ESERCIZI 3 DICEMBRE 2019

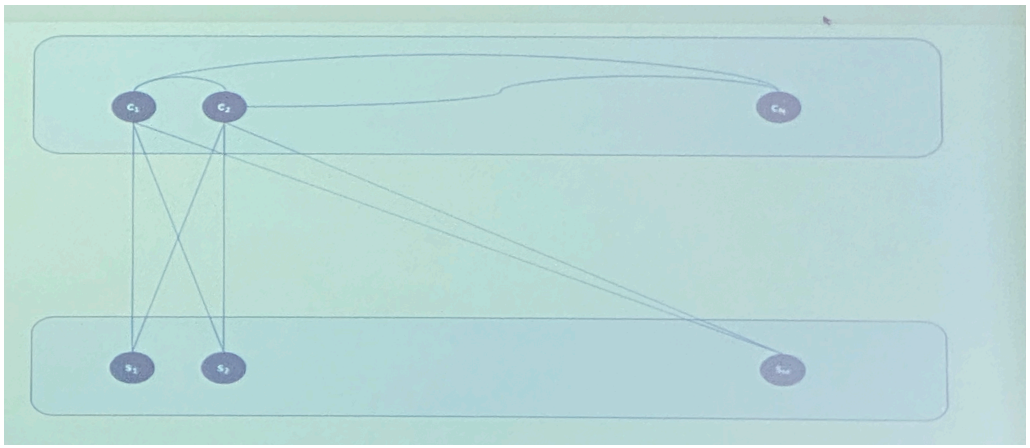
EX7

Ex 7:Consider a distributed system constituted by N client processes $C=\{c_1, c_2 \ldots c_N\}$ and M server processes $S=\{s_1, s_2 \ldots S_M\}$ with unique identifiers. Both clients and servers may exchange messages through perfect point-to-point links and each process (both clients and servers) may potentially send a message to any other.
The system is assumed to be synchronous: clocks are perfectly synchronized, the computation time is negligible and the latency of communication channels is upper bounded by a constant δ known by every processes.
Clients may experience crash failure while up to f servers (with $f < M/3$) may experience Byzantine failures.
Servers have access to a Byzantine tolerant consensus primitive.
Write the pseudo-code of an algorithm implementing the replication of a shared object X by following the active replication schema.

Solutions

Ex 1.1

***Solution*:**



Init Client

$waiting_i$ = false

$response_i$ = null

Init Server

$state_i$ = 0

$pending_i = empty$

consensus_$running_i$ = false

Client Code

upon event execute_operation(op)
    if not $waiting_i$:
        $waiting_i$ = true
        for each s_j in {s1, s2, … sm} **do**
            trigger pp2pSend(OP_REQ, op, i) to s_j
    else trigger abort(op)

upon event pp2pDeliver(ACK, op, j)
    $response_i$ = response U { j }

when | $response_i$ | > f
    $waiting_i$ = false
    response = empty
    trigger operation_completed(op)

Server Code:

upon event pp2pDeliver(OP_REQ, op, j):
    if <op, j> is not in executing in executed:
        $pending_i = pending_i$ U {<op, j>}

when $pending_i$ is not empty and not in consensus_$running_i$
    consensus_$running_i$ = true
    candidate = select_from(pending)
    trigger propose(candidate)

upon event decide (<op, j>)
    $pending_i = pending_i$ \ {<op, j>}
    state = execute(op)
    executed = executed u {<op, j>}
    trigger pp2pSend(ACK, op, I) to c_j
    consensus_$running_i$ = false