Round-trip Times for Performance Analysis.

send $t_{prop.} + t_{proc(dest)} + t_{frame.}$
receive. $+ t_{prop.(back)} + t_{proc(dest)} + t_{ack.}$ } $t_{rt.}$

Note $t_{frame} = \frac{n_{frame}(\#\text{ of bits})}{R}$ when R is transmission rate.

Bandwidth & delay.

suppose for "some reason" (processing, multiple unknown hops etc.) R.t. delay is rt
∴ to keep channel full.
$t_{rt} \cdot \text{rate}\left(\frac{\text{bits}}{\text{sec}}\right) =$ delay × bandwidth.

4 sec. rt. delay.
2000 bits/sec. transmsn. rate.
∴ 8000 bits is full pipeline.
∴ if packet is of size 1000 bits can have 8 packets in the pipeline.

However, transmission of a packet is only $\frac{1}{2}$ second. } rest of rt. time for unknown ack may be $\frac{1}{20}$ sec. } reasons.

Sequence Space

Number Sequence #'s.

3 bits. 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2

What is the maximum window the sender can have?

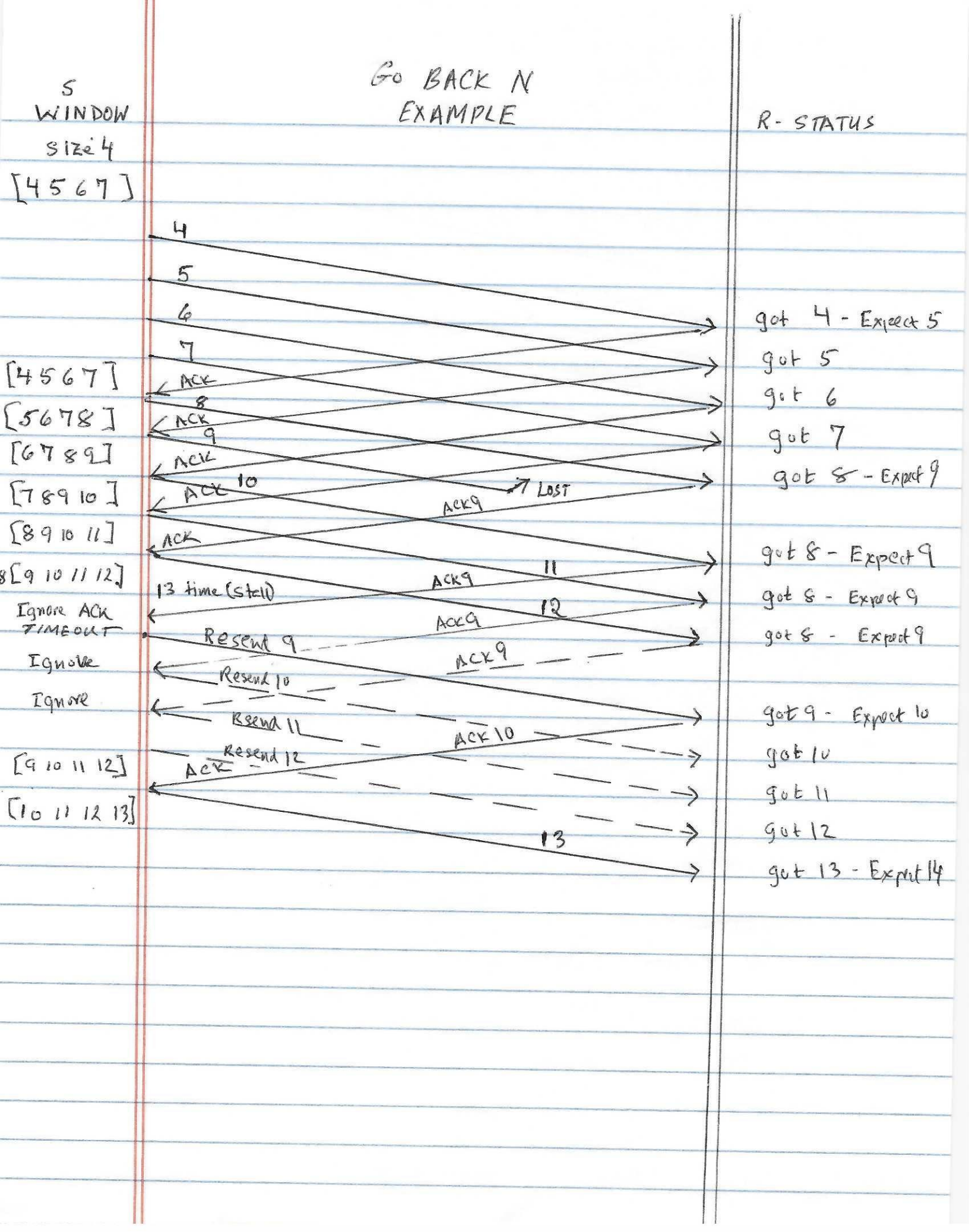
0 1 2 3 4 5 6 7

$S_{window} + R_{window} \leq N$

where N is size of sequence space

Example of selective repeat?

GO BACK N EXAMPLE



GO BACK N
EXAMPLE.

Notes

1. Received ACK 9 at time step 12.
This acks packet 8 and moves window to [9, 10, 11, 12]
Received ACK 9 again at time step 13.5
Why not resend 9? Note that this could cause problems a la Sorcerer's Apprentice since packet 9 might just have been delayed and don't want to send too early - wait for timeout

2. Note: receiver does not ack receiving packet 10 since it is expecting 9. Thus 10 must also be resent.

3. Note that sender is "stalled" from timestep 13, but needs to wait for timeout.
5 time periods were "wasted" because of the packet loss of 9, until sender could start sending normally again

4. Go Back N is not really very good under noisy channel conditions.

5. Note that choice of window (4) and timeout reflect delay-bandwidth product.

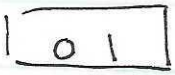
Examples

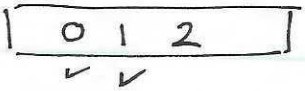
Window problems.

4 bit sequence # 0, 1, 2, 3.

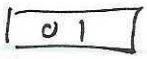
Window size of receive 3

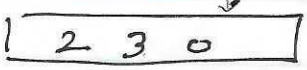
Sender size is 2.





receive gets both acks. both acks lost



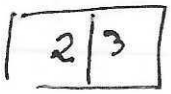


using a larger window means accepting out of order packets

receiver says can receive 2, 3 or next 0.

sender resend a 0, 1 again

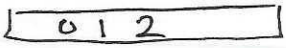
receive accepts as new ack for next 0. Thus wrong. as cannot distinguish new frame



suppose both acks arrive sender would eventually send new 0.

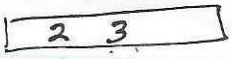
window size of sender is 3

receive size is 2.

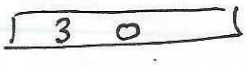


receive gets 0 1 ack's lost

sender resends 0, 1, 2 0, 1 lost.



gets 2 correct



sender retransmit 0

receiver will accept as new wrong

Suppose sender would have received all acks.

sender then sends 3 & new 0

receive get 0 with 3 lost will accept new as correct new

Timeouts
Go-back N.

(1)

Timer for a packet sent should be.
$t_{tout} \geq t_{rt}$. timer value ≥ round-trip time.

Often set to 1.5 – 2 times estimated rt. time.

(2) How to handle timer(s).

1 packet case.

@ send a packet; set timer.
on ack, [send next packet
set timer
on timeout, set timer; send next packet.

window

5	6	7	8

↑

set timer send 5 ← timeout. resend 5, 6, 7, 8
set timer send 6 go back to 5 & resend
set timer send 7 entire rest of window

---

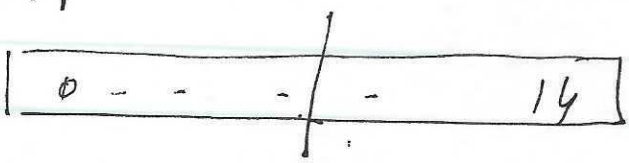
set timer send 5 (get. ack, move window)
set timer send 6
set timer send 7

---

single timer? on timeout, ~~ack~~. resend "window" go back N
on ack?; move window, reset timer
for next entry in window. ... Need to adjust

Handling Window

① Suppose rt. time is equivalent to 10 packets being sent.
Then window should be at least 10
Suppose window i set to 15

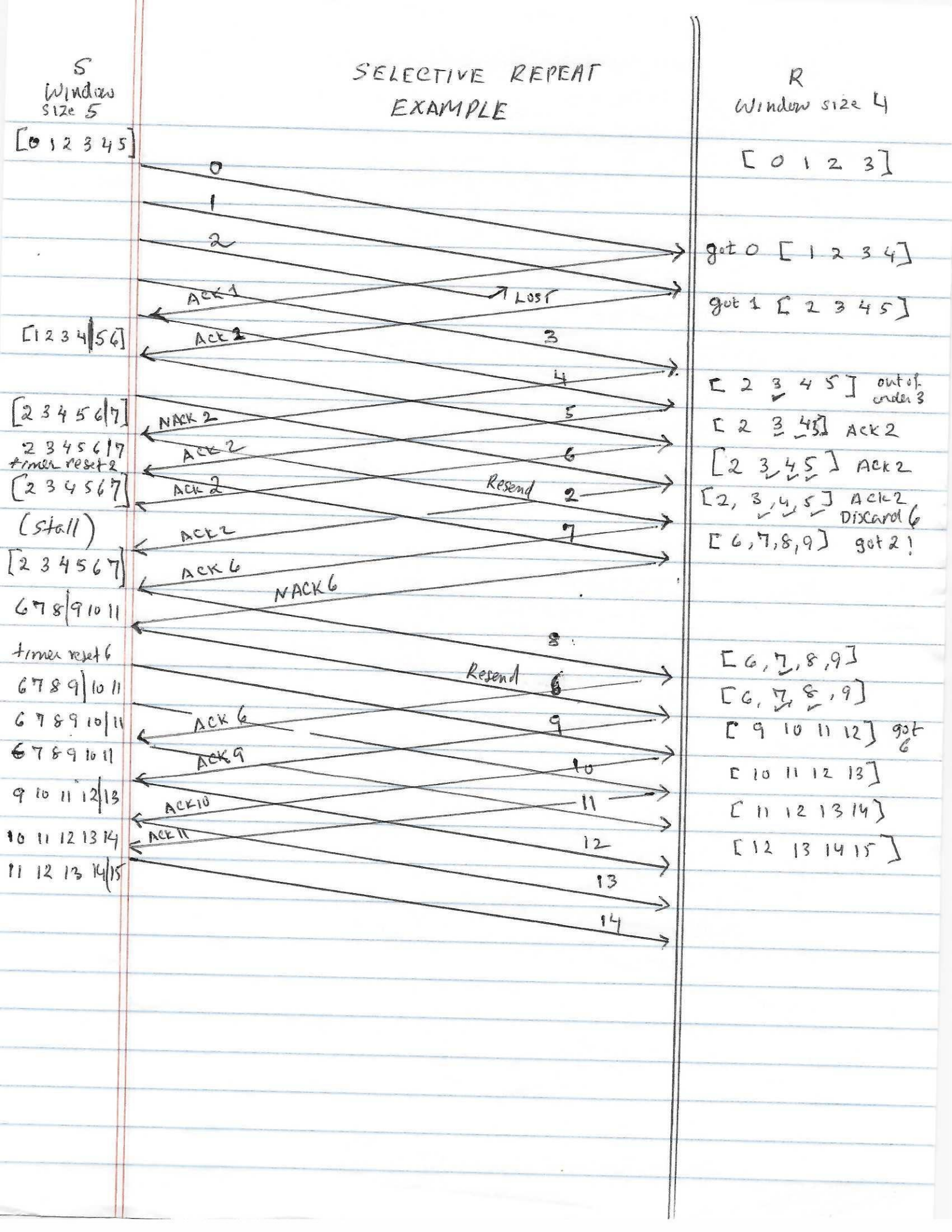


window can be larger.
What is the problem of a very large window?

Swamping receiver possibly.
Congestion in the network. etc.

—>.

SELECTIVE REPEAT EXAMPLE



SELECTIVE REPEAT EXAMPLE

Notes

1. First out-of order packet of a specific expected packet is nacked.
2. ACK's are always cumulative and ack the lowest packet in the window that is expected
3. Receiver processing is more complex
4. Timer's should be used used for ~~exact~~ each packet in sender window individually.
5. 4 time slots wasted.
6. Supposed to be better for noisy channels