Unit - 3

1. Ricart - Agrawala's Algorithm:

* Ricart - agrawala's algorithm is an optimization on Lamport algorithm

* Ricart - agrawala algorithm uses only two types of messages. they are Request and reply.

* It keeps logical clock. which updated according to the clock rules

* This algorithm requires total ordering of messages or requests.

* Requests are ordered according to the global logical timestaps.

* Each process keeps it's state with respect to the critical section they are released, requested or held.

Requesting the Critical Section:

1) Request when S_i wants to enter the Critical Section: it broadcast timestamped Request message to all site

2) when a process receives a Request message, it may be in one of three states:

Case 1:

The receiver is not interested in the Critical Section, Send reply (ok) to the Sender

Case 2:

The receiver is in the Critical Section: do not reply and add the request to the local queue of requests

Case 3:

The receiver also wants to enter the Critical Section and Send no reply.

Executing the Critical Section:

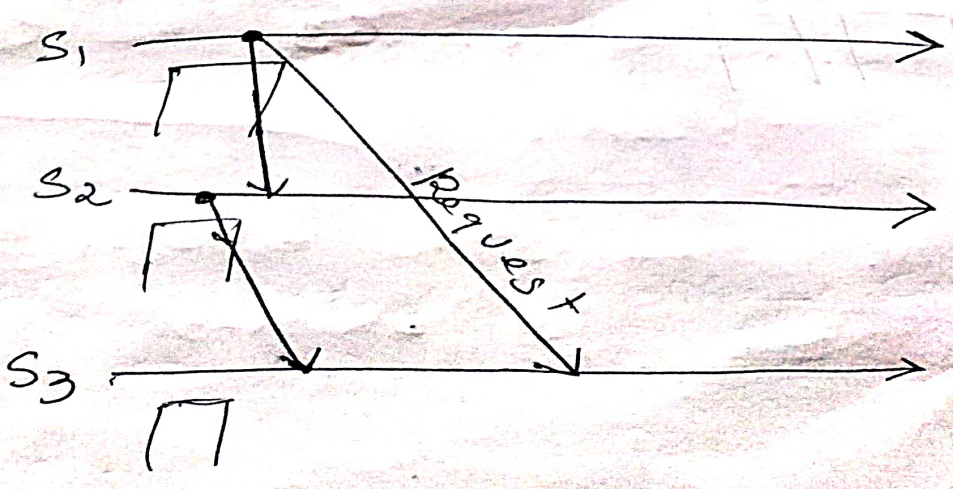
3) When the site S_i enters the Critical Section it ~~received~~ sends REPLY message from all the site in H_i's request set

Releasing the Critical Section:

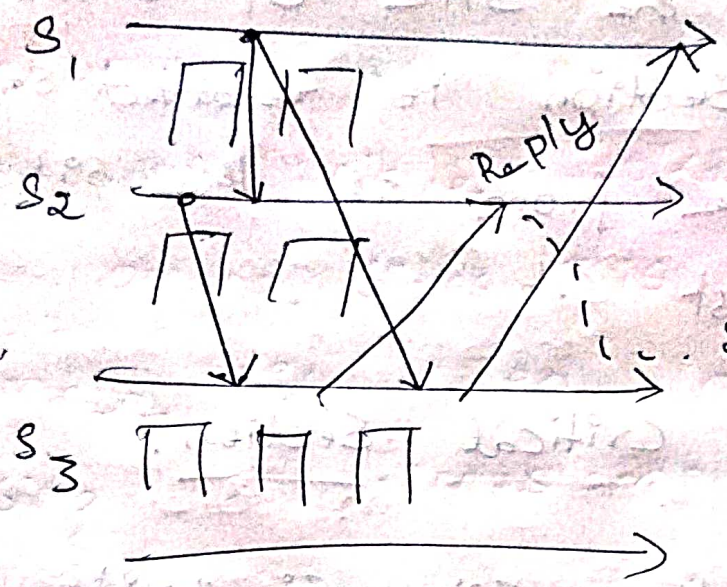
4) When the site S_i exits the Critical Section, it sends RELEASE message to all sites in its request Set.

Diagram:

Step 1: Site S_1 and S_2 are making request for CS:



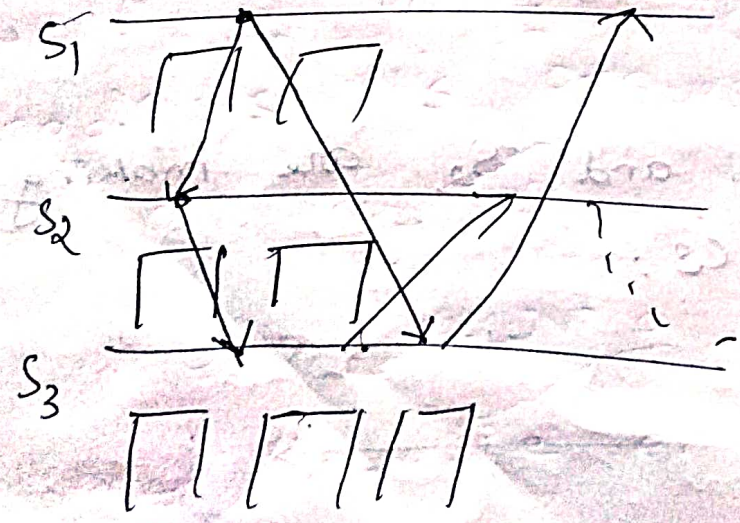
Step 2:



S_2 enters the CS

Step 3:

Site exits the CS.



S_2 leaves the CS.

2) Token - Based Algorithms.

* In token's - based algorithms a unique token is shared among all the sites in distributed computing System.

* In non - token based algorithm there is no concept of sharing token for access

* Token based algorithms are Suzuki - Kasami algorithm and Raymond's tree

* A site can access the lock if it has a token. Every process maintains a Sequence number (request id)

Suzuki - Kasami's Algorithm:

* Logical token representing the access right to shared resources.

* whoever holds the token is allowed to enter the Critical Section.

* A unique token is shared all among sites. This means a sequence number is used instead of using a timestamp.

* Sites increments its sequence number every time it requests the token.

* To enter the CS, a site broadcasts it's REQUEST message the all other sites.

* upon receiving a REQUEST message, the site enters the CS if it has token.

* When the site exists the CS, the token is exipred

Major Design issue

i) No release / Reply

ii) Each Sites need to request token

iii) The site need to know which Site to send the next token.

performance:

* This algorithm is simple and efficient.

* It requires 0 or N messages per CS invocations

* No Synchronization delay.

Algorithm:

Requesting CS:

When Si request to enter the CS, it generate a token and by incrementing Sequence id.

Executing CS:

Executes the site Si if it has received the token.

Releasing CS:

If token queue is empty it relasing the CS.

3) Deadlock:

* A process can be in two states; Running or blocked.

* In running state, a process has all the needed resources. and is either execution or ready for execution.

* In the blocked state, a process is waiting to acquire some resources.

following condition hold if deadlock occurs:

1) mutual exclusion - one process executes at time

2) No-premption

3) Hold and wait

4) Circular wait

Deadlock Avoidance:

*Deadlock can be avoided by allocating a resource and by resulting global state.

Deadlock Detection

Models of Deadlock:

AND Model

OR Model

AND-OR Model

AND Model:

* Set of deadlocked processes where each process waits for a resource held by another process
* Uses AND Condition.
* It is used in a single resource model.

OR Model:

* Set of deadlocked processes waits to receive messages from other processes.
* Uses OR Condition.
* In the OR Model, the presence of a knot indicates a deadlock.

AND-OR Model:

It is a generalization of previous two models.

4) Lamport's Algorithm:

* Each process, freely and equally competes the right to use the shared resource

* Requests are arbitrated by central control site or by distributed agent or agreement.

* permission to access from all processes.

* It uses a logical timestamp

* It is an non-token based algorithm.

* This algorithm is similar to Ricart-Agarwala, Raicourol-Carvaho algorithm

Algorithm

* Requesting CS
* Executing CS
* Releasing CS.

optimization:

* In Lamport's algorithm, Reply messages can be omitted in certain situations.
* With this optimization, Lamport algorithm requires between 3(N-1) and 2(N-1) messages per CS execution

5) Chandy-Misra-Haas Algorithm:

* It is the one of the best deadlock detection algorithm in distributed systems.
* This is considered an edge chasing and probe-based algorithm.
* If a process request a resource which fails or times out. Then the process generates a probe

message and sends to each of processes.

* Each probe message contains the id of the process that is blocked, and version of the probe message.

* When the process receive the probe message, it checks to see if it is also waiting for the resource. If not, it will eventually finish and release the probe message.

Advantages:

* Easy to implement
* Each probe Message is of fixed length
* There is very little computation
* " " " " overhead
* No need of construction graph
* There is no special data structures