PDC Home Assignment - 4

2200030301-Hemanth

1: Fault Tolerant Message Passing in Distributed System

Fault-tolerant message passing ensures that communication between processes continues correctly despite failures. Techniques include message acknowledgment, retransmission, checksums for data integrity, and using persistent queues. Redundant paths and consensus protocols also help maintain communication integrity.

2: Chandy-Lamport Snapshot (P2 Initiates)

P2 records its local state.
P2 sends a marker on all outgoing channels.
Upon receiving the first marker, P1/P3/P4 record their local state and the state of the channel as empty.
If a marker is received later, the process stops recording that

channel.
Complete global state is compiled.
3: Total Storage = (10s / 0.2s) * 5MB = 50 * 5MB = 250MB

4: Recovery Time = Time after fault + next reset interval = 2s + 0.5s = 2.5s

5: Dijkstra's Self-Stabilizing Algorithm

Each process maintains a state and periodically updates it based on its neighbors. Regardless of the initial state, it converges to a legitimate configuration in finite steps, thus recovering from transient faults without external intervention.
6: Ricart-Agarwala (3 Nodes)
P1 and P2 request access (send REQUEST(timestamp, id)).

P3 receives both, replies to the lower timestamp.
P1 enters critical section.
P2 waits for REPLY from P1.
P1 exits, sends REPLY to

P2.
P2 enters critical section

.

7: Fault Tolerant Message Handling
Use acknowledgment and timeout.
If no ACK from P2, P1 retries or uses backup nodes. Persistent logging or consensus algorithms (e.g., Paxos) ensure eventual consistency or alternate resolution.
8: Global State Consistency
M1 -> M2 (happened-before). P3 records before M2 => It misses M2. Hence, it's not a consistent global state.

9: Bully Algorithm Execution (P2 Initiates)

P2 sends ELECTION to P3 and P4.
P3 and P4 respond OK.
P3 starts its own election.
P4 responds OK to P3.
P4 starts its own election, wins (highest ID), sends COORDINATOR.
Conflict Resolution: Higher ID wins. Lower ID processes stop upon receiving

OK/COORDINATOR.
10: Ring Election (P3 Initiates)
P3 sends (30) to P4 -> P5 -> P6 (fails) -> P1 -> P2 -> back to P3.
P3 elects highest ID (P5: 50), sends ELECTED(50).
Multiple initiators: Messages circulate; only highest ID remains.
11: Ricart-Agarwala (P2 then P3)
P2 sends REQUEST to P1, P3, P4.
All reply.
P2 enters CS.
P3 sends REQUEST (timestamp > P2).
Others reply; P2 delays REPLY.
P2 exits CS, replies to P3.
P3 enters CS.
Q12: Chandy-Lamport with Lost Marker
P1, P2, P3 record local states. P3 gets P1 marker but not P2. P4 doesn't get any.
Resulting snapshot is incomplete, as P4's state isn't recorded.
Q13: Marker Sending Rules for P1
Upon recording local state, send marker on all outgoing channels.
Upon receiving marker for first

Time:
Record local state.
Send marker on outgoing channels.
Record messages received on channel until marker arrives.
Correctness: Ensures consistent global state.
Complexity: $O(n)$ time and messages where n = number of processes.