**System** **model**

* Perfect p2p
* Reliable broadcast

**Client** **code**

**INIT**

Primary = r0

Timer = delta

**Upon** **event** invoke [x op(args) ci]

Trigger pp2pSend(REQUEST, x, op(args), ci) to primary

Start(timer)

Upon event pp2pDeliver(REQUEST,x, op(args), ci) to client

Stop (timer)

Timer=delta

Trigger[x ok(args) ci]

**Primary code**

**INIT**

Primary=r0

Correct = {r0,r1,r2…rn}

Backups = correct/primary

X = set\_initial\_value()

Ack = empty

Last\_executed\_operation = null

Pending={}

Busy=false

Upon event pp2pDeliver(REQUEST,x, op(args), ci) from ci

If not busy

X = process\_operation(op(args))

Last\_executed\_operation <= <op(args), ci>

For each p € backups

Trigger pp2pSend(UPDATE, x, last\_executed\_operation)

Else

Pending=pending unito a {<x, op(args), ci>}

Upon event pp2pDeliver(ACK,Leo,ri) from ri

Ack=ack unito {ri}

When backups contained in set of ACK

Trigger pp2pSend(REPLY, last\_execution\_operation) to client

**Backups code**

**INIT**

Primary=r0

Correct = {r0,r1,r2…rn}

Backups = correct/primary

X = set\_initial\_value()

Ack = empty

Last\_executed\_operation = null

**Upon event** pp2pDelive(UPDATE, x’, Leo)

X=x’

Last\_executed operation= Leo

Trigger pp2pSend(ACK, Leo)

Before reply we should reset the ack

Finisci da mezz’ora alla fine