Maekawa 2.0 (Sanders Sec. 5)

Request are timestamped with Totally-Ordered Lamport Timestamps

Message types:
REQUEST, GRANT, RELEASE
FAIL, INQUIRE, YIELD  *(process can now take back a GRANT if it has received YIELD)*

**State:**

- reqQ: priority queue, ordered by timestamps
- lockHolder: who \( p_i \) has sent GRANT to by not yet received RELEASE from
- lockTS: timestamp of lockHolder’s request
* also keep track of messages that have been received

**Initially:**

- reqQ is empty
- lockHolder = null
- lockTS = null

**When \( p_i \) wants to request resource:**

Send \( \text{REQUEST}(i, TS_i) \) to all processes in \( S_i \) *(self included)*
Wait for \( \text{GRANT} \) msgs from all in \( S_i \)
Access Resource

**When \( p_i \) is done with resource:**

Send \( \text{RELEASE} \) to all processes in \( S_i \) *(self included)*

**When \( p_j \) receives \( \text{REQUEST}(i, TS_i) \) from \( p_i \):**

- if lockHolder = null
  
  Send \( \text{GRANT} \) to \( p_i \)
  
  lockHolder = \( p_i \)
  
  lockTS = \( TS_i \)

- else

  Put \( \text{(REQUEST}_i, TS_i) \) in reqQ
  
  if lockTS < \( TS_i \)
  
  Send \( \text{FAIL} \) to \( p_i \) *(you don’t stand a chance of getting resource right now so you should YIELD – give up locks)*

  **else

  Send \( \text{INQUIRE} \) to lockHolder (see if it will give up the lock for \( p_i \))
  
  Send \( \text{FAIL} \) to all processes in reqQ with timestamps > \( TS_i \) ***(you are later in reqQ, so you should YIELD – give up locks)***
When \( p_i \) receives \textbf{INQUIRE} from \( p_j \): (should I release the lock I am holding for you?)

If \( p_i \) has received \textbf{FAIL} from any process

or if it has sent \textbf{YIELD} to any process and not yet received a new \textbf{GRANT},

send \textbf{YIELD} to \( p_j \) (I give up my lock)

\textbf{When \( p_j \) receives \textbf{YIELD} from \( p_i \):}

Add \((\text{lockHolder}, \text{lockTS})\) to \( \text{reqQ} \) (put locked process back in queue)

lockHolder = null
lockTS = null

if queue ≠ empty

\((\text{lockHolder}, \text{lockTS}) = \text{dequeue}(\text{reqQ})\)

send \textbf{GRANT} to \text{lockHolder}

\textbf{When \( p_j \) receives \textbf{RELEASE} from \( p_i \):}

lockHolder = null
lockTS = null

if reqQ ≠ empty

\((\text{lockHolder}, \text{lockTS}) = \text{dequeue}(\text{reqQ})\)

send \textbf{GRANT} to \text{lockHolder}

** Missing in Maekawa paper: may lead to deadlock**