AIM : Build a program to implement Ricart –Agrawala Algorithm

**Prerequisite :** Should Know Mutual Exclusion Concept

OBJECTIVES:

To learn communication methodology in distributed systems

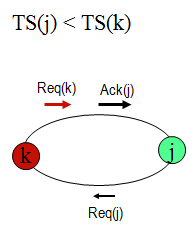
**Course Outcome :** Able to learn and apply the concept of remote method invocation and Remote Procedure Calls

THEORY:

The Ricart-Agrawala Algorithm is definitely an algorithm with regard to mutual exclusion on a distributed system. This algorithm is an extension as well as optimization associated with Lamport's Distributed Mutual Exclusion Algorithm, by removing the need for release messages. It had been developed by Glenn Ricart and Ashok Agrawala.

**Ricart-Agrawala Algorithm:**

The Ricart-Agrawala algorithm presumes the communication channels tend to be FIFO. The actual algorithm utilizes two kinds of messages: REQUEST as well as REPLY.



A process sends a REQUEST message to any or all additional procedures in order to request their permission to enter the critical section. A process sends a REPLY message to a process to give its permission to that process.

Processes use Lamport-style logical clocks to assign a timestamp to critical section requests as well as timestamps are used to decide the priority of requests.

Each process pi keeps the actual Request-Deferred array, RDi, the size of which is the same as the number of processes in the system.

At first, ∀i ∀j : RDi [j]=0. Whenever pi defer the request sent by pj, it sets RDi [j]=1 as well as following it's sent a REPLY message to pj, this sets RDi [j]=0.

### Terminology

* A *site* is any computing device which runs the Ricart-Agrawala Algorithm
* The *requesting site* is the site which is requesting to enter the critical section.
* The *receiving site* is every other site which is receiving a request from the requesting site.

### Algorithm

**Requesting Site**

* Sends a message to all sites. This message includes the site's name, and the current timestamp of the system according to its [logical clock](https://en.wikipedia.org/wiki/Logical_clock) (*which is assumed to be synchronized with the other sites*)

**Receiving Site**

* Upon reception of a request message, immediately sending a timestamped *reply* message if and only if:
* the receiving process is not currently interested in the critical section OR
* the receiving process has a lower priority (*usually this means having a later timestamp)*
* Otherwise, the receiving process will defer the reply message. This means that a reply will be sent only after the receiving process has finished using the critical section itself.

**Critical Section:**

* Requesting site enters its critical section only after receiving all reply messages.
* Upon exiting the critical section, the site sends all deferred reply messages.

### Performance

* Number of network messages; 2\*(N-1)
* Synchronization Delays: One message propagation delay

### Common Optimizations

Once site P i {\displaystyle P\_{i}} has received a r e p l y {\displaystyle reply} message from site P j {\displaystyle P\_{j}} , site P i {\displaystyle P\_{i}} may enter the critical section multiple times without receiving permission from P j {\displaystyle P\_{j}} on subsequent attempts up to the moment when P i {\displaystyle P\_{i}} has sent a r e p l y {\displaystyle reply} message to P j {\displaystyle P\_{j}} . This is called Roucairol-Carvalho optimization or Roucairol-Carvalho algorithm.

### Problems

One of the problems in this algorithm is failure of a node. In such a situation a process may starve forever. This problem can be solved by detecting failure of nodes after some timeout.

**Conclusion**

Thus, students understood implement Ricart –Agrawala Algorithm

**FAQ’s :**

What is Mutual Exclusion in distributed operating system?

Differentiate between Non-TokenBased and Token Based algorithm?

Explain the Logic of Ricart-Agrawala algorithm.