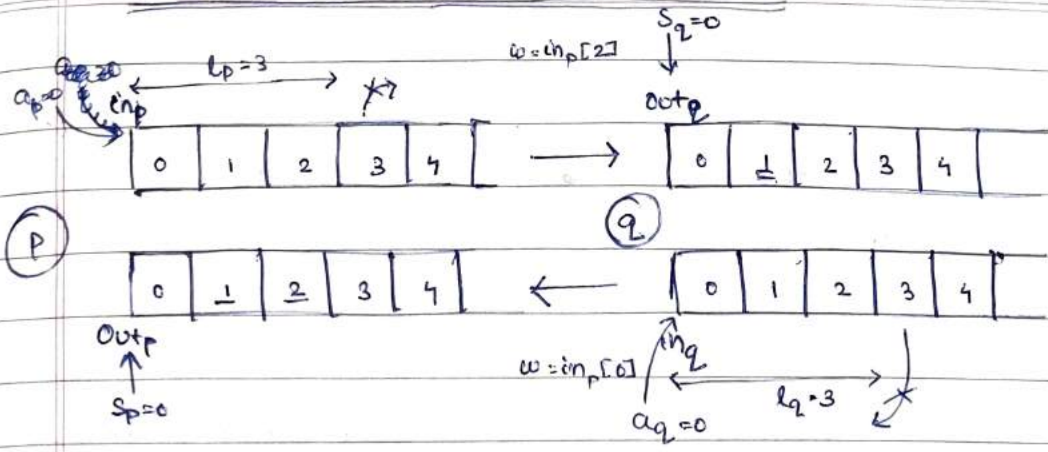
BALANCED SLIDING WINDOW PROTOCOL



safe delivery:

$$out_p[0 \ldots s_p - 1] = in_q[0 \ldots s_p - 1]$$

$$out_q[0 \ldots s_q - 1] = in_q[0 \ldots s_q - 1]$$

eventual delivery:

$s_p \geq k$, $s_q \geq k$ $\forall$ $k \geq 0$

liveness:

$P \rightarrow$ $s_p - l_q \leq a_p \leq s_q \leq a_q + l_p \leq s_p + l_p$

$P \Rightarrow$ $w = in_p[s_q]$ by p $\vee$ $w = in_q[s_p]$ by q is applic.
(no deadlock is possible).

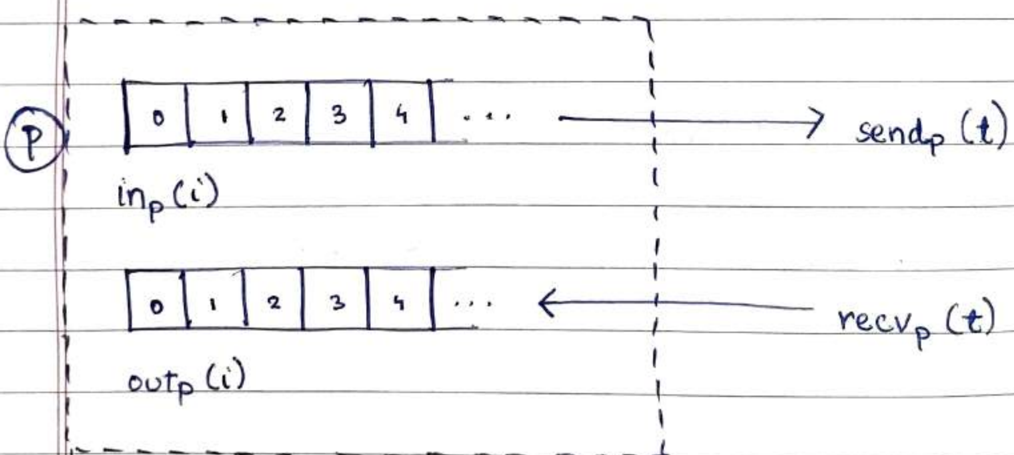
protocol satisfies requirement of eventual delivery.

SYSTEM I

- a process may send any packet, regardless of what it has recieved.
  $\{ send(i) \} \quad : i \in \mathbb{N}$

- a channel may choose to send any of the packets it has recieved at its input?
  $\therefore c_{out} \in \{ in_p[i] \mid in_p[i]$ has been sent by $p \}$.

  or, $out_c(i) = \{ in_c(j) \mid j < i \} \quad \forall \; i, j \geq 0$.



non-deterministic:

at $send_p(t)$ as it can choose which inp to send

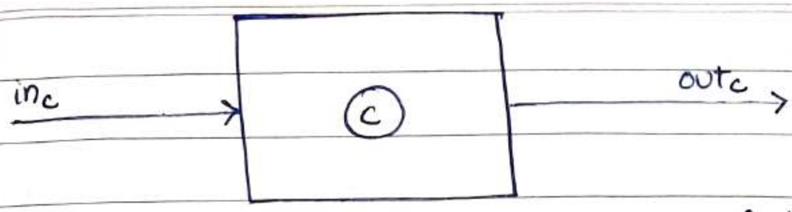
i.e., $send_p(t) = in_p(i) \; : \; i \in \mathbb{N}$

choice.

but, $recv_p(t) = out_p(j) \; : \; j \in \mathbb{N}$

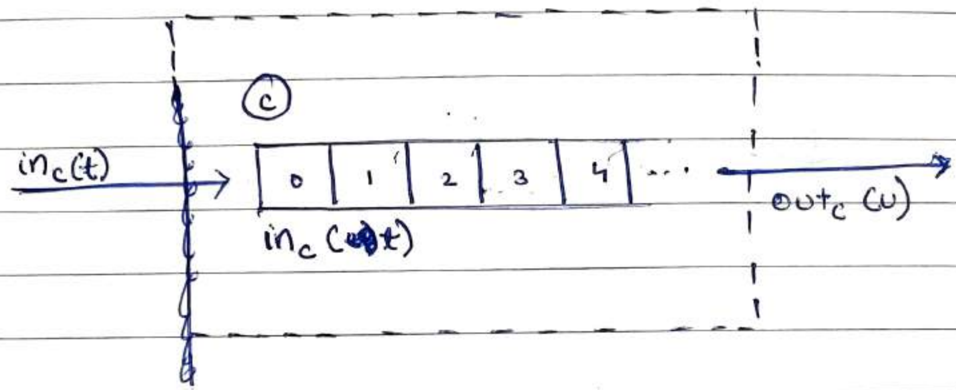
no choice

as $out_p(j) = in_q(j)$ choice



choice (nd.)

$$out_c(t) = in_c(u) \quad \text{s.t.} \quad u < t \qquad t, u \geq 0$$



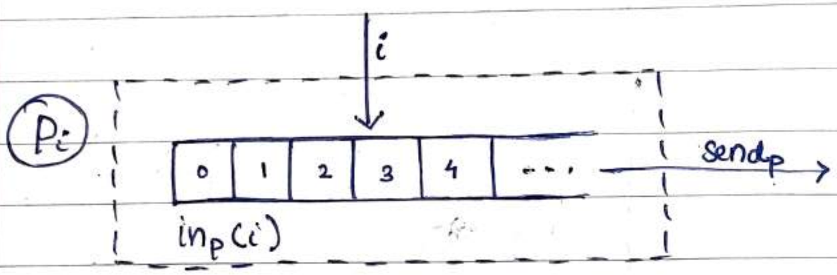
not allowing $out_c(u) = \phi$
as that could lead to no progress.

assumption: the channel transmits the message it recieved, as its, without any modification.

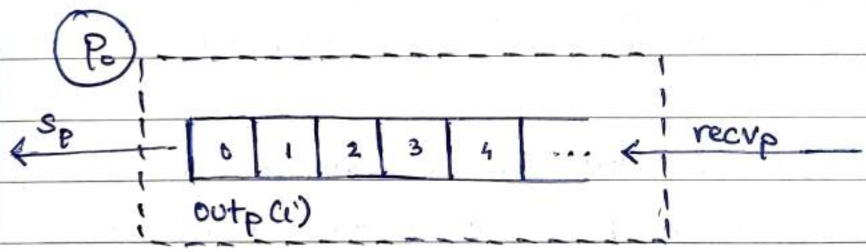
Date 12.2.2020
Page 4

SYSTEM 1 - attempt 2

now i am thinking, it would be (or could be) simpler to subdivide a process P into 2 subprocesses P_i and P_o.



$$send_p = \langle in_p(i), i \rangle$$



s_p : lowest frame no. not yet recieved.

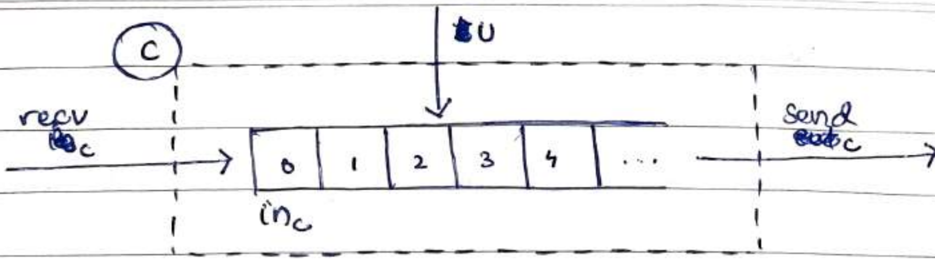
$$out_p(i) = recv_p = \langle in_q(j), i \rangle$$

$$s_p = \min(\{ i \mid i \neq \text{undef.} \})$$

or

$$s_p(t) = \min(\{ i \text{ in } recv_p(t) = \langle \_, i \rangle \})$$

$$= j \text{ st. } j \geq i \ \forall i \text{ in } recv_p(t) = \langle \_, i \rangle$$



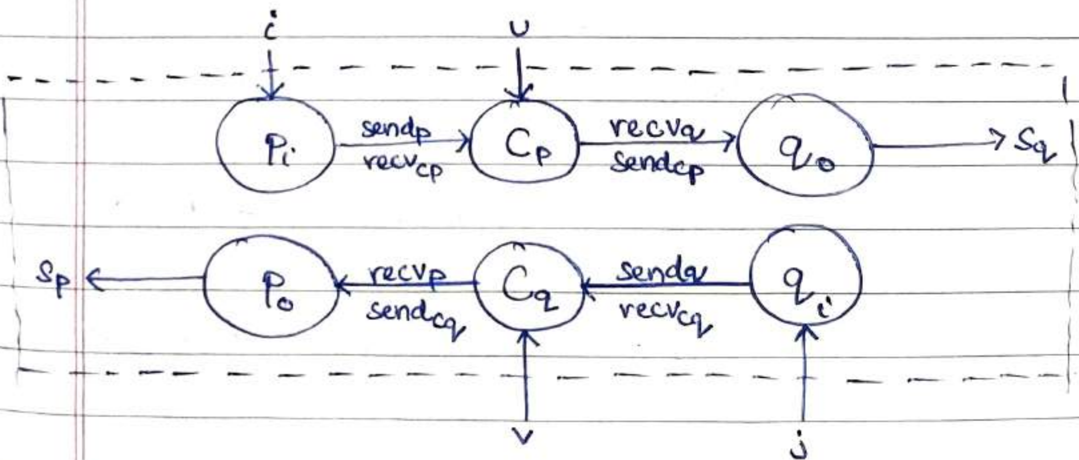
$out_c(t) = recv_c$

$send_c(t) = recv_c(u)$ s.t. $u < t$ $u, t \geq 0$.

or

$in_c(t) = recv_c(t)$

$send_c(t) = in_c(u)$ s.t. $in_c(u) \neq undef$.



4 inputs to interact with.

$S, \langle X_S, X_S^0, U_S, \xrightarrow{S}, Y_S, h_S \rangle$

$X_S = \langle in_P, out_P, in_q, out_q, in_{cp}, in_{cq} \rangle$

st. $out_P(i) = in_q(i)$ or undef.

$out_q(i) = in_P(i)$ or undef.

$in_{cp}(t) = in_P(i)$ or undef. $i \in \mathbb{N}$

$in_{cq}(t) = in_q(i)$ or undef. $i \in \mathbb{N}$.

$X_S^0 = \langle in_P^0, out_P^0, in_q^0, out_q^0, in_{cp}^0, in_{cq}^0 \rangle$

st. $out_P^0(i) =$ undef. $\forall i \in \mathbb{N}$

$out_q^0(i) =$ undef. $\forall i \in \mathbb{N}$

$in_{cp}(t) =$ undef. $\forall t \in \mathbb{N}$

$in_{cq}(t) =$ undef. $\forall t \in \mathbb{N}$.

$$U_s = \{ send_p(i), send_q(j), send_{cp}(v), send_{cq}(v) \}$$

s.t. $in_p(i) \neq undef.$

$in_q(j) \neq undef.$

$in_{cp}(v) \neq undef.$

$in_{cq}(v) \neq undef.$

Objective:

$$out_p(j) = in_q(j) \quad \forall \quad j \leq k$$

$$out_q(i) = in_p(i) \quad \forall \quad i \leq k$$

$k \in \mathbb{N}$

k: frames to be transferred.

in_p: frames ~~to~~ of process p

in_q: frames of process q

out_p: frames recieved by process p (of q).

out_q: frames recieved by process q (of p).

in_{cp}: frames sent through channel by process p
in_{cq}: frames sent through channel by process q

(i) $\langle in_p, out_p, in_q, out_q, in_{cp}, \cancel{out} \; in_{cq} \rangle$

$$\Big\downarrow send_p(i)$$

$\langle in_p, out_p, in_q, out_q, in'_{cp}, in_{cq} \rangle$

$in'_{cp} = \{ t \to \langle in_p(i), i \rangle \} \, in_{cp}$

s.t. $in_{cp}(t) = \text{undef.}$

$\wedge \quad t \leq x \;\; \forall \{ x \mid in_{cp}(x) = \text{undef.} \}$.

(ii) $\langle in_p, out_p, in_q, out_q, in_{cp}, \cancel{out} \; in_{cq} \rangle$

$$\Big\downarrow send_q(j)$$

$\langle in_p, out_p, in_q, out_q, in_{cp}, in'_{cq} \rangle$

$in'_{cq} = \{ t \to \langle in_q(j), j \rangle \} \, in_{cq}$

s.t. $in_{cq}(t) = \text{undef.}$

$\wedge \quad t \leq x \;\; \forall \{ x \mid in_{cq}(x) = \text{undef.} \}$.

(iii) $\langle in_p, out_p, in_q, out_q, in_{cp}, out_{cp} \rangle$

$$\Big\downarrow send_{cp}(v)$$

$\langle in_p, out_p, in_q, out'_q, in_{cp}, out_{cp} \rangle$

$out'_q = \{ i \rightarrow F \} \; out_q$

st. $i = i$ in $in_{cp}(v) = \langle F, i \rangle$

$F = F$ in $in_{cp}(v) = \langle F, i \rangle$

(iv) $\langle in_p, out_p, in_q, out_q, in_{cp}, in_{cq} \rangle$

$$\Big\downarrow send_{cq}(v)$$

$\langle in_p, out'_p, in_q, out_q, in_{cp}, out_{cp} \rangle$

$out'_p = \{ j \rightarrow F \} \; out_q$

st. $j = j$ in $in_{cq}(v) = \langle F, j \rangle$

$F = F$ in $in_{cq}(v) = \langle F, j \rangle$

objective:

we want to eliminate the choice of i in $send_p(i)$, and the choice of j in $send_q(j)$. this could make processes p & q deterministic.

classmate

Date 17.2.2020
Page 10

(2) $S_2 \langle X_{S2}, X^0_{S2}, U_{S2}, \xrightarrow[S2]{}, \rangle$

$X_{S2} = \langle in_P, out_P, S_P, l_P, in_Q, out_Q, \ldots in_{CP}, in_{CQ} \rangle$

here, S_P represents that all frames till $S_P - 1$ have been recieved at out_P and that S_P is the first frame yet to be sent.

l_P represents the frames window for sending, implying that any frame $\geq S_P + l_P$ cannot be sent, and process P must ensure first that frame S_P is recieved at out_P, after which it can increment S_P, and then will be able to send the frame $S_{P_{old}} + l_P$.

(this is a constant.)

the above repperesents additional 2 contrants that apply to the choice of $send_P(i)$. i is still choosable, but its choice must satisfy above constraints.

$X^0_{S2} = \langle in_P, out^0_P, S^0_P, l^0_P, in_Q, out^0_Q, in^0_{CP}, in^0_{CQ} \rangle$

st. (additional)

$S^0_P = 0$ $\quad\quad S^0_Q = 0$

$l^0_P = L_P$ (say 3) $\quad l^0_Q = L_Q$

$2y^3 + 2x3y^2 + x^2 6y$

$6 + 6 + 6$

$H = \nabla^2 x + \nabla x \nabla y + \nabla^2 y$

$U_{S_2} = U_{S_1}$ $= 2xy^3 + 2x$

(i) $\langle in_p, out_p, s_p, l_p, in_q, out_q, s_q, l_q, in_{cp}, in_{cq} \rangle$

$\big\downarrow \text{send}_p(i)$

$\langle in_p, out_p, s_p, l_p, in_q, out_q, s_q, l_q, in'_{cp}, in_{cq} \rangle$

iff $i \geq s_p \wedge i < s_p + l_p$

$\therefore$ same as for S_1

(ii) $\langle in_p, out_p, s_p, l_p, in_q, out_q, \ldots \rangle$

$\big\downarrow \text{send}_q(j)$

$\langle \ldots, in'_{cq} \rangle$

iff $s_q \leq i < s_q + l_q$

$\therefore$ Same as for S1.

(iii) $\langle \ldots \rangle$

$\big\downarrow \text{send}_{cp}(u)$

$\langle \ldots, out'_q, s'_q, \ldots \rangle$

$s'_q = u$ iff $u > s_q$

$\wedge\ out'_q(i) \neq \text{undef.}\ \forall i < u.$

$\therefore$ Same as for S_1

(iv) $\langle \dots \rangle$

$$\downarrow \text{send}_{cq}(v)$$

$\langle \dots, out'_p, s'_p, \dots \rangle$

$s'_p = v$ iff $v > s_p$

$\wedge\ out'_p(i) \neq \text{undef.}\ \forall\ i < v$

$\therefore$ same as for S_1.

③ $S_3 \langle X_{S3}, X^0_{S3}, U_{S3}, \xrightarrow{S3}, \rangle$

$X_{S3} = \langle in_p, out_p, s_p, l_p, t_p, in_q, out_q, s_q, l_q, t_q, in_{cp}, in_{cq} \rangle$

t_p represent the frame that was last sent.

$X^0_{S3}: t_{p_0} = -1$.

$U_{S3} = \{ send_p, send_q, \underbrace{send_{cp}(v), send_{cq}(v)}_{\text{channel is still non deterministic.}} \}$

(i) $\langle \dots \rangle$

$$\downarrow send_p$$

$\langle \dots t'_p, \dots, in'_{cp}, \dots \rangle$

$in'_{cp} = send_p(i)$ where $i = \max(t_p + 1, s_p)$

$t'_p = i$

(ii) similarly for $send_2$.

(iii), (iv) same as for S_2.

RTP:

(1) safety (all frames are recieved as is, in order).

(2) eventuality (atleast k frames can be recieved by both process with a finite no. of steps).