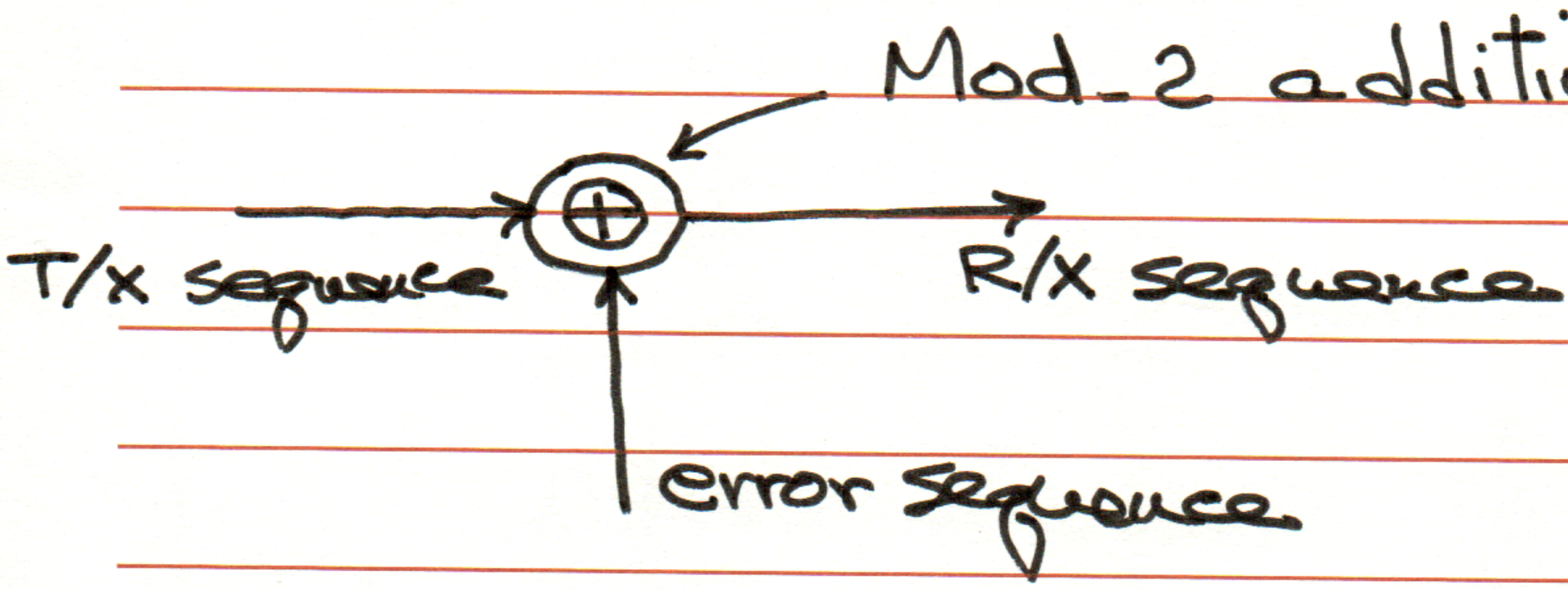
EE450, Fall 2021

Error Detection & Control



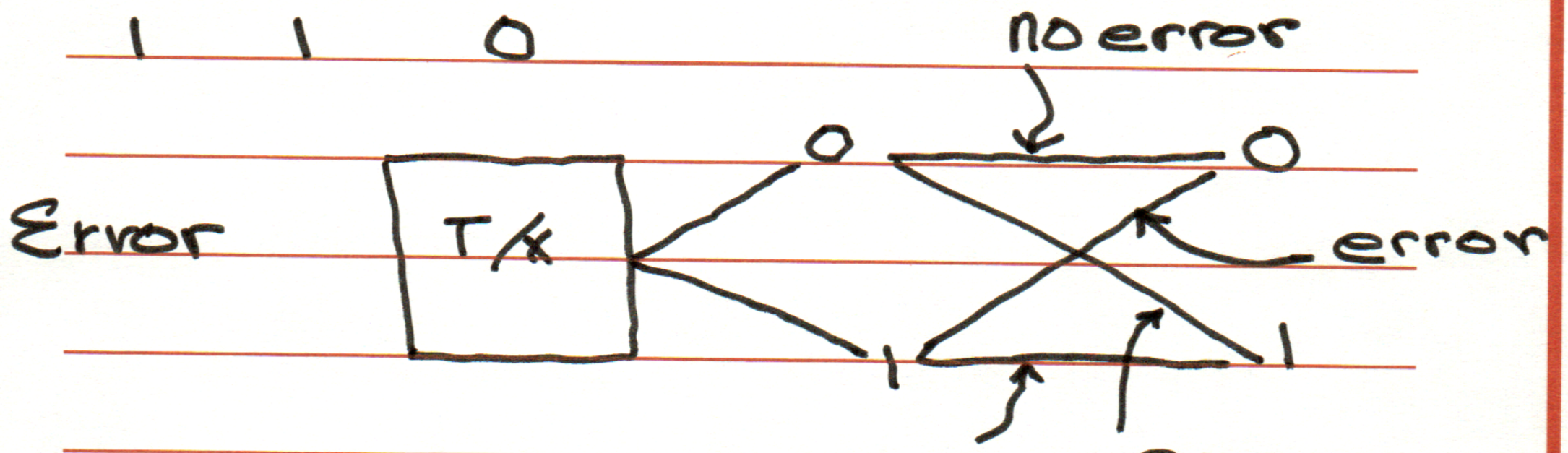
Mod-2 addition ⇒ X-OR Logic

0 0 0

0 1 1

1 0 1

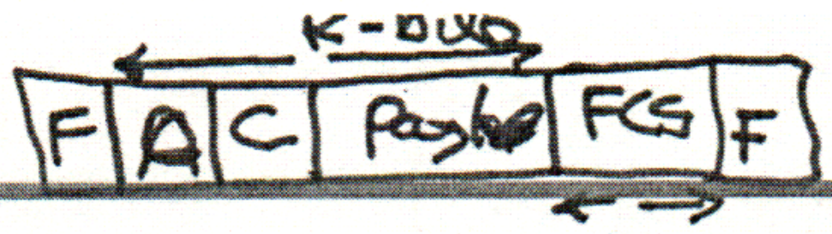
1 1 0



T/x 1 0 1 1 0 1 no

Error 0 1 1 0 0 1 error

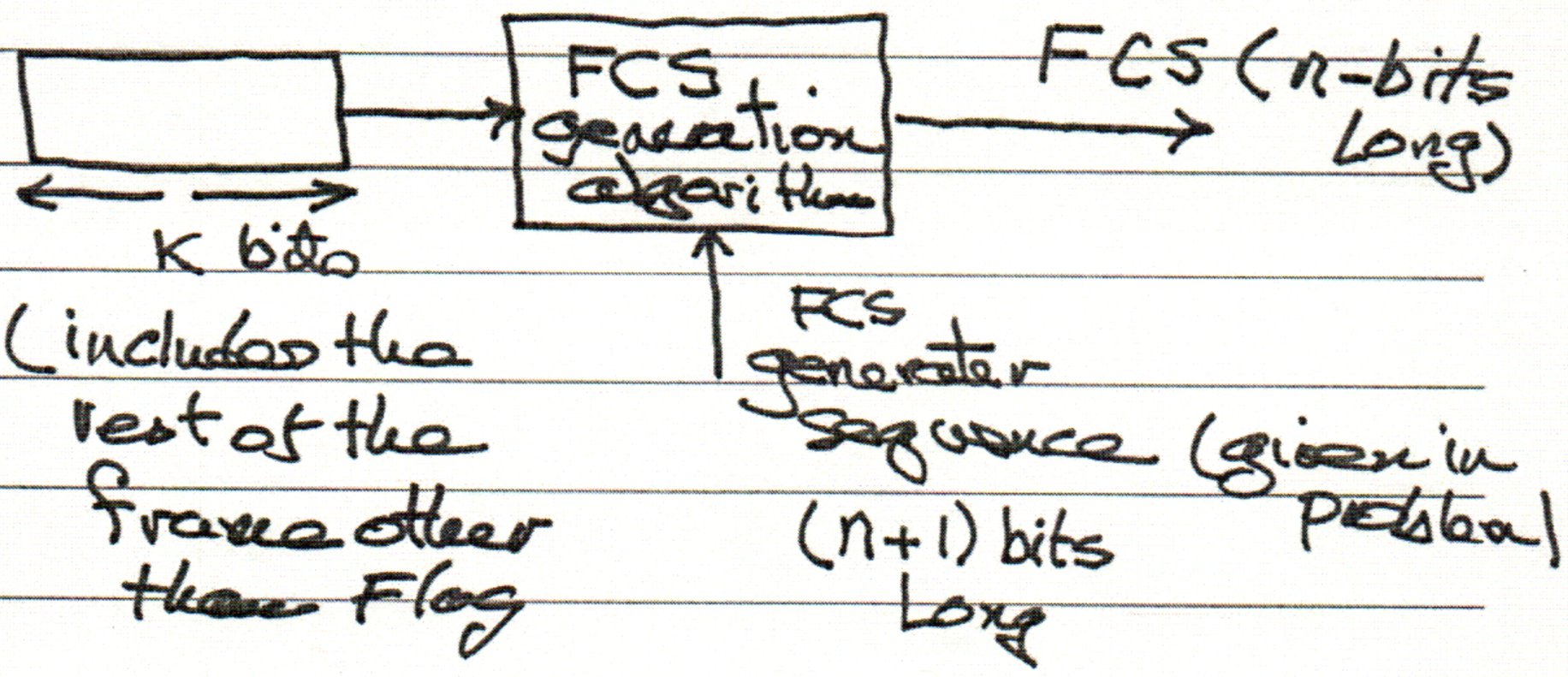
R/x 1 1 0 1 0 0 ← 3 errors have occured.



Error Detection algorithm

@ T/x side: The T/x will generate a n-bit sequence called the FCS Sequence (Frame check Sequence) and append them at the end of the frame (Trailer).

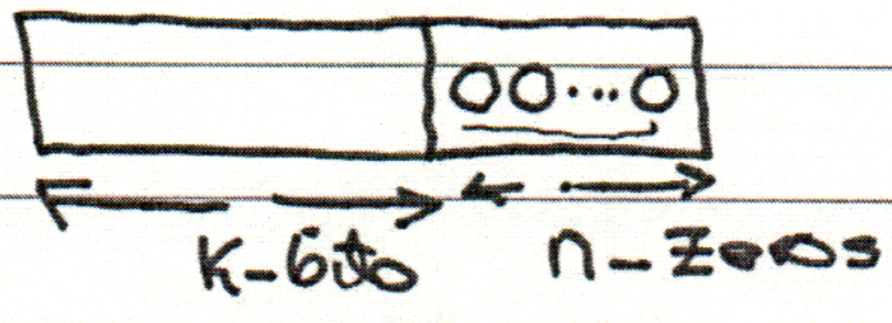
How does he generate the FCS bits (which are not part of the payload but they will help R/x to detect errors.



FCS generation algorithm (Divisor is Mod 2 opti

(n+1)

Generator polynomial

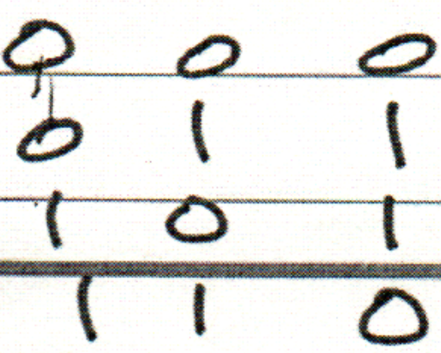


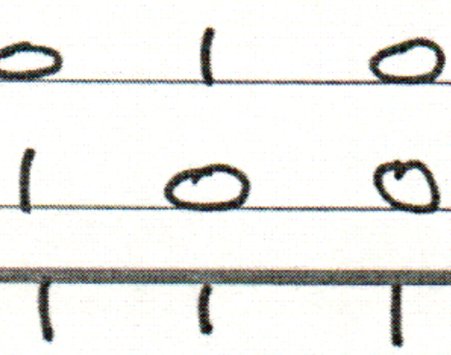
Remainder (n-bits long)

FCS-bits

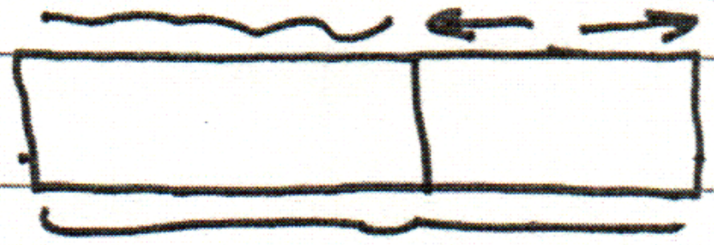
addn MOD2 multiplication

addition/subtract



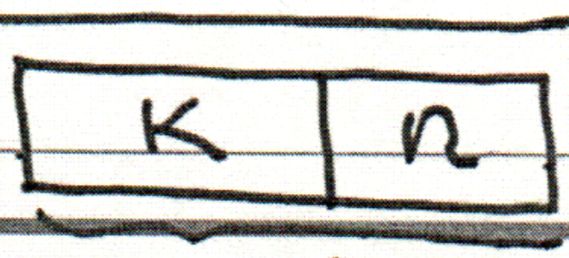


@ R/X



received sequence (which is NOT necessarily the TX sequence (i.e. some errors may have occurred).

generator sequence



receiver sequence

Remainder

If the remainder $\neq 0$

R/X declare the frame is erroneous and he will drop it.

If the remainder is 0, he will declare no errors are detected (but he could be wrong) (and he will always be correct in his decision.)

If errors did occur but remainder is still 0 we say the receiver failed to detect errors.

Polynomial representation

$x^5\ x^4\ x^3\ x^2\ x^1\ x^0$

101010 → x^5+x^3+x

101101 → $x^5+x^3+x^2+1$

@ receiver

$$x^3+x+1 \overline{)\ x^6+x^2+x\ }\ \ x^3+x+1$$

$x^6+x^4+x^3$

$x^4+x^3+x^2+x$

x^4+x^2+x

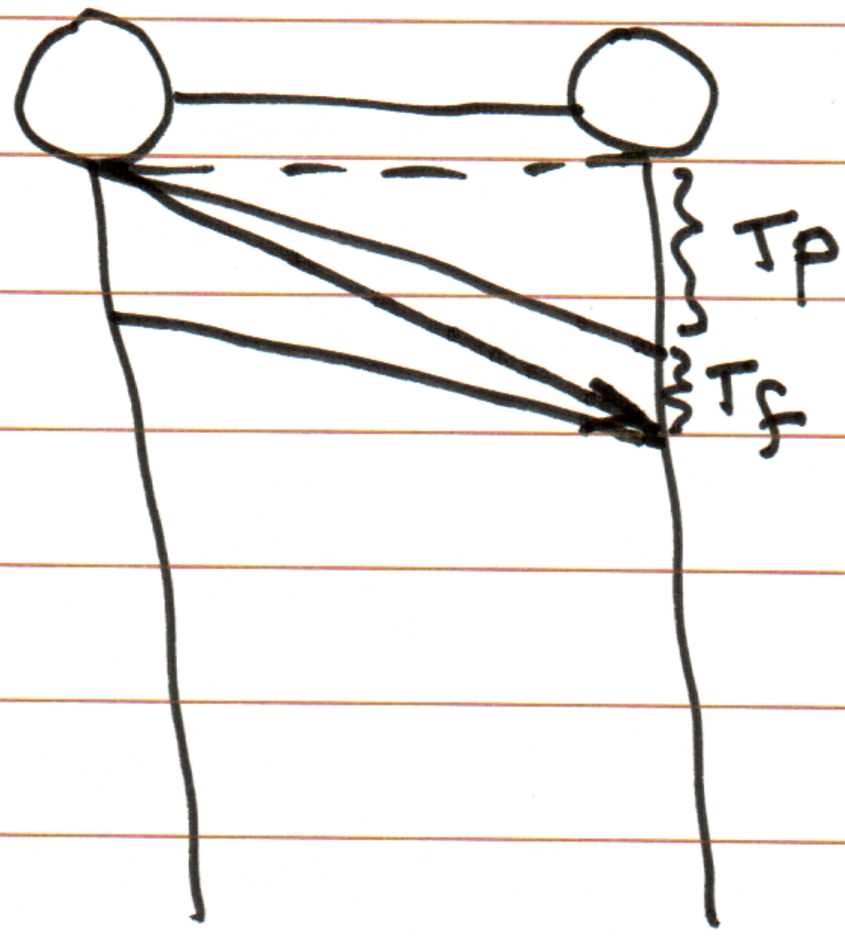
x^3

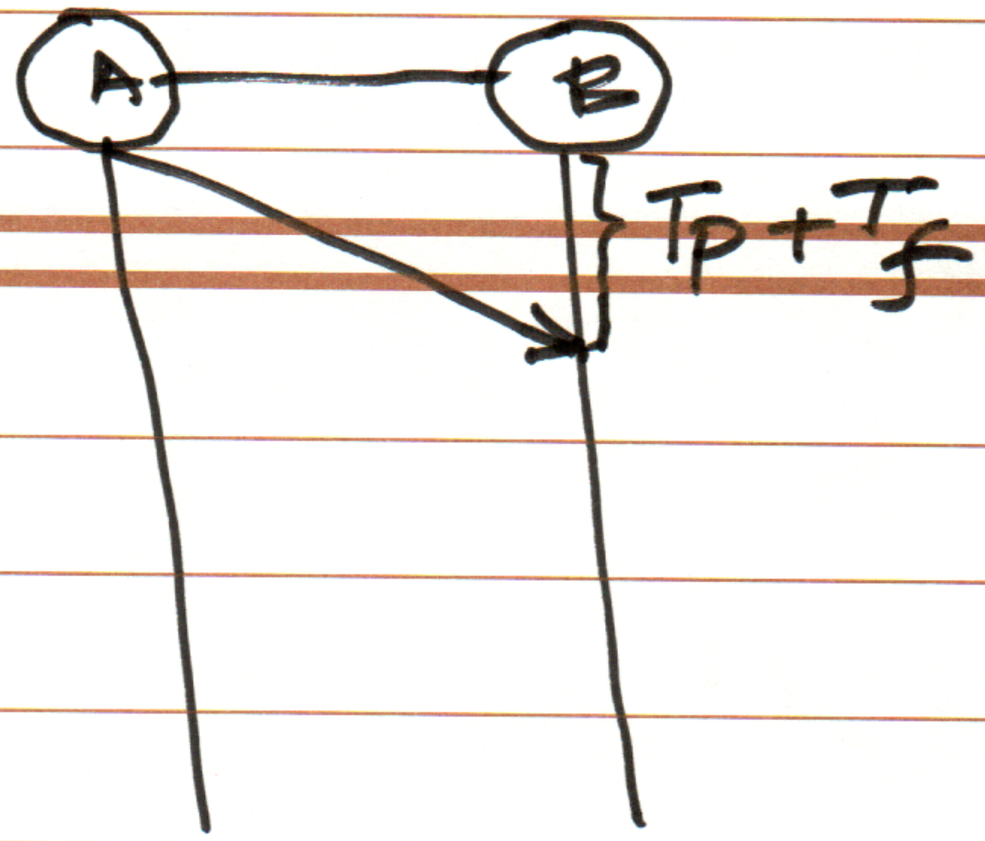
x^3+x+1

$\rightarrow x+1$

remainder

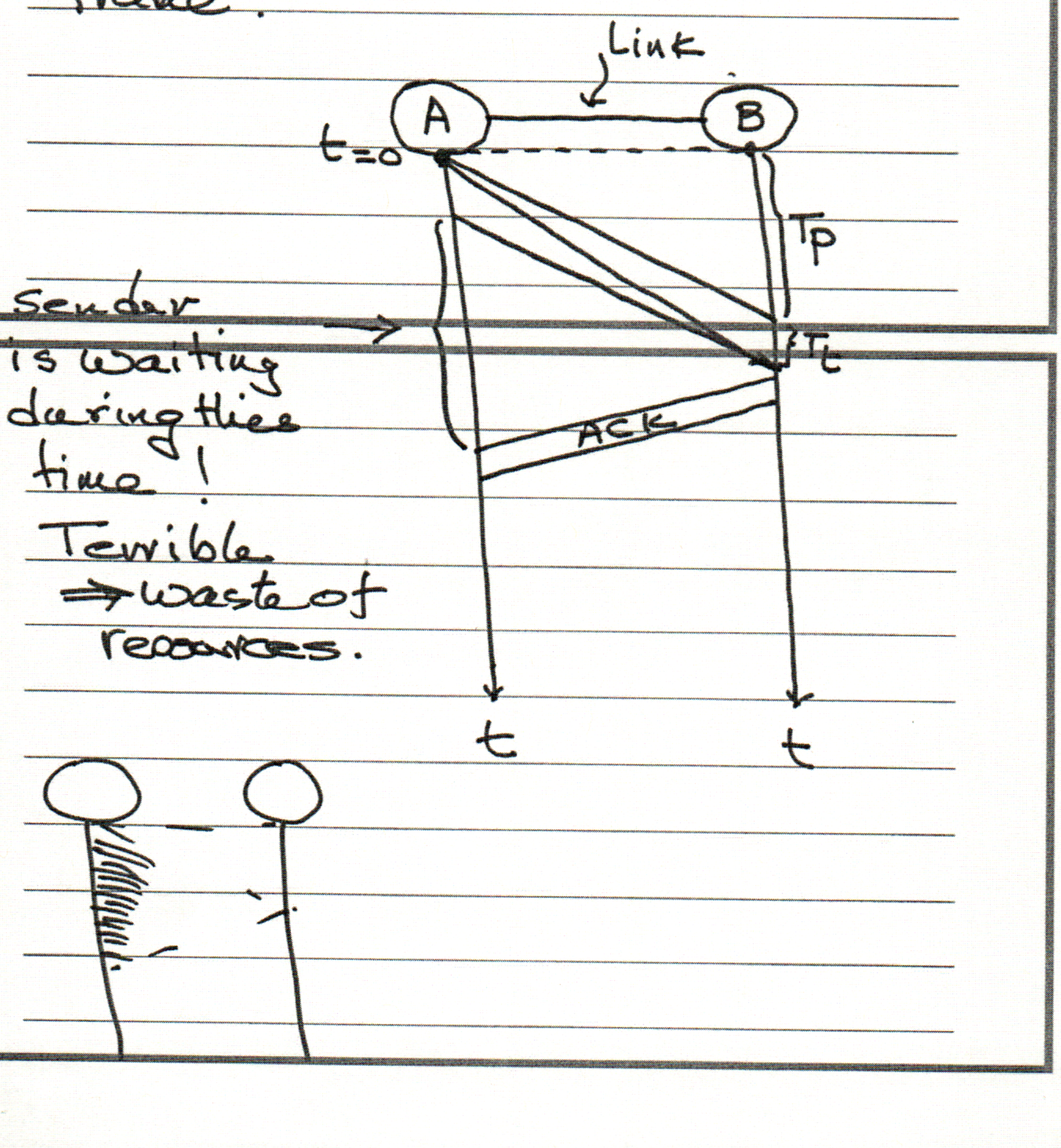
011


T_p
T_f


A
B
$T_p + T_f$

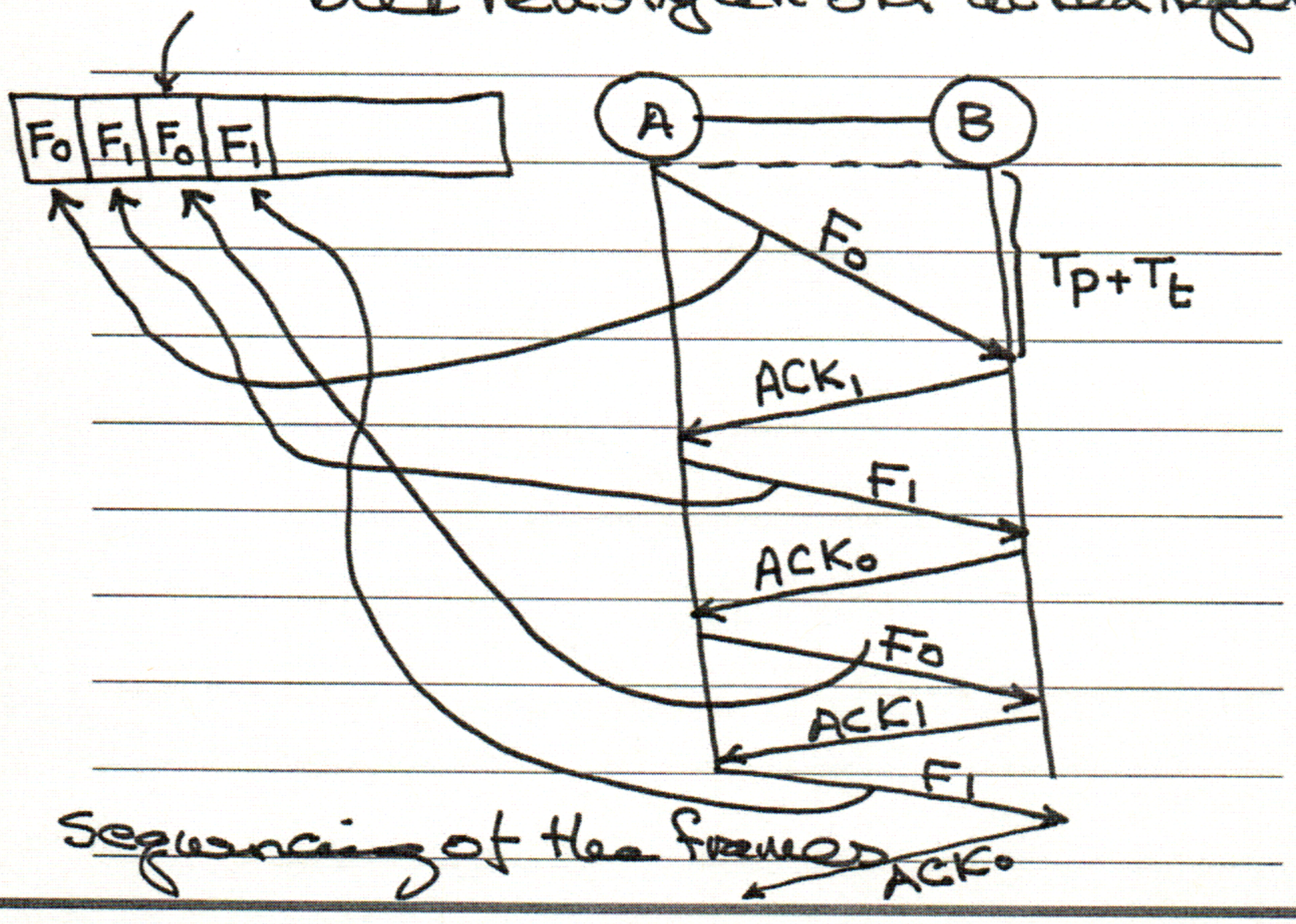
Stop & wait ARQ

T/X can transmit one frame at a time & then he has to stop, wait for ACK, before he can send another frame.



Terrible ⇒ waste of resources.

a brand new frame but resuing an old acked sequence #

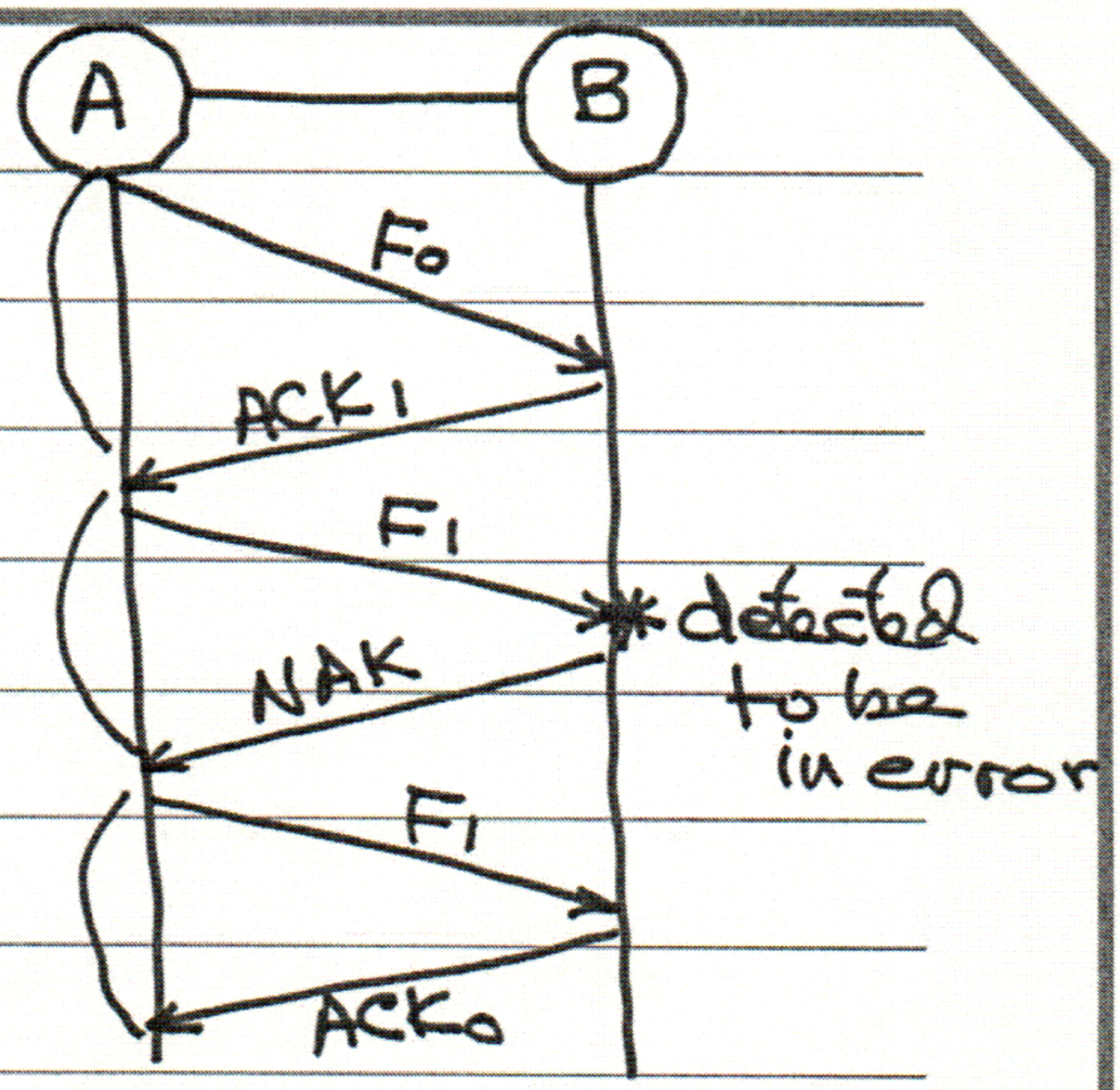


Sequencing of the frames

Here in S&W ARQ we only need two sequences 0 & 1 because there can only one frame in transit.

⇒ We need only one bit to sequence the frames. That bit resides in the "control field" of the frame.

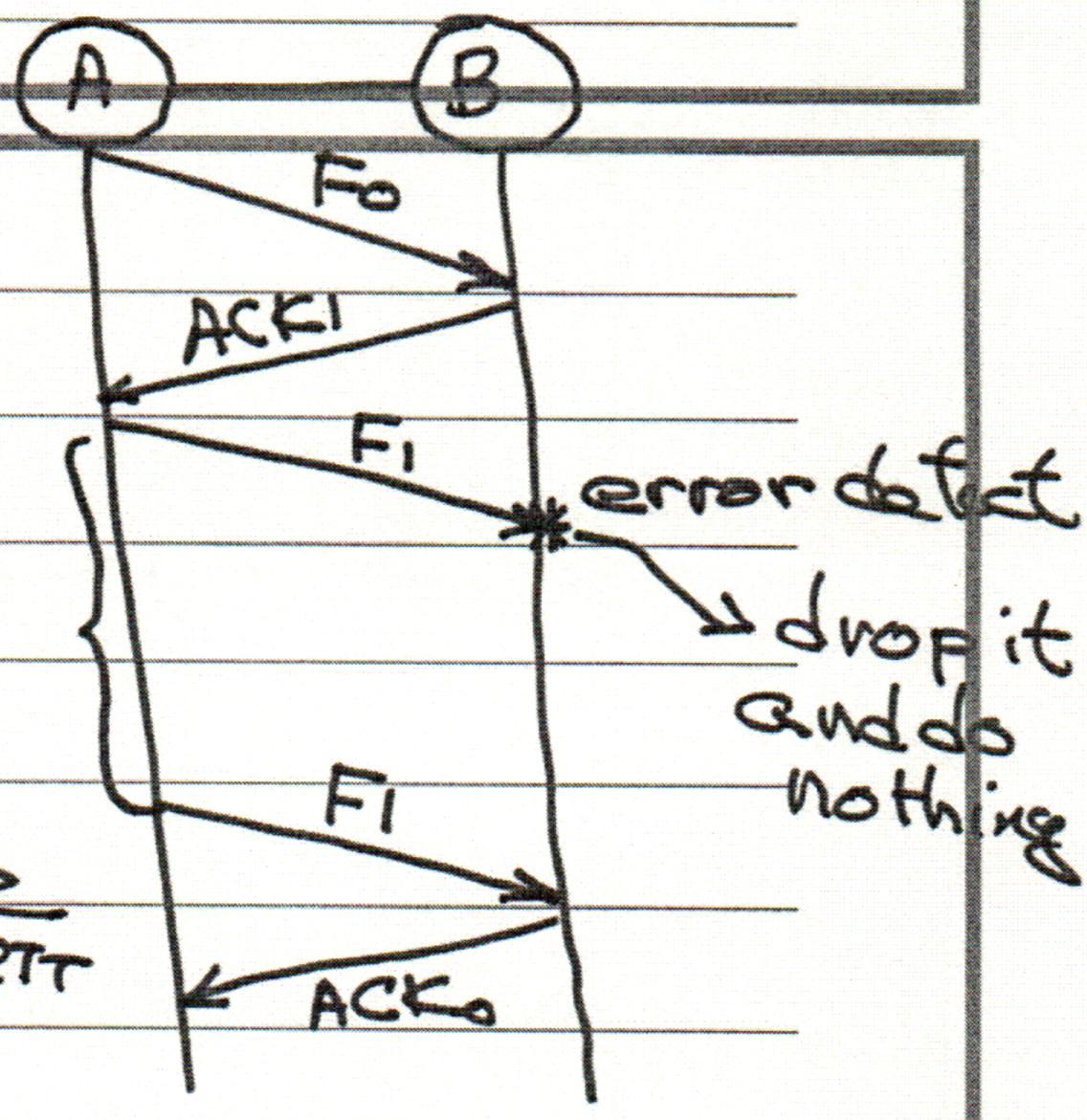
$$\text{Throughput} = \frac{4 \text{ frames}}{4 \text{ RTT}} = 1 \text{ frame/RTT}$$



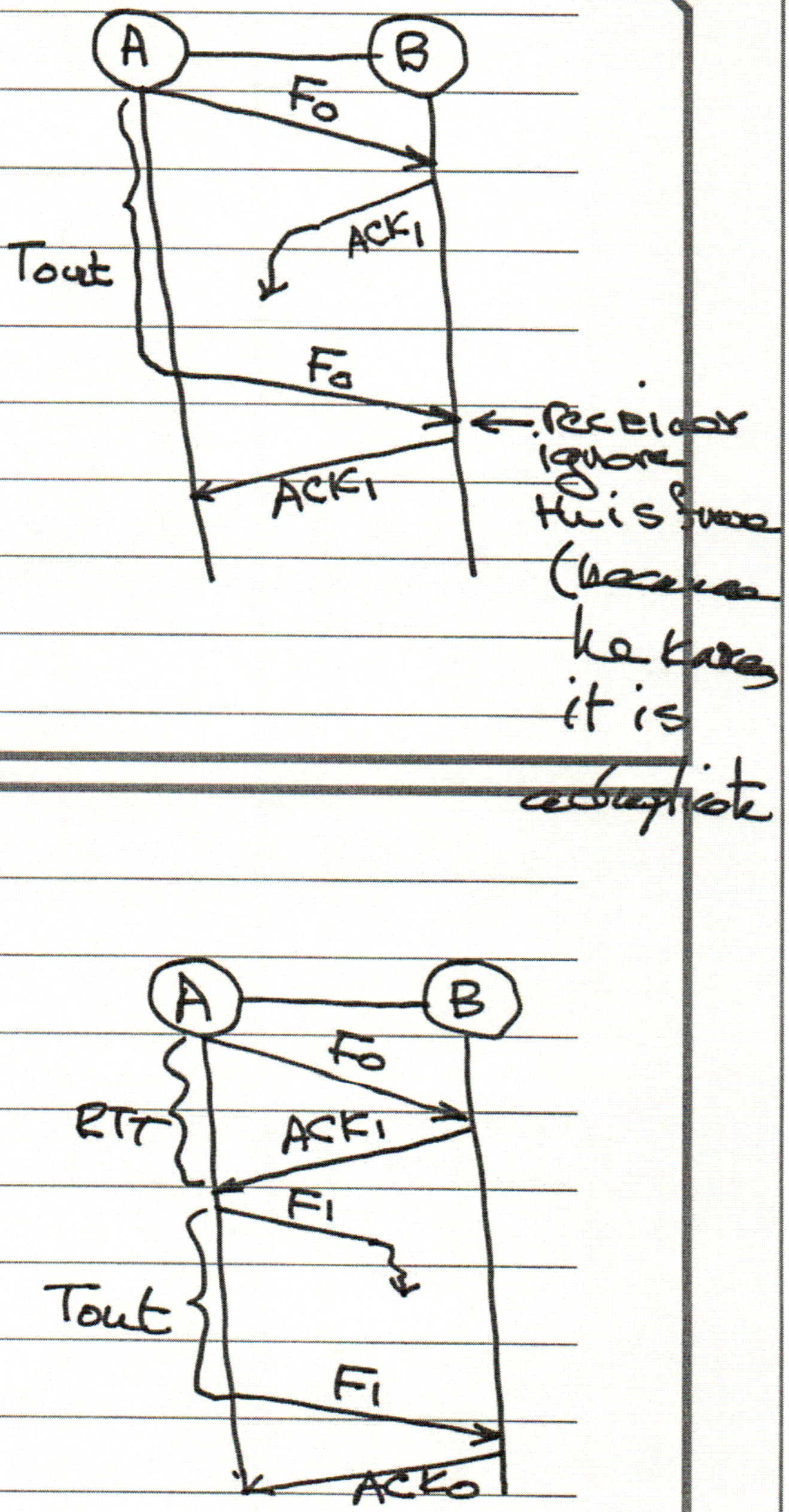
Throughput $= \frac{2 \text{ Frames}}{3 \text{ RTT}}$

→ not used

Protocols do not like NAK.



Throughput $= \frac{2 \text{ Frames}}{RTT + Tout + RTT}$



Throughput

$$= \frac{2 \text{ Frames}}{2RTT + Tout}$$

Sliding Window ARQ

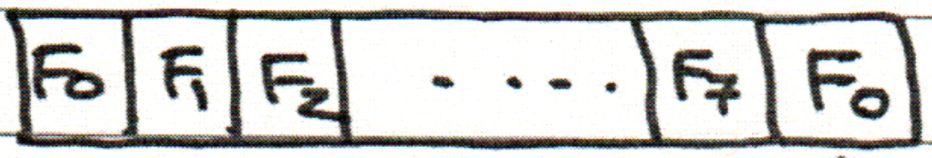
① Here the sender can transmit multiple frames continuously before he has to stop!.

When should he stop?

- # of frames sent in one RTT can't exceed the Bw x Delay product.
- you ran out of sequence numbers

let m = # of bits used for sequencing the frames (recall m = 1 for S&W ARQ)

for ex. Suppose m = 3

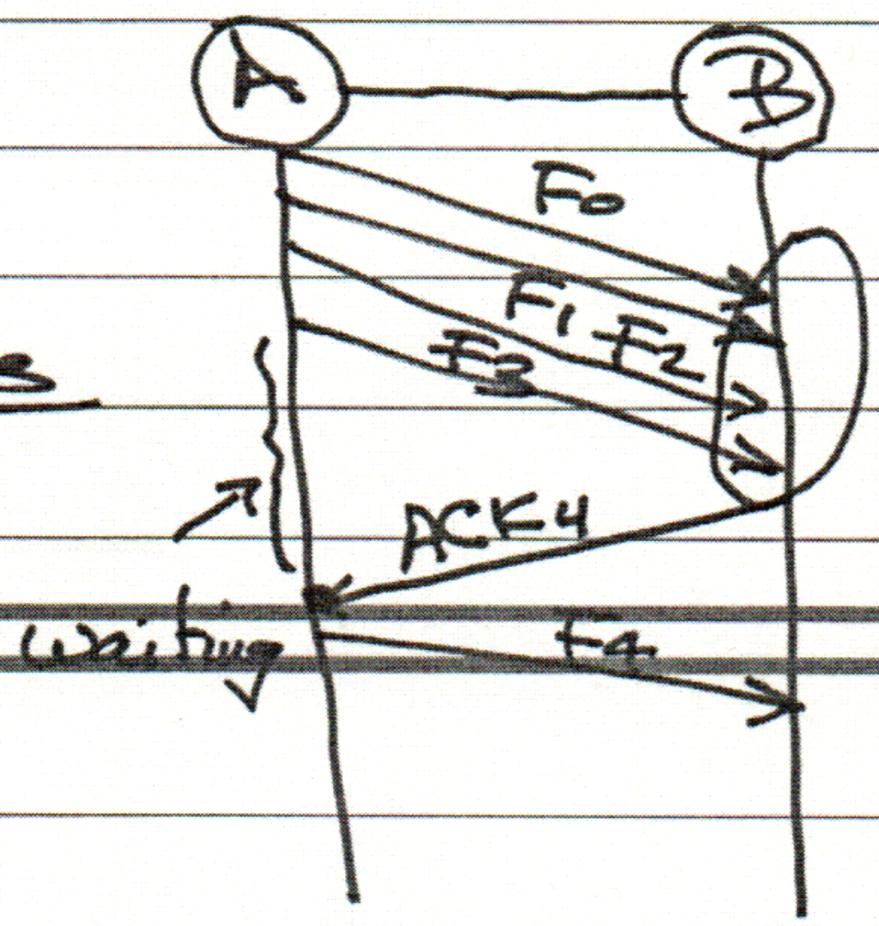


② In sliding window, the receiver has the option of ACK frames accumulatively (all frames need to be ACKed but not necessarily individual)

for $n = 3$

Throughput

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



③ In sliding Window ARQ, each side must maintain two sets of Windows

SWS: Sender Window Size

= Max # of unacknowledged frames the sender can send.

RWS : The max # of frames (both in-order & out-of order) the receiver is willing to accept (i.e. to buffer)

In Go-back-N ARQ

RWS = 1 The RX is only willing to accept the in-order frame.