Variables to remember:

To distinguish between frames, we add a sequence number to each frame.

frame(0) is the first data message

frame(1) is the next data message

frame(2), frame(3), frame(4), etc.

(SWS = sender window size)

SWS outstanding messages in the channel

All frames in the range from frame(0) up to and including frame(i) must be acknowledged before the sender can send frame(i+SWS).

Sender

* Two important variables in the sender process:

1. Largest Frame Sent (**LFS**):

sequence numbers in 0 .. LFS have been sent by sender,

1. Largest Acknowledgement Received (**LAR**)

sender has received an acknowledgement for frame messages 0 .. LAR.

Always: LAR ≤ LFS, why?

* Therefore, for an arbitrary seq. no. i:

1. if i ≤ LAR, then   
    an acknowledgment for frame(i) has been received by sender
2. if LAR < i ≤ LFS, then  
    frame(i) has been sent but, its ack has not been received
3. if i > LFS, then  
    frame(i) has not been sent.

Receiver

Receiver process has two variables/constants:

1. Next Frame Expected (**NFE**)
   * frame(0) up to frame(NFE - 1) have been received by receiver.
2. Receive Window Size (**RWS**) : out-of-order packets buffer size (often = SWS)
   * Last Frame Acceptable(**LFA**) : LFA **=** NFE + RWS – 1

Note that frame(NFE) has not been received yet, since otherwise we would have incremented NFE.

When frame(NFE) is received, it is given to the application, and NFE is increased.

If frame(i), where i > NFE, is received, then store it only if i ≤ LFA.

Therefore, for an arbitrary seq no i:

1. if i < NFE, then frame(i) has been received
2. if i = NFE, then frame(i) has not been received
3. if NFE < i ≤ LFA, then frame(i) *may have*  been received

Slide 25 onward

Bounded Sequence Numbers