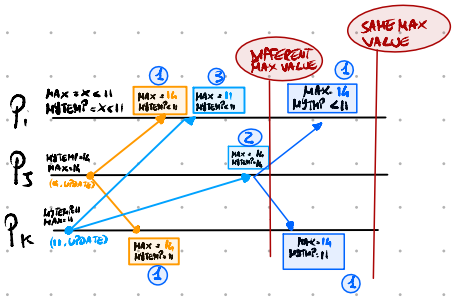
HOMEWORK 4 - LEONARDO SERILLI (UBINGT)

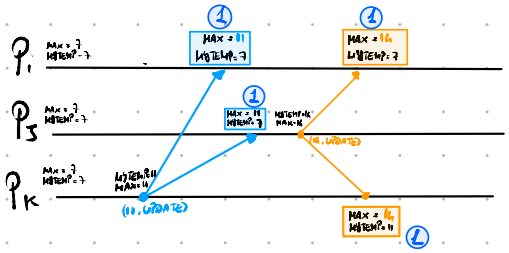
① It's impossible to run LE algorithm in ASYNCHRONOUS Networks with unknown topology.
Because of the NON-UNIFORM configuration: Processes doesn't know the total number of nodes ; A process won't know how many messages to wait in order to compute wich is the leader (the one wich measured the highest temperature). We have also the problem that, if the leader leaves the network during the election, he won't notify all processes about the max temperature, to avoid this a timeout must be implemented, but because of asynchronority, this will add a non negligeble overhead to the complexity. The last problem with LE is the fact that we must use the measured temperatures as IDs, and of course, more than one process can have the same ID.

② The problem appears in the code block written on the right.
Suppose P_i have MYTEMP < 11.
If P_i receives (14, 'UPDATE') from P_5 before (11, 'UPDATE') from P_k, by the 1st statement it will set MAX = 14, then by the 3rd statement P_i will set MAX = 11.
In confusion P_i, P_5 and P_k will know different value: $MAX_i = 11$ AND $MAX_5 = MAX_k = 14$.
Then thanks to the 2nd statement P_5 will broadcast (14, 'UPDATE') setting all MAX equal.



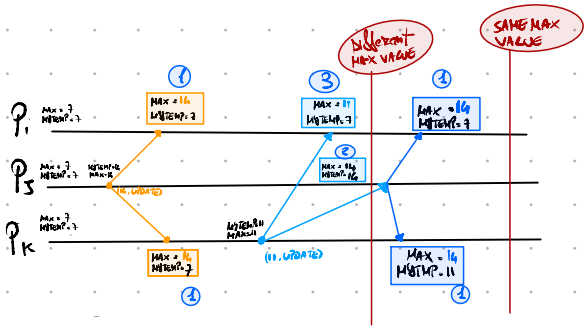
{ A MSG J ARRIVES, TAGGED WITH "UPDATE" }
1. IF (J > MAX)
   MAX = J ;
2. ELSE IF (J ≤ MAX AND MYTEMP > J)
   MAX = MYTEMP
   BROADCAST (MAX, UPDATE)
3. ELSE (J ≤ MAX AND MYTEMP ≤ J)
   MAX = J

③ CASE 11 THEN 14 : In the first case the max temperature is correctely set by all processor, we won't end up in the situation described in the question ②, because P_i, thanks to the TOTAL ORDERING PROPERTY, will set MAX to 11 and then to 14.

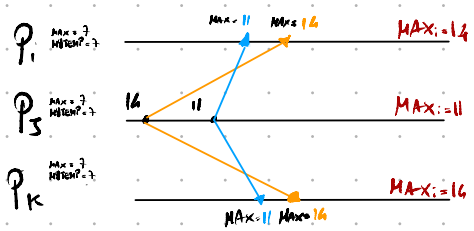


{ A MSG J ARRIVES, TAGGED WITH "UPDATE" }
1. IF (J > MAX)
   MAX = J ;
2. ELSE IF (J ≤ MAX AND MYTEMP > J)
   MAX = MYTEMP
   BROADCAST (MAX, UPDATE)
3. ELSE (J ≤ MAX AND MYTEMP ≤ J)
   MAX = J

CASE 14 THEN 11 . We have the same situation described in 'QUESTION ②'



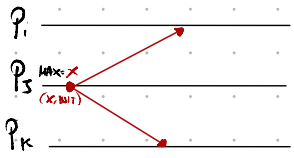
④ At the end we will have only one process knowing the right RIGHT MAX temperature.



· To avoid it we must assume tha the sending order is the same of the delivery (if P_5 broadcast 14 then 11, P_i must receive 14 then 11)

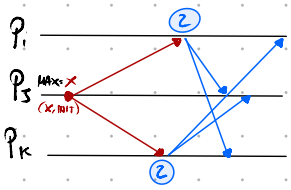
(5) We can easily remove the 'INIT TAG' and treat it as an 'UPDATE' message.
The only difference is that it can trigger the 2nd Statement generating another broadcast, increasing in this case the message complexity.
In the worst case it would trigger a $(n-1)$ broadcast, each one generating $(n-1)$-messagese, so the message complexity is increased of $O((n-1)(n-1)) = O(n^2)$ messages.

with INIT TAG



{ A MSG J ARRIVES, TAGGED WITH "UPDATE" }
IF (J > MAX)
MAX = J;

Without INIT TAG



{ A MSG J ARRIVES, TAGGED WITH "UPDATE" }
1. IF (J > MAX)
   MAX = J;
2. ELSE IF (J ≤ MAX AND MYTEMP > J)
   MAX = MYTEMP
   BROADCAST (MAX, UPDATE)
3. ELSE (J ≤ MAX AND MYTEMP ≤ J)
   MAX = J