MODULE-2

Ricart–Agrawala Algorithm

- This algorithm is an extension and optimization of Lamport's Distributed Mutual Exclusion Algorithm.
- Like Lamport's Algorithm, it also follows permission based approach to ensure mutual exclusion.
- It dispenses RELEASE messages by cleverly merging them with REPLY messages

- In this algorithm:
- Two type of messages (**REQUEST** and **REPLY**) are used and communication channels are assumed to follow FIFO order.
- A site send a **REQUEST** message to all other site to get their permission to enter critical section.
- A site send a **REPLY** message to other site to give its permission to enter the critical section.
- A timestamp is given to each critical section request using Lamport's logical clock.
- Timestamp is used to determine priority of critical section requests. Smaller timestamp gets high priority over larger timestamp. The execution of critical section request is always in the order of their timestamp.

Algorithm:

- **To enter Critical section:**
  - When a site S_i wants to enter the critical section, it send a timestamped **REQUEST** message to all other sites.
  - When a site S_j receives a **REQUEST** message from site S_i, It sends a **REPLY** message to site S_i if and only if
    - Site S_j is neither requesting nor currently executing the critical section.
    - In case Site S_j is requesting, the timestamp of Site S_i's request is smaller than its own request.Otherwise the request is deferred by site S_j.

- **To execute the critical section:**
  - Site S_i enters the critical section if it has received the **REPLY** message from all other sites.
- **To release the critical section:**
  - Upon exiting site S_i sends **REPLY** message to all the deferred requests.
- **Message Complexity:**
  Ricart–Agrawala algorithm requires invocation of 2(N – 1) messages per critical section execution. These 2(N – 1) messages involves
- (N – 1) request messages
- (N – 1) reply messages

Optimization

- Consider two sites Si and Sj
- Si gives a request and gets a reply from Sj
- Si can execute for n no of times in critical section until Sj sends a request and Si gives a reply.
- With this change a site in Ricart-Agrawal algorithm requests permission from a dynamically varying set of sites and requires 0 to 2(N-1) messages per Critical Section.