John doesn't know I received this ACK. So unless I ACK(ACK), he won't risk coming.

State Machines

Strict problem than what transport layer faces.
Reliable Pkt transmission.

Sleep → Read CNN → give pkt to Network layer → get packet → wait for pkt from app layer → give pkt to app layer → receive pkt → wake up alarm 6am → keep on not 8am

Tx → Rx → Tx → Rx

Packet was corrupt

Time

10110001010011101
- Ack corruption causes duplication at RX.
- NACK corruption is fine.

Add seq. # to packets.
Packet counter (seq # for each pkt)

Under bit errors (packet corruption)

1 bit seq # is enough.

Error Model: Bit error + pkt loss

Set timer/alarms

Retx after some timeout
Link Error: Bit Error + Loss + Delay
Packet sequence number needs to be increased.

Yes, there is delay, but delay \( \leq \text{upper bound} \). 

\[ \text{RTT} = \text{Round Trip Time} \]

\[ \log\left(\frac{T}{2 \cdot \text{RTT}}\right) \]

For simplicity, replace NACKs with duplicate ACKs.
NACK-free protocol.

\[ N > \log \left( \frac{C}{2 \times \text{RTT}} \right) \]
Stop and Go

How bad is this?

Every 60 ms → Rx receives 1 millisecond data.
Stop & Go.
Transport Layer

Multiplexing/Demux

Reliable packet transport

Principles

Correct Behavior

Stop & Go

Performance

Pipelining

Real world

TCP

III. Tags

Tx transmits

Rx transmits Cumulative Acks

Rx does not have any packet buffer.

Tx sends a window of packets

Rx ACKs only a single packet

Rx buffers some out of order (000) packets, depending on its own window size.

GBN (Go Back N)

Selective ACK SACK

Congestion Control

Flow Control
1. Pelipalled

2. Go Back N (GBN)

- Transmits a window of packets
  (Window size = W = 4)

- Does not have a buffer
- ACKs the last in order packet received correctly

\[ W = 3 \]

\[ \text{# of un-ACKed pkts Tx can send to not waste any time} = \frac{\text{RTT}}{T.T} \]
Tx maintains a window = \langle base, tail \rangle

\[
\begin{array}{cccc}
\text{i} & \text{j} & \text{k} & \text{l} \\
\end{array}
\]

\[
A_p, \ p > i
\]

Yes, if ACKs get reordered.
GBN: Transmitter's window says which pkts can be sent with pending Accs.

Selectiv ACK

Tx window

Rx window says how many out of order packet can I hold while waiting for ACKs.
In selective ACK (SACK), the ack $A_i$ indicates that $Rx$ has received packet $P_i$.

(In GBN, $A_i$ indicates $Rx$ has received every in-order pkt till and including $P_i$)

Example (after class)
\[ w = 4 \text{ at } T_x \quad \text{and} \quad w = 4 \text{ at } R_x \]
T/F:  $R_x\_base < T_x\_base$  \[\text{False}\]

T/F:  $T_x\_tail > R_x\_base$  \[\text{Seq#\# }\]

T/F:  $R_x\_base$ can be $> T_x\_tail$  \[\text{True.}\]

T/F:  $R_x\_base$ can be $> T_x\_tail + 1$  \[\text{False.}\]

- Should $R_x$ ACK packets that are less than $R_x\_base$?
  
  Yes should be ack-ed

  $P_j$

  $i-W$  \[Rx\_base = i\]

  $w$ = window size (equal for both $Tx$ & $Rx$)

  $j < i$

  $j < (i-k)$
TCP (Transport Control Protocol) → Congestion Control → Adaptively vary/Control window size W

RTT goes up → Reduce W

TCP → Self clocking protocol