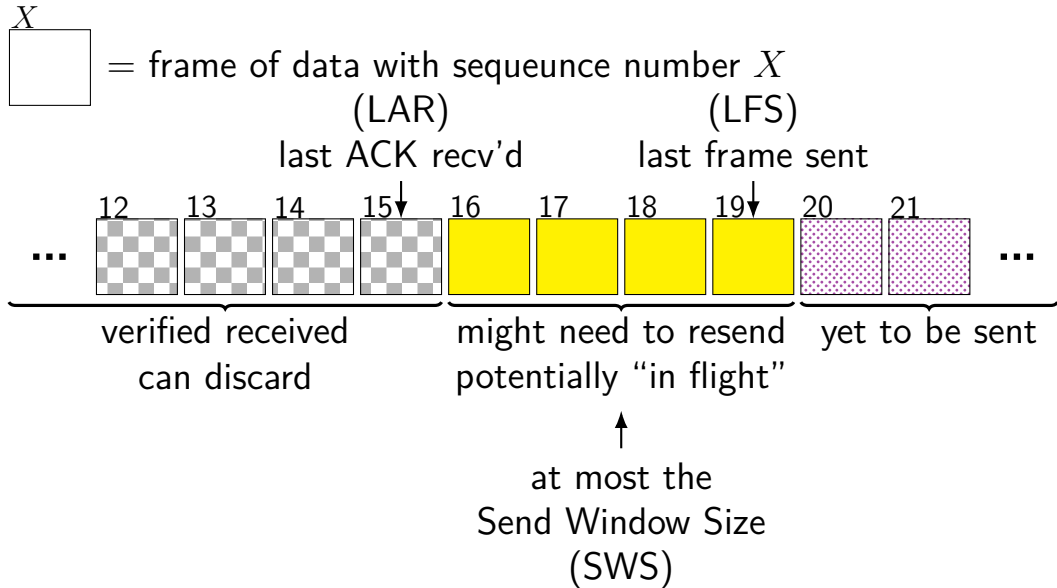
(LFS)

last ACK recv'd

last frame sent



sender window tracking



exercise 1: out-of-bounds ACK

last ACK recv'd (LAR)	10
last frame sent (LFS)	15
send window size (SWS)	5

what probably happened if we receive an ACK for...

9? 10? 13? 16?

- A. only possible if network reorders frames
- B. only possible from undetected frame corruption
- C. lost ACK for frame ≤ 10
- D. lost ACK for frame > 10
- E. lost frame 11
- F. resent frame from timeout

exercise 2: sender logic

last ACK recv'd (LAR)	10
last frame sent (LFS)	15
send window size (SWS)	5

In this case, there's a timeout that will trigger frame 13 to be resent. If still active, this timeout should be cancelled upon ...

- A. receiving ACK 12 B. receiving ACK 13
- C. receiving ACK 14 D. sending frame 16

exercise 3a: new data

last ACK recv'd (LAR)	4
last frame sent (LFS)	6
send window size (SWS)	5

if we compute a new frame of data with sequence number 7 to eventually send, we should

- A. send it now, advancing LFS
- B. wait until we get an ACK for 5 or 6 to send it
- C. wait until we get an ACK for 6 to send it
- D. wait until the frame with sequence number 6 is resent to send it D. some

exercise 3b: new data

last ACK recv'd (LAR)	4
last frame sent (LFS)	8
send window size (SWS)	4

if we compute a new frame of data with sequence number 9 to eventually send, we should

- A. send it now, advancing LFS
- B. wait until we get an ACK for 5 or 6 to send it
- C. wait until we get an ACK for 6 to send it
- D. decline to accept the data because we will never be able to send it
- E. something else