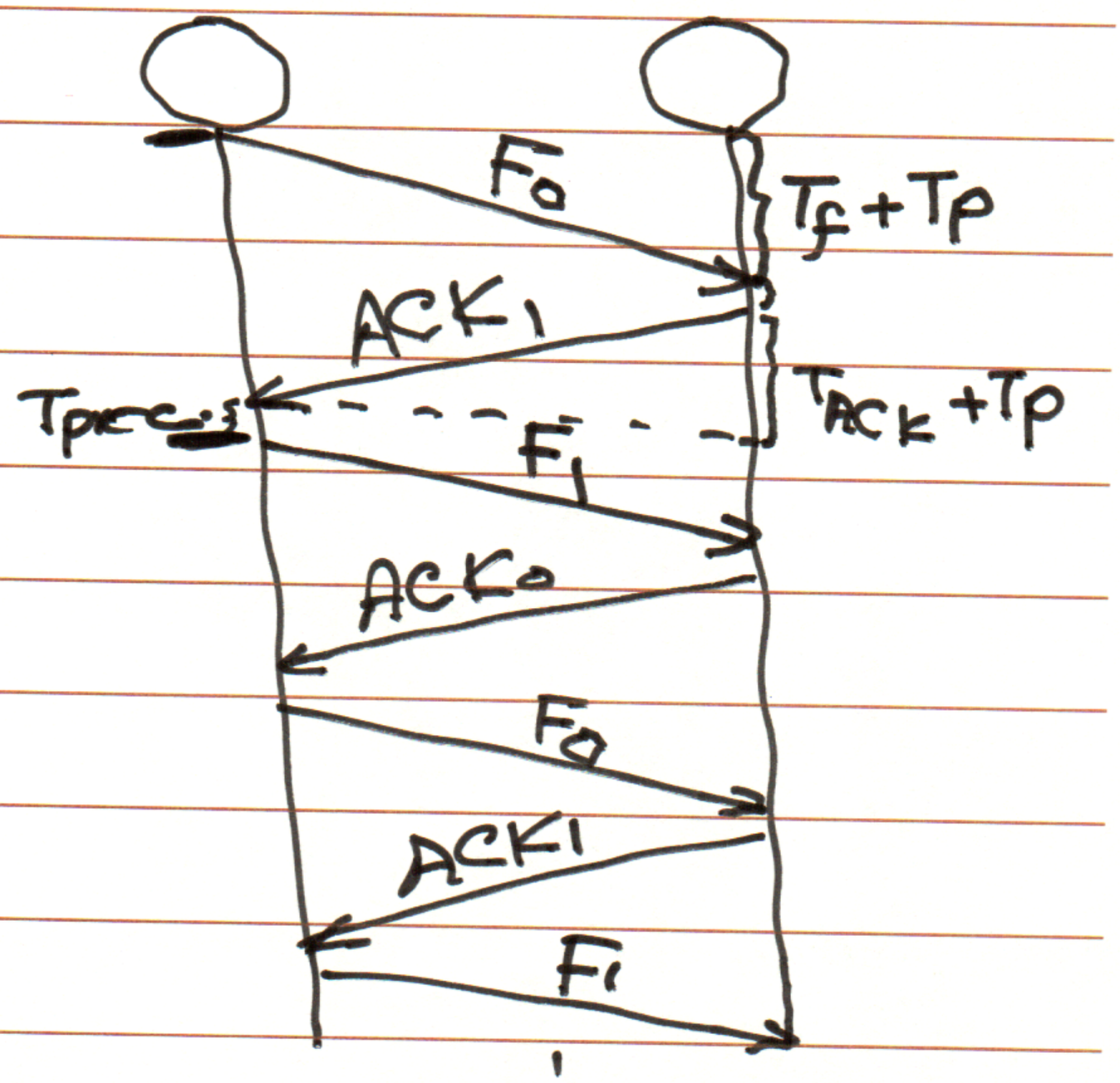
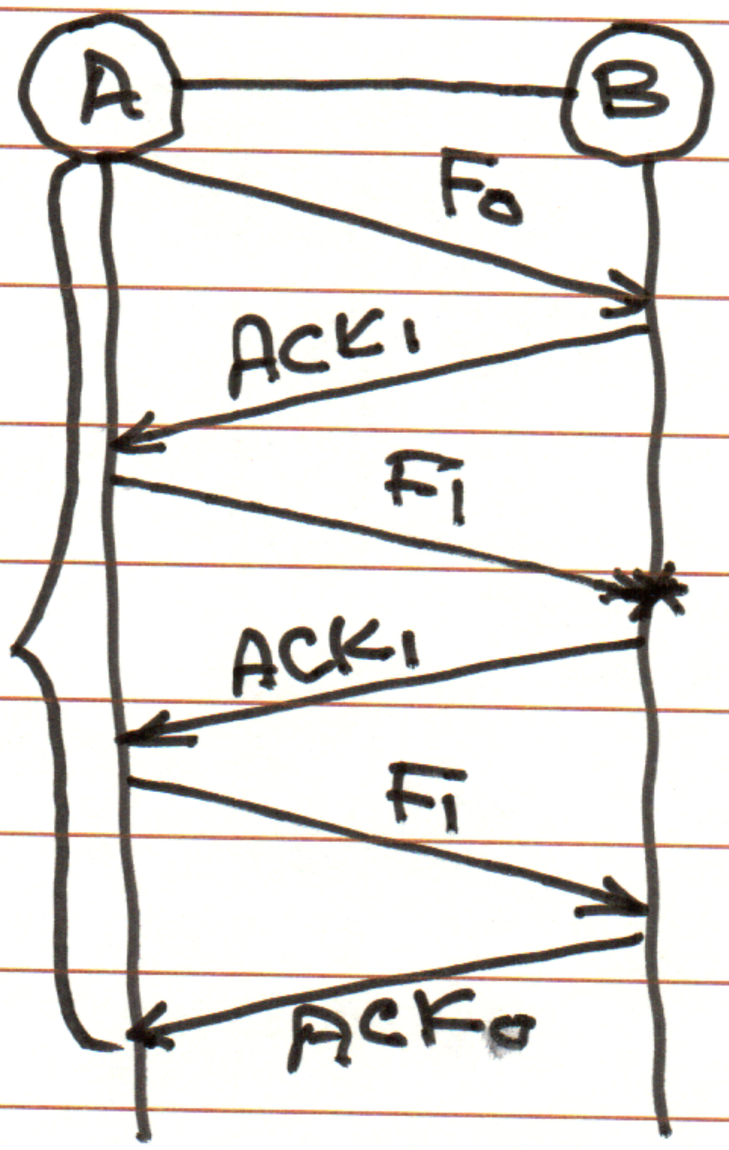


$$\text{Throughput} = \frac{1 \text{ frame}}{T_f + T_p + T_{proc} + T_{ACK} + T_p + T_{proc}}$$

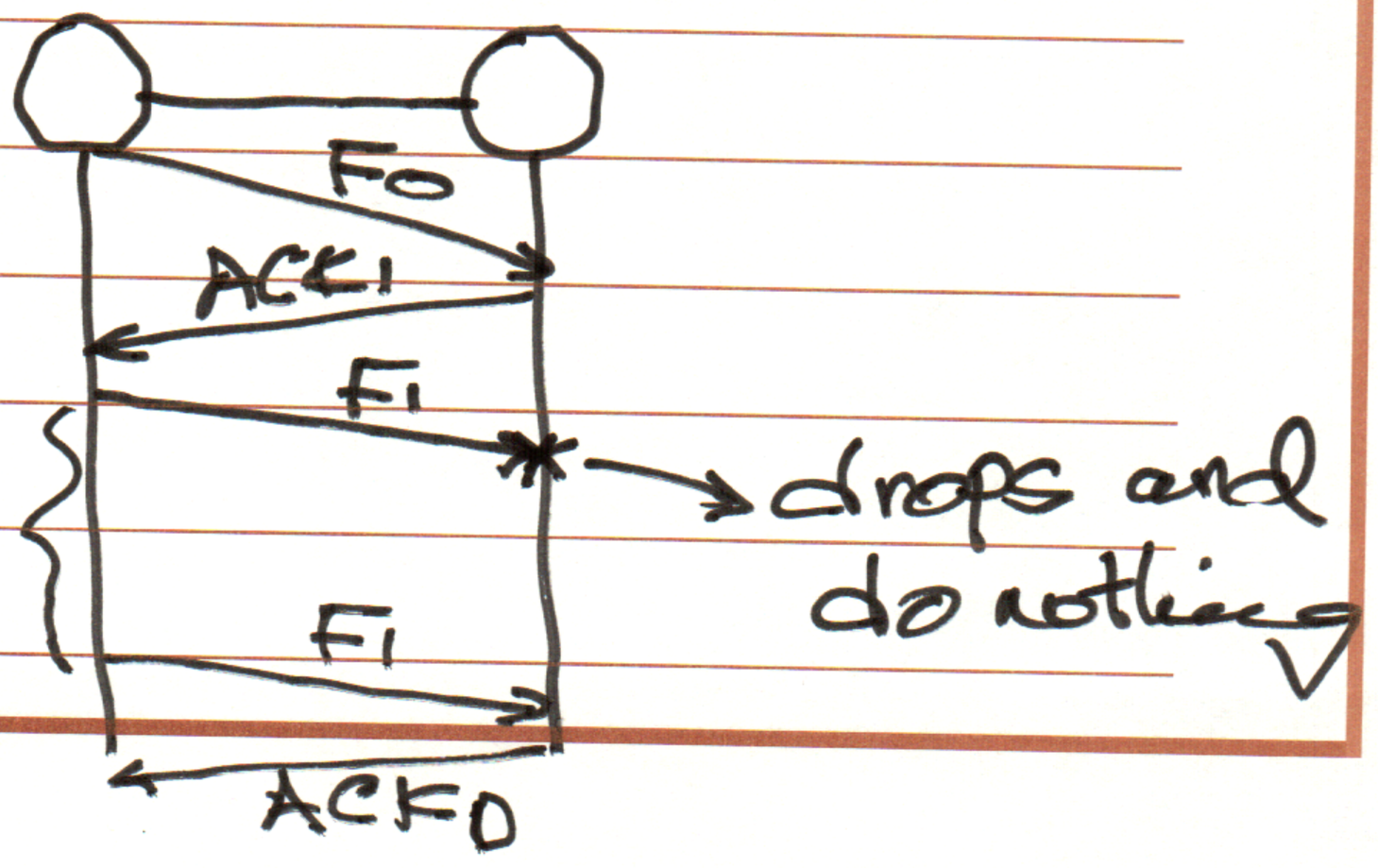
$$= \frac{1 \text{ frame}}{T_f + 2T_p} \quad \text{bps}$$

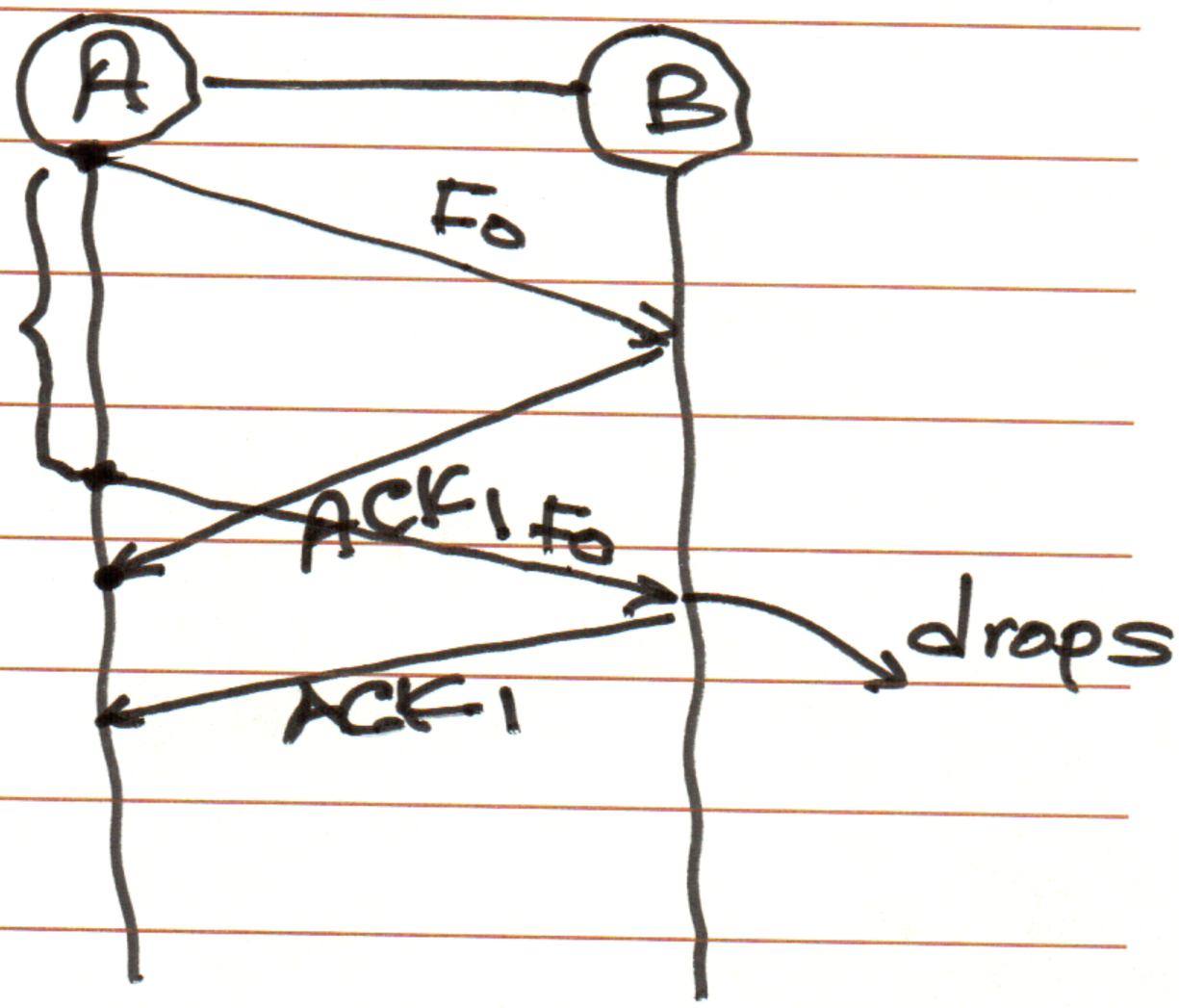
$$\text{Link utilization (\%)} = \frac{T_f}{T_f + 2T_p}$$

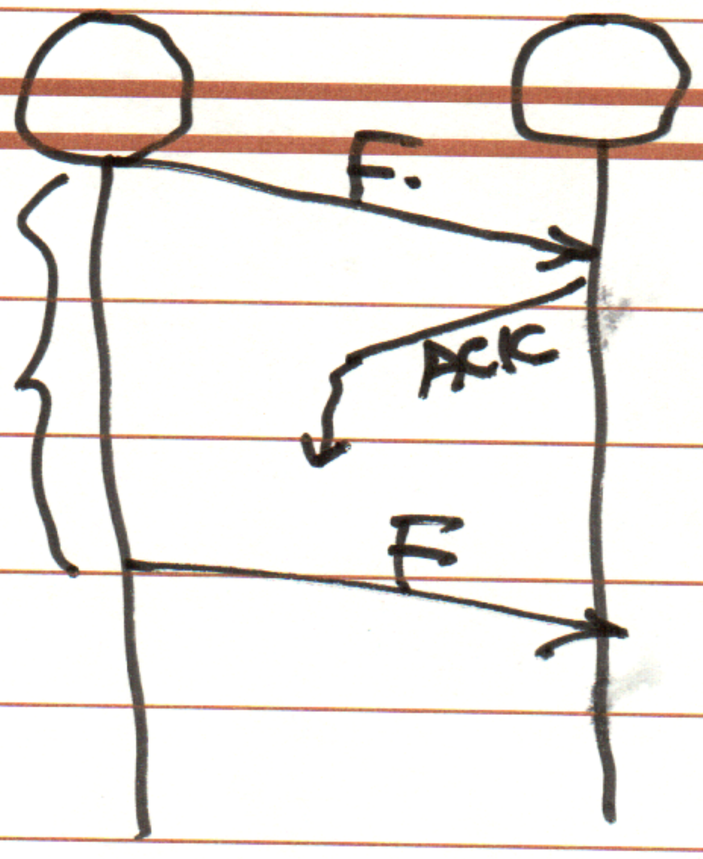


$$\text{Throughput} = \frac{2 \text{ frames}}{3T_f + 6T_p}$$

$$\text{Link utilization} = \frac{2T_f}{3T_f + 6T_p}$$




A
B
F0
ACK1 F0
drops
ACK1


F.
ACK
F

Sliding Window ARQ features

(1) Here the sender can transmit more than one frame, i.e. he doesn't have to wait for Acks. ⇒ "pipelining"

what determines the max. # of frames he can send?

BW x Delay product

Sequence bits $\triangleq m$

ex: $m=3$

0, 1, 2, ..., 7, 0, 1

Each side of the link maintains two windows

SWS = Sender Window Size

= max # of unacknowledged frames the sender can send.

RWS = Receiver Window Size

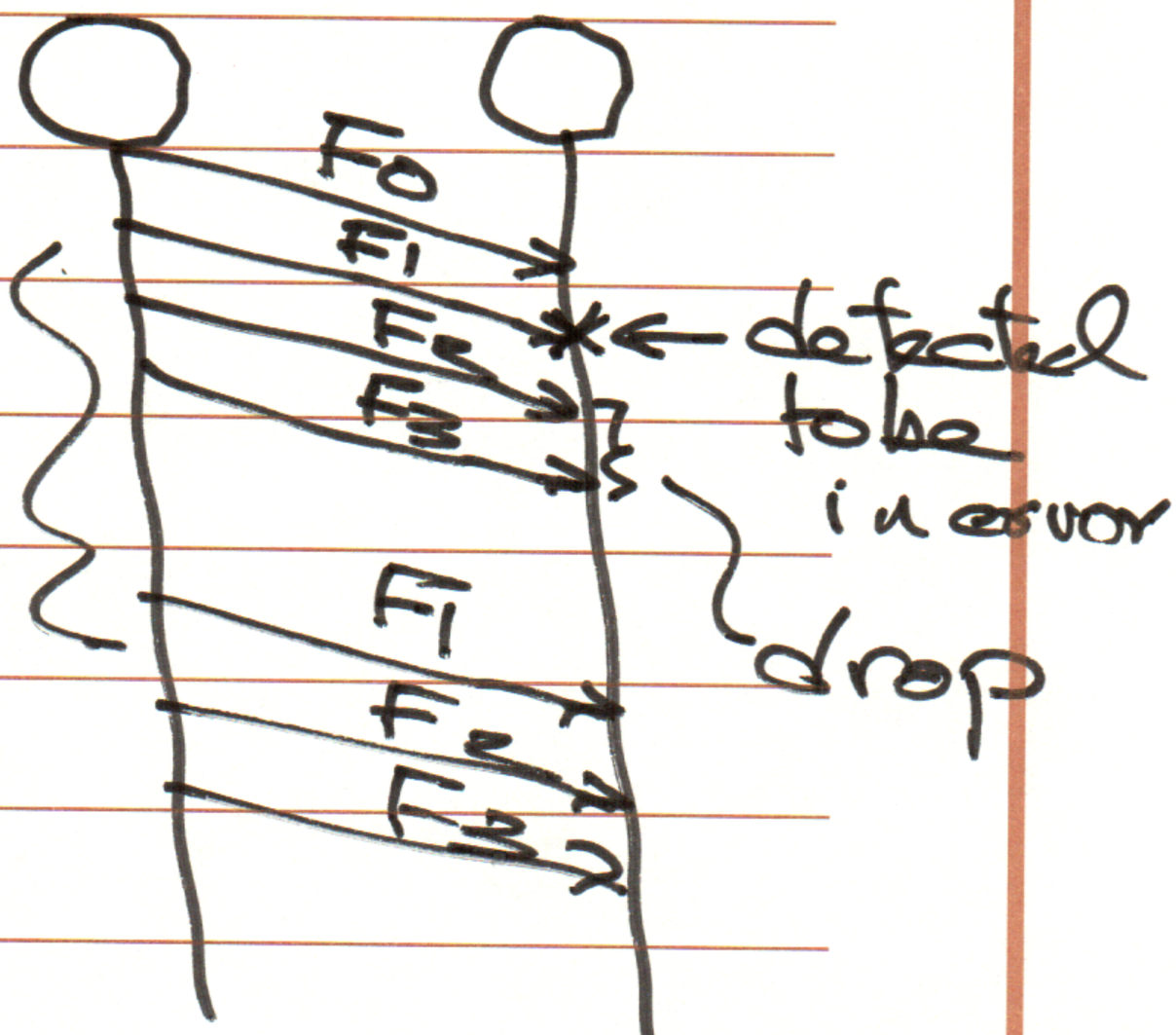
= max # of frames the receiver is willing to accept in any order.

In Go_back_N ARQ

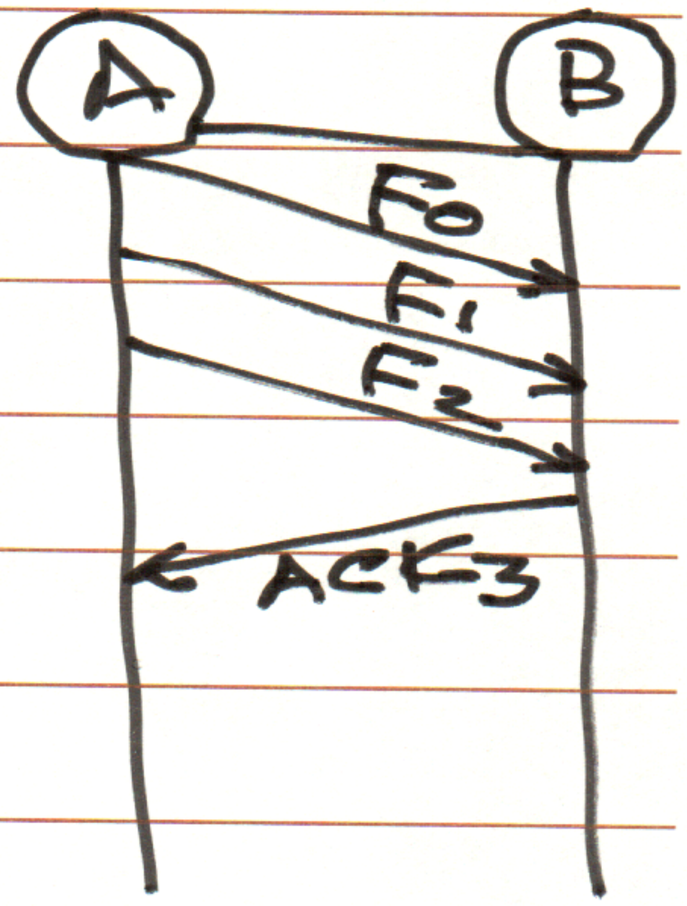
The receiver is not willing to accept any out_of order frames (Just like SkW ARQ)

⇒ RWS = 1 (receiver is willing to accept only the in_order frame.

Go_Back_N



Go_back can also employ individual ACKs or cumulative ACKs



let m = # of bits used for sequencing

max # of sequences = 2^m

for eg. m = 4

0, 1, ---, 15, 0, 1, ...

SWS (Go-back-N) must be

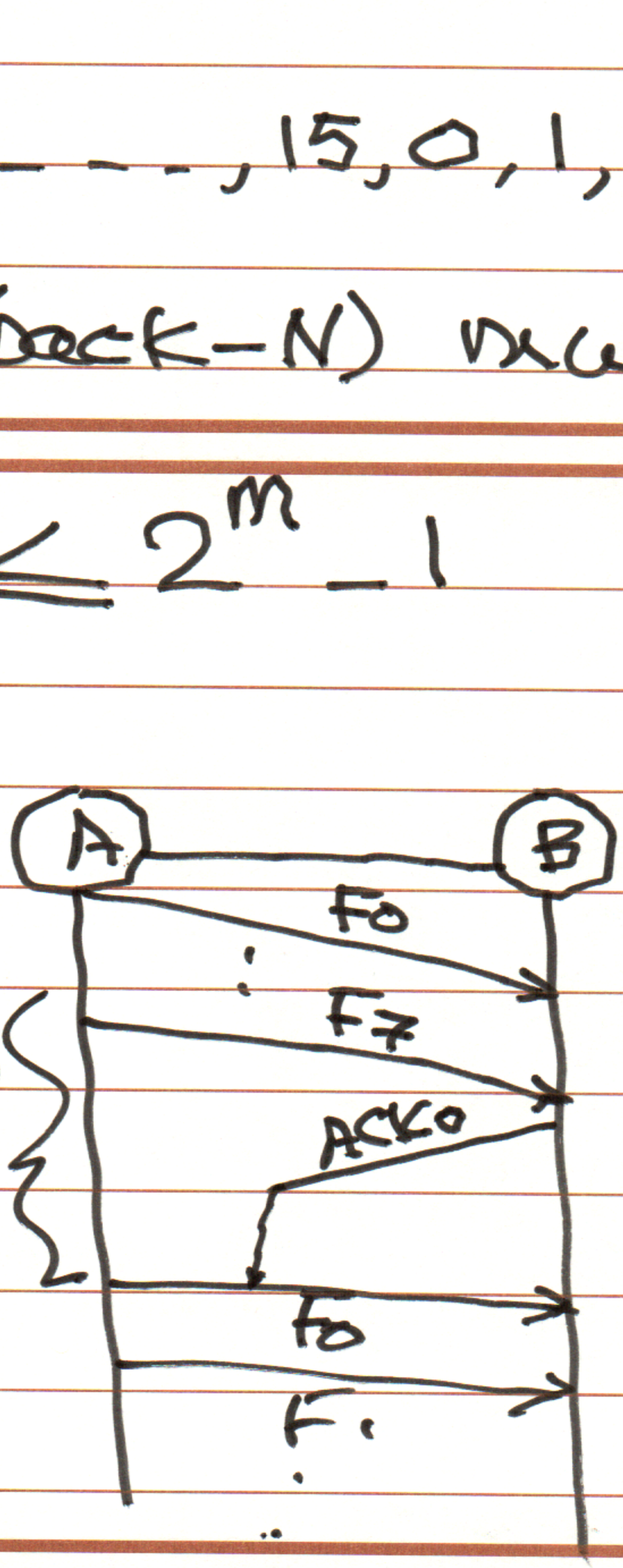
$SWS \leq 2^m - 1$

why?

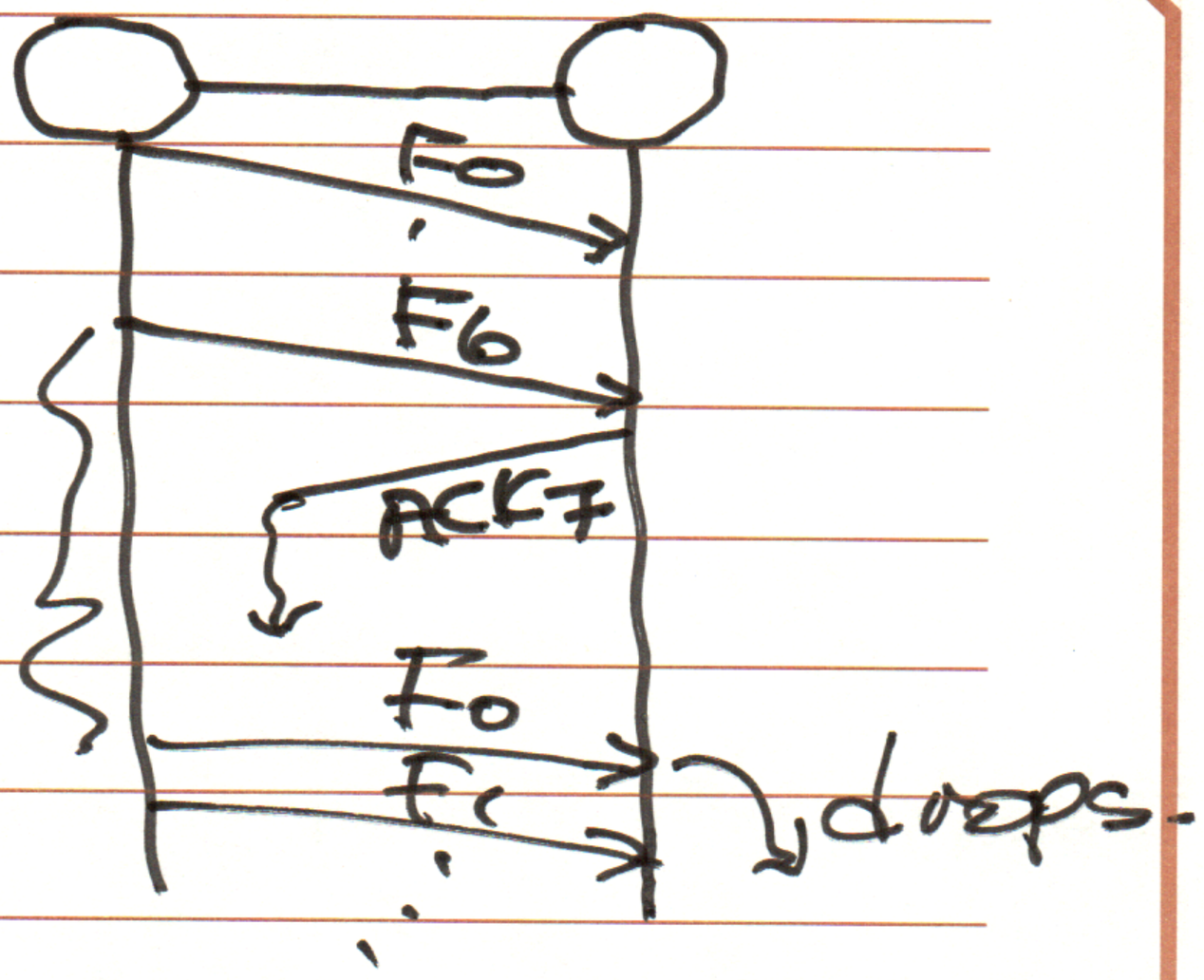
m = 3

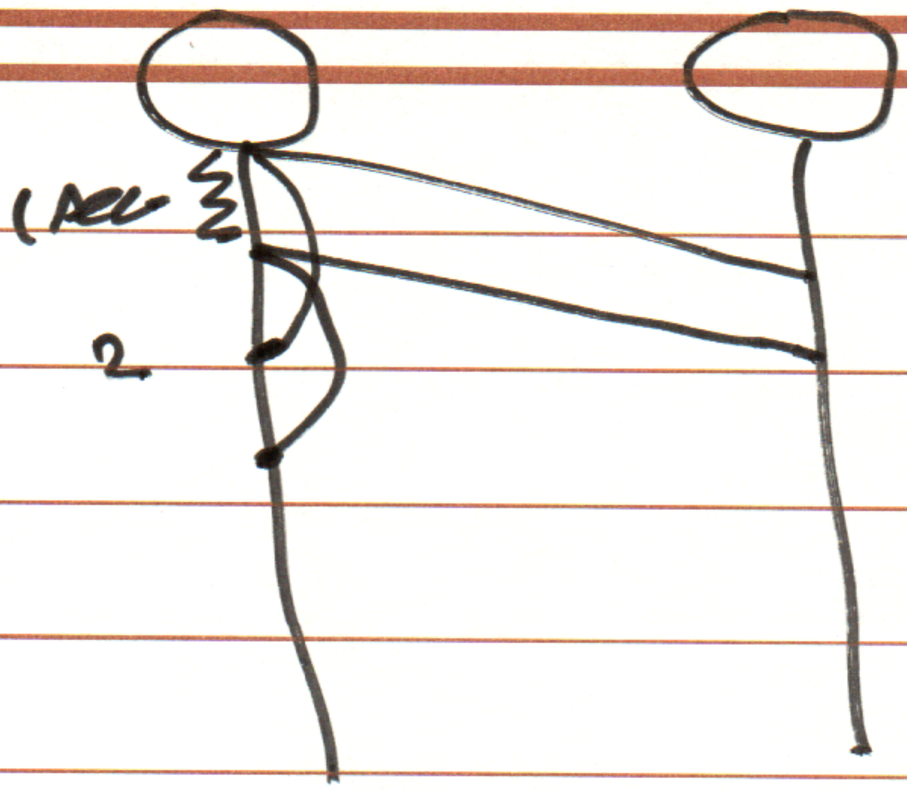
SWS = 8

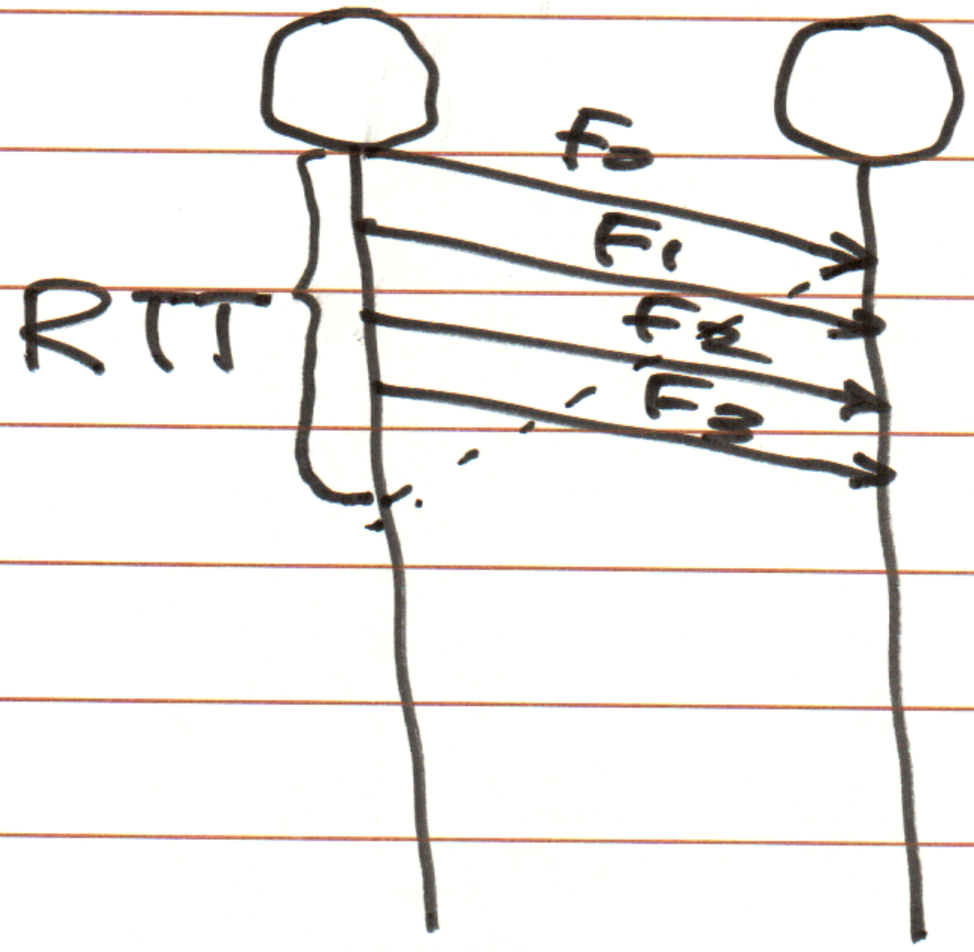
(violation)



SWS=7
(no violation)







$$\text{Throughput} = \frac{4 \text{ frames}}{RTT}$$

$$\text{Link utilization} = \frac{4T_f}{4T_f + 2T_p}$$

$$= \frac{4T_f}{RTT}$$