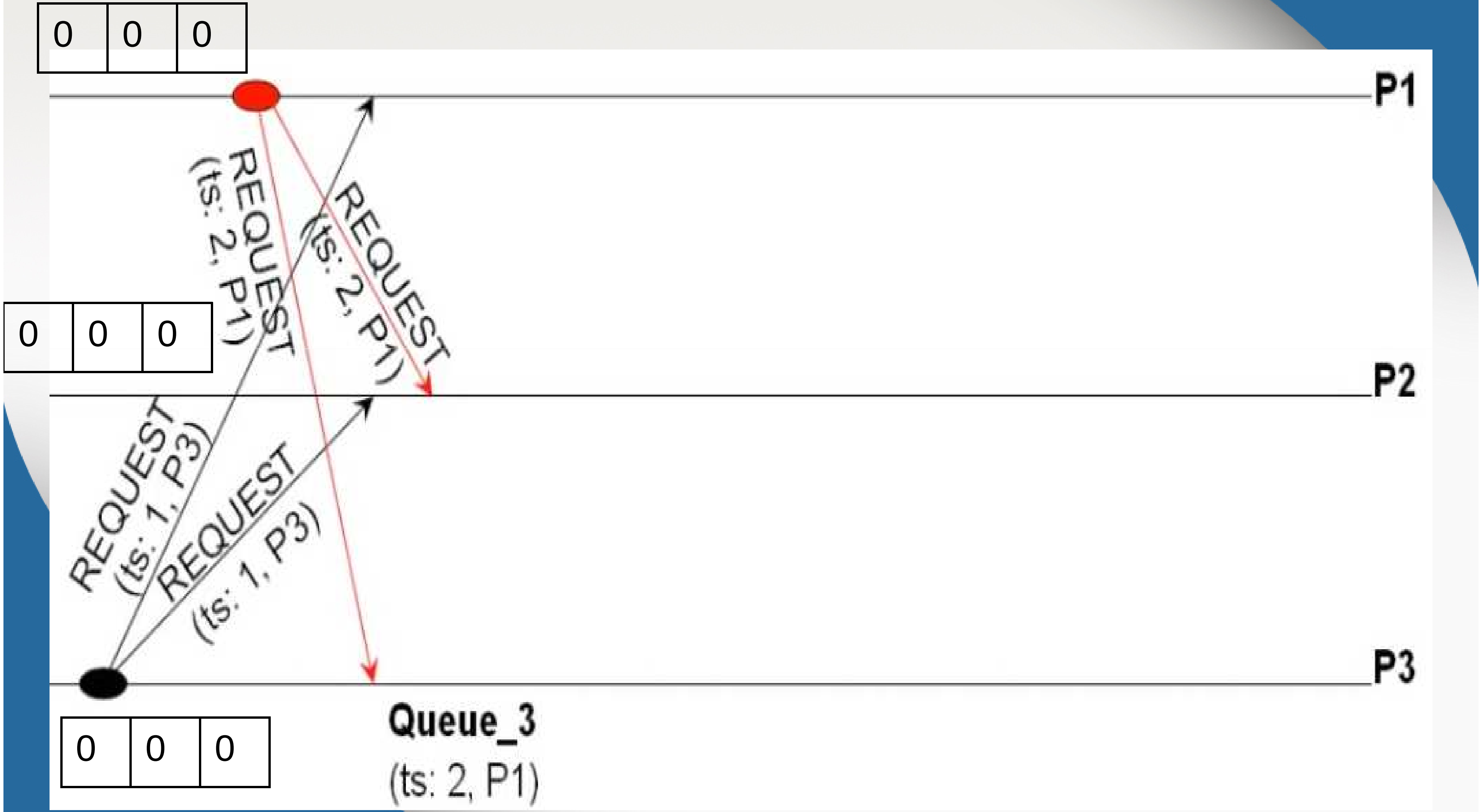
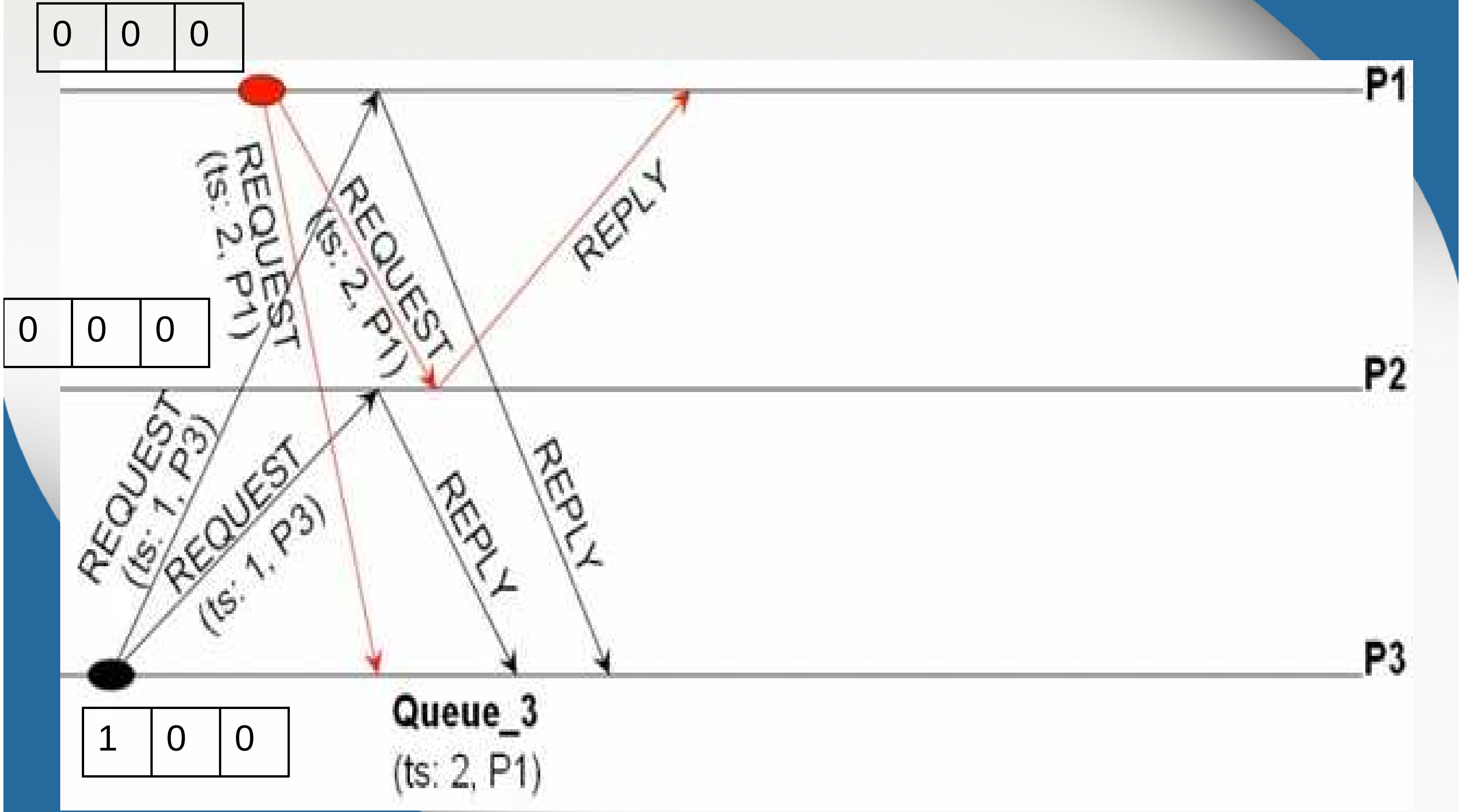
Ricart Agrawala's Distributed Mutual Exclusion Algorithm

Reference : Mukesh Singhal & N.G. Shivaratri, Advanced Concepts in Operating Systems, 5th Edition

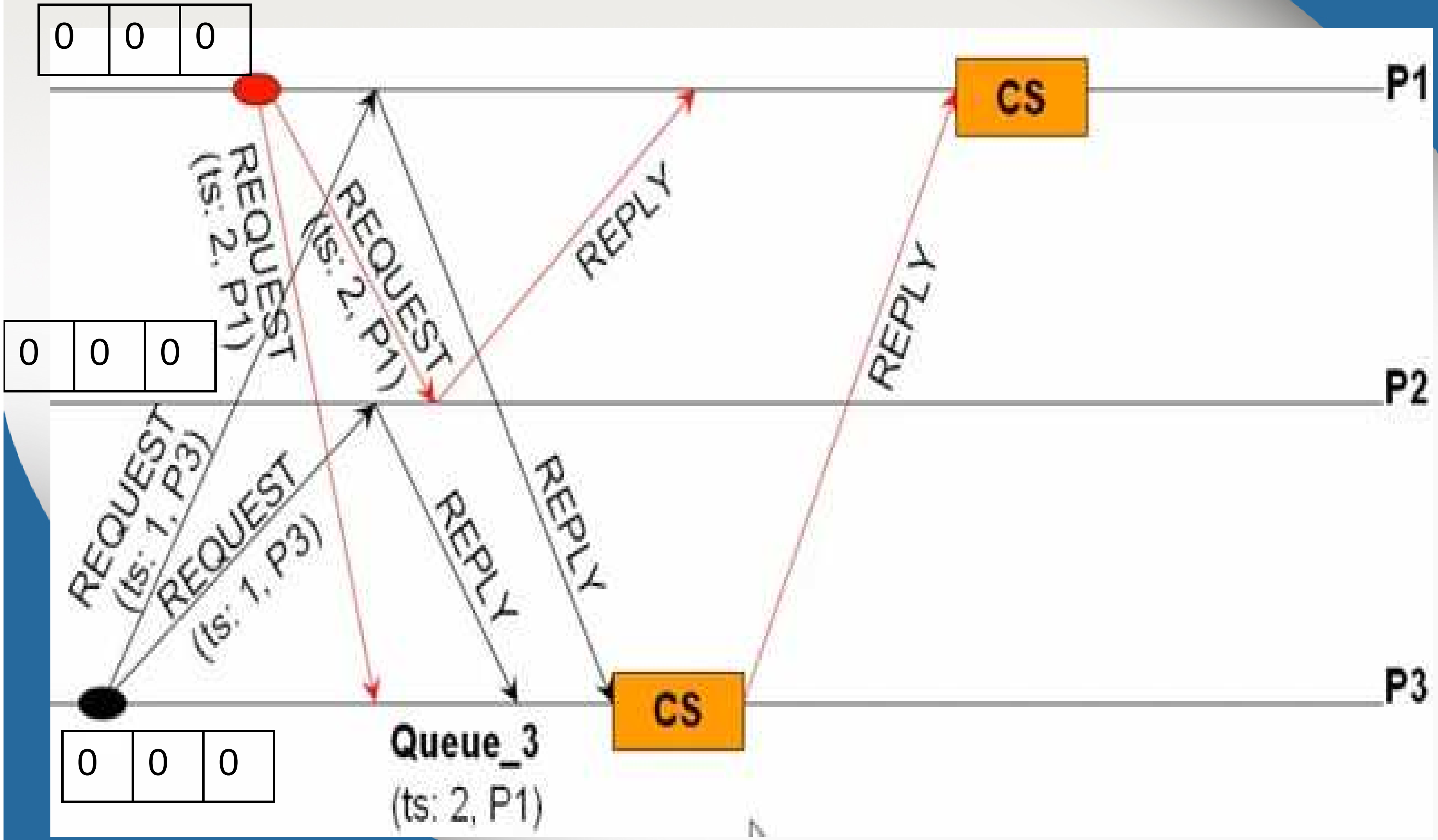
Ricart Agrawala's Distributed Mutual Exclusion Algorithm



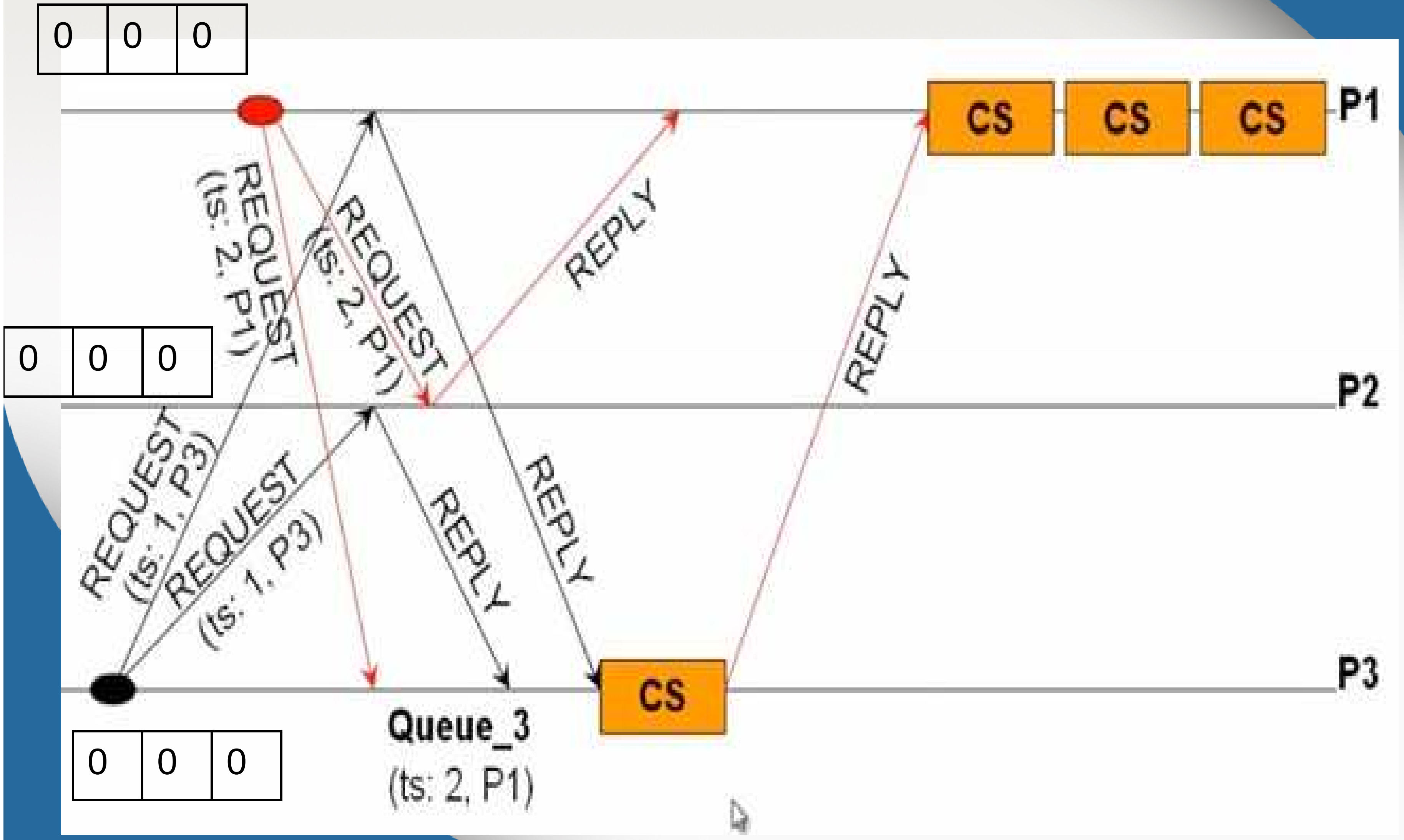
Ricart Agrawala’s Distributed Mutual Exclusion Algorithm



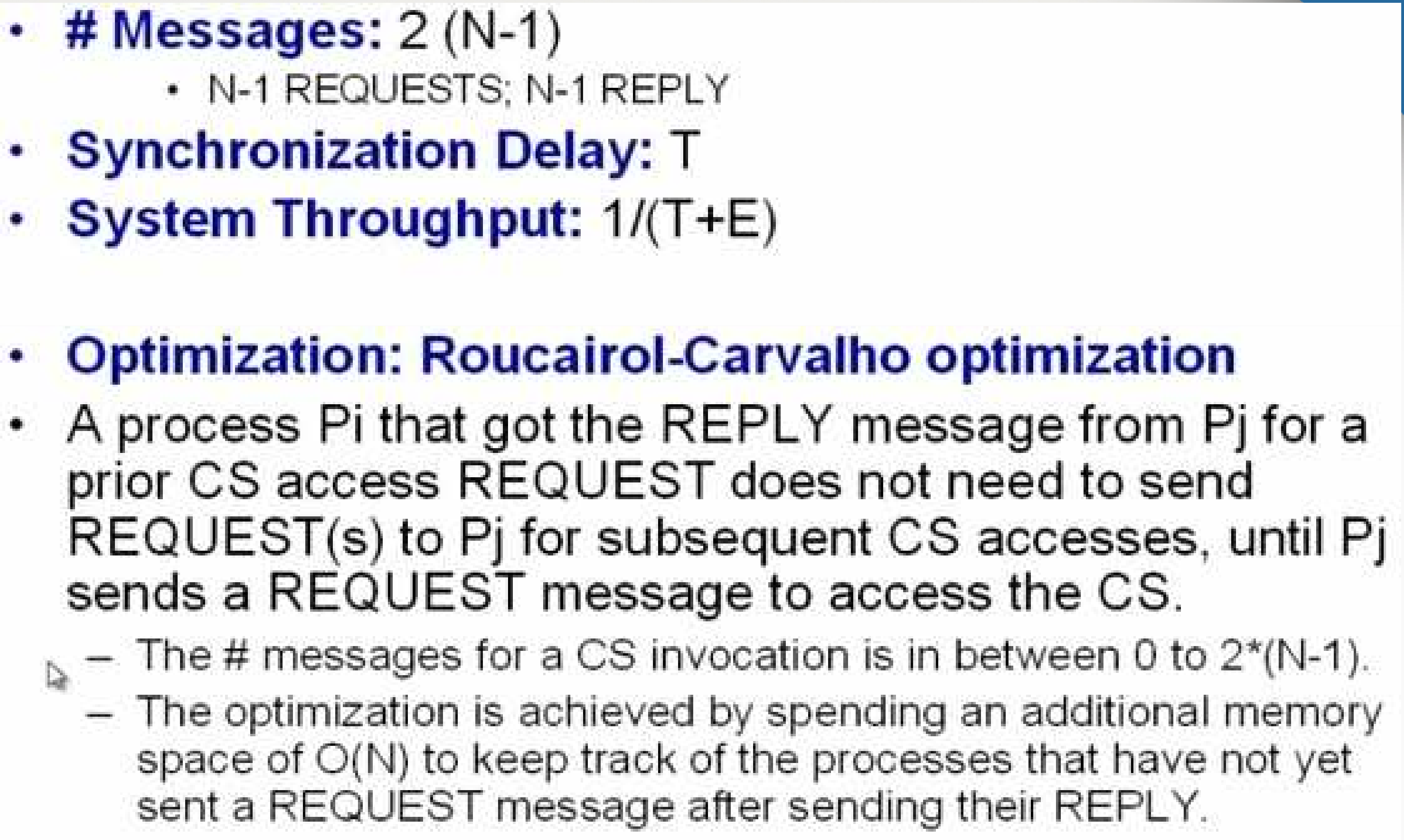
Ricart Agrawala's Distributed Mutual Exclusion Algorithm



Ricart Agrawala's Distributed Mutual Exclusion Algorithm



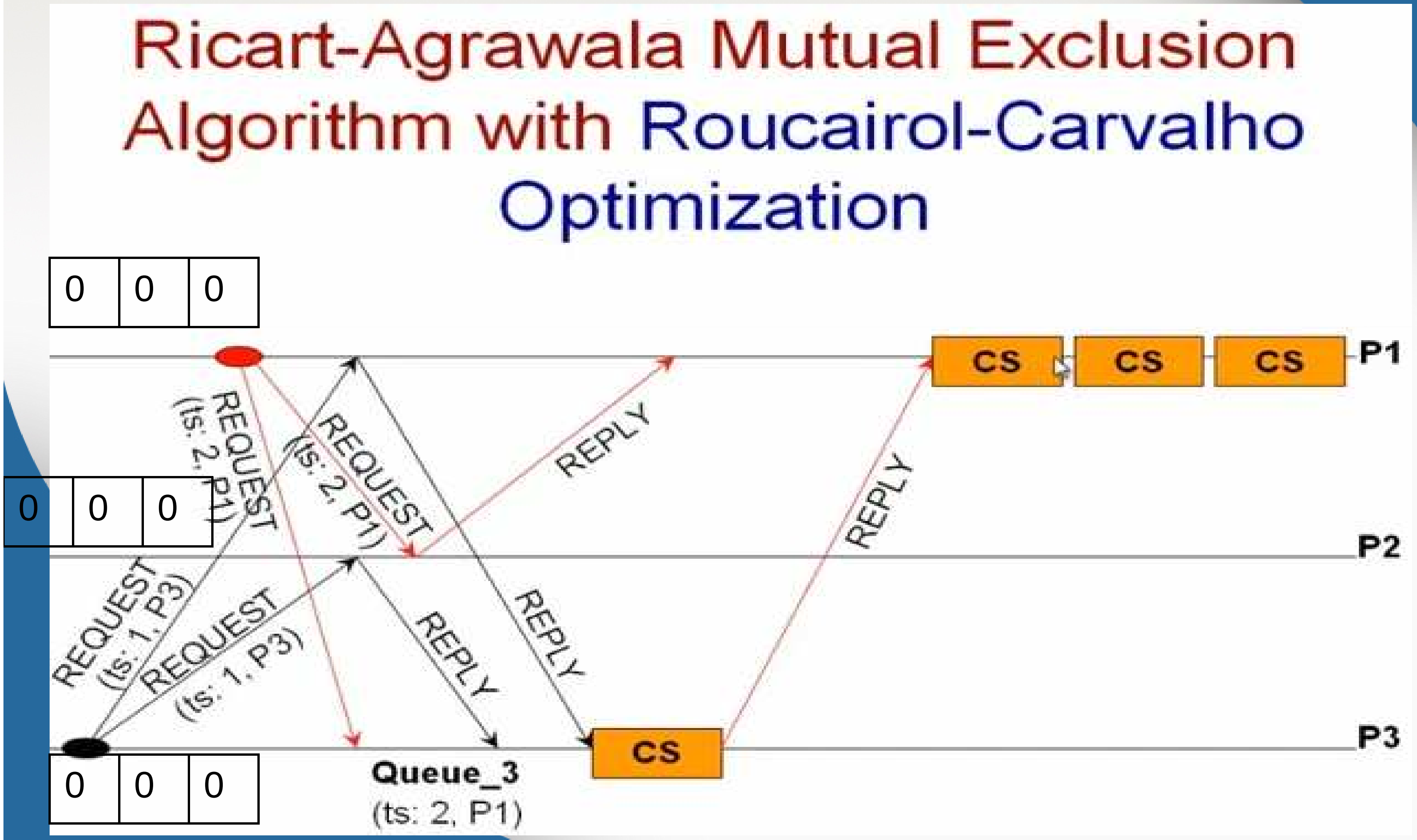
Ricart Agrawala's Distributed Mutual Exclusion Algorithm



- **# Messages:** 2 (N-1)
  - N-1 REQUESTS; N-1 REPLY
- **Synchronization Delay:** T
- **System Throughput:** 1/(T+E)

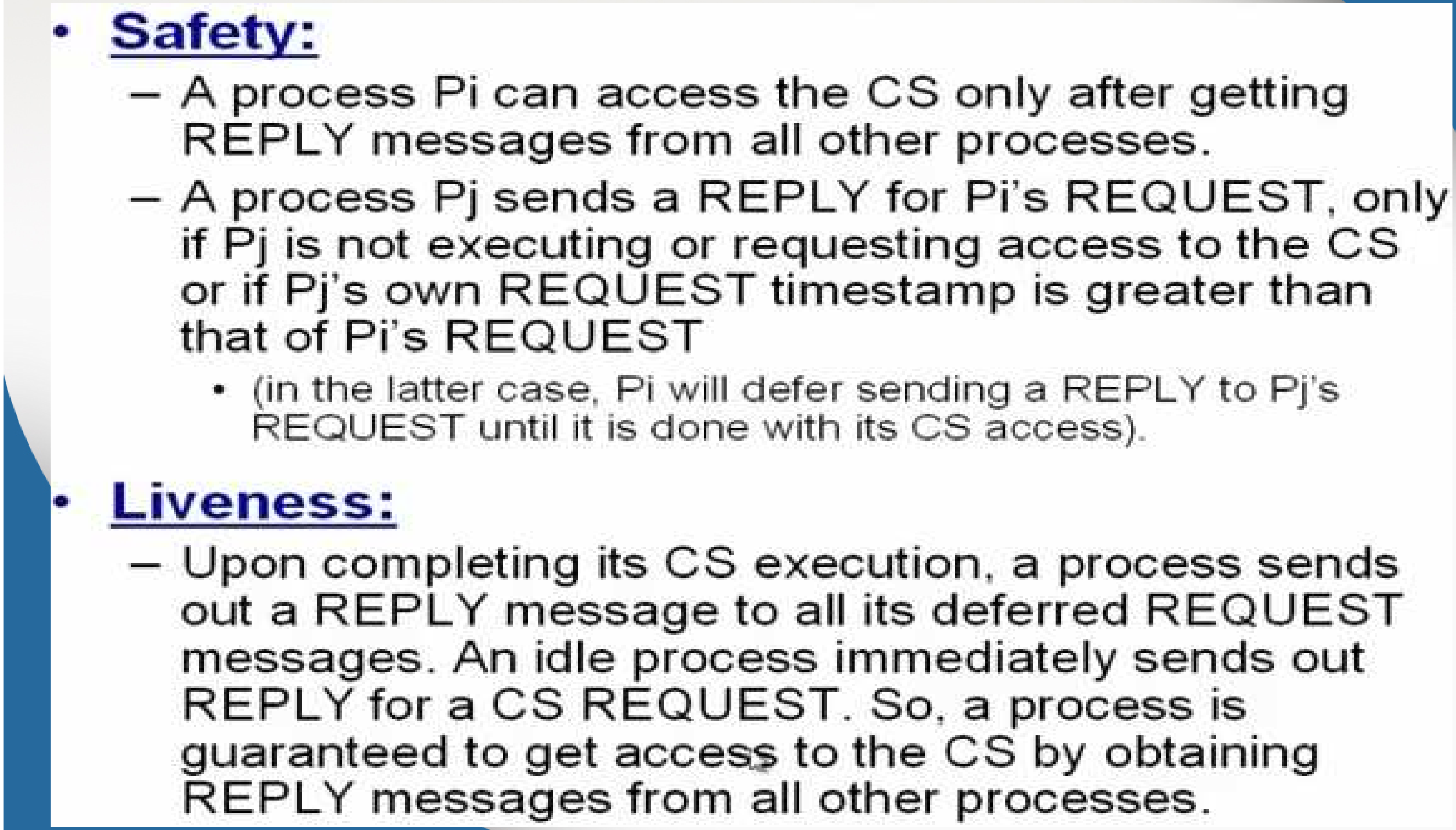
- **Optimization: Roucairol-Carvalho optimization**
- A process Pi that got the REPLY message from Pj for a prior CS access REQUEST does not need to send REQUEST(s) to Pj for subsequent CS accesses, until Pj sends a REQUEST message to access the CS.
  - The # messages for a CS invocation is in between 0 to 2*(N-1).
  - The optimization is achieved by spending an additional memory space of O(N) to keep track of the processes that have not yet sent a REQUEST message after sending their REPLY.

Ricart Agrawala’s Distributed Mutual Exclusion Algorithm



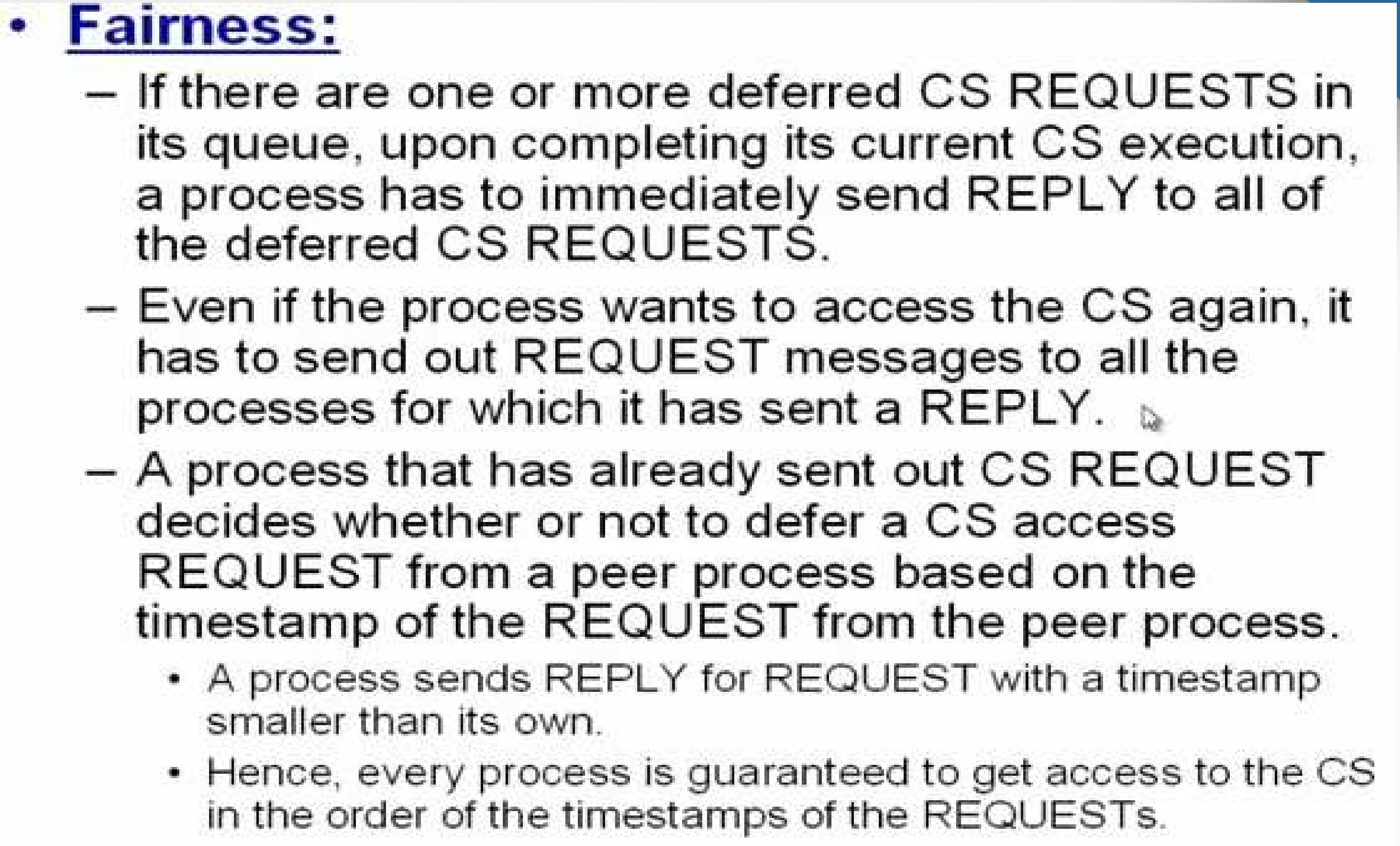
Ricart-Agrawala Mutual Exclusion Algorithm with Roucairol-Carvalho Optimization

Ricart Agrawala's Distributed Mutual Exclusion Algorithm



- **Safety:**
  - A process Pi can access the CS only after getting REPLY messages from all other processes.
  - A process Pj sends a REPLY for Pi's REQUEST, only if Pj is not executing or requesting access to the CS or if Pj's own REQUEST timestamp is greater than that of Pi's REQUEST
    - (in the latter case, Pi will defer sending a REPLY to Pj's REQUEST until it is done with its CS access).
- **Liveness:**
  - Upon completing its CS execution, a process sends out a REPLY message to all its deferred REQUEST messages. An idle process immediately sends out REPLY for a CS REQUEST. So, a process is guaranteed to get access to the CS by obtaining REPLY messages from all other processes.

Ricart Agrawala's Distributed Mutual Exclusion Algorithm



- **Fairness:**
  - If there are one or more deferred CS REQUESTS in its queue, upon completing its current CS execution, a process has to immediately send REPLY to all of the deferred CS REQUESTS.
  - Even if the process wants to access the CS again, it has to send out REQUEST messages to all the processes for which it has sent a REPLY.
  - A process that has already sent out CS REQUEST decides whether or not to defer a CS access REQUEST from a peer process based on the timestamp of the REQUEST from the peer process.
    - A process sends REPLY for REQUEST with a timestamp smaller than its own.
    - Hence, every process is guaranteed to get access to the CS in the order of the timestamps of the REQUESTs.

Source

https://www.youtube.com/watch?v=r7SJOhGF4Nc

THANK YOU