**Init**

Pending = []

ACK = [null]

Left = get\_left()

Right = get\_right()

Correct =

**Upon Event Crash(p) do:**

correct = correct \{p}

**Upon Event New\_Left(p)**

left = p

for all elem in pending

**trigger** FIFO\_send(pending.m,pending.ts,pending.sender) to left

**Upon Event New\_Rigth(p)**

right = p

for all elem in pending

**trigger** FIFO\_send(DATA, <pending.m,pending.ts,pending.sender>) to right

**Upon Event TO\_broadcast(m)**

if m not in pending //forse qui me lo posso risparmiare l’if perché lo sto //creando io il messaggio.

pending.add(<m,self,ts>)

if left != null

**trigger** FIFO\_send(DATA,<m,ts,self>) to left

if right != null

**trigger** FIFO\_send(DATA,<m,ts,self>) to right

**Upon Event Fifo\_deliver** (Data,<m,ts,s>) from pj

if m not in pending

pending.add(<m,self,ts>

if pj == right and left != null

**trigger** FIFO\_send(DATA,<m,ts,s>) to left

if pj == left and right != null

**trigger** FIFO\_send(DATA,<m,ts,s>) to right

**trigger** FIFO\_send(ACK,<m,ts,s>,myid) to pj //mando l’ack a chi me lo ha //inviato

**Upon event Fifo\_deliver**(ACK,<m,ts,s>,pj) from p

if s == self //se io ho mandato il messaggio allora raccolgo gli ack

ack[m] = pj

else

if p == left and right != null

**Trigger Fifo\_send**(ACK,<m,ts,s>) to right

if p == right and left != null

**Trigger Fifo\_send**(ACK,<m,ts,s>) to left

**Upon event** correct is a subset of ack and peding not empty

pending.sort() //sort secondo timestamp e process id

for all elem in pending

if pending.m not in delivered

delivered = delivered u {m}

**trigger\_TODeliver(m)**

pending = null

Validity -> siccome se qualcuno crasha il processo alla sua destra e sinistra rinviano i messaggi in pending a nella rete i messaggi verranno deliverati.

Integrity->ogni messaggio viene deliverato una volta sola perché si fa il check dei messaggi che sono già nella lista delivered

Agreement -> tutti i processi corretti che hanno m in pending mandano un ack al sender quindi tutti hanno la stessa lista di processi

Order-> I processi nella lista pending vengono ordinati alla fine dalla stessa funzione e vengono deliverati secondo l’ordine imposto da timestamp e id del processo.