RICART AGRAWALA ALGORITHM

GROUP 4

INTRODUCTION

- Distributed systems consist of multiple independent processes
- Processes often share resources
- Mutual exclusion is required to avoid conflicts
- Ricart–Agrawala is a permission-based algorithm for distributed systems

NEED AND IMPORTANCE

- Prevents simultaneous access to shared resources
- Avoids race conditions and data inconsistency
- Works without shared memory or central coordinator
- Ensures fairness among processes

PROBLEM STATEMENT

- Multiple processes request access to a critical section
- No global clock or shared memory
- Communication only through message passing
- Need a distributed mutual exclusion solution

ALGORITHM DESCRIPTION

- Uses REQUEST and REPLY messages
- Each process maintains a Lamport logical clock
- Process sends REQUEST to all others
- Enters critical section after receiving all REPLY messages

MESSAGE HANDLING RULES

- Send REPLY immediately if not requesting the resource
- Defer REPLY if:
    - Process is in critical section, or
    - Process has higher priority (earlier timestamp)

PRIORITY AND ORDERING

- Requests ordered by Lamport timestamps
- Tie broken using process ID
- Guarantees mutual exclusion and fairness

REAL-TIME APPLICATIONS

- Distributed databases
- Cloud resource allocation
- Distributed file systems
- Collaborative applications

REQUIREMENTS TO IMPLEMENT

- Unique process IDs
- Reliable message passing
- Logical clock implementation
- Request and reply handling mechanism

CONCLUSION

- Simple and effective mutual exclusion algorithm
- No deadlock or starvation
- High message overhead
- Suitable for small to medium distributed systems

THE END

THANK YOU FOR LISTENING