

HOMEWORK 4 - LEONARDO SERILLI (UBINGT)

① It's impossible to run LE algorithm in ASYNCHRONOUS Networks with unknown topology.

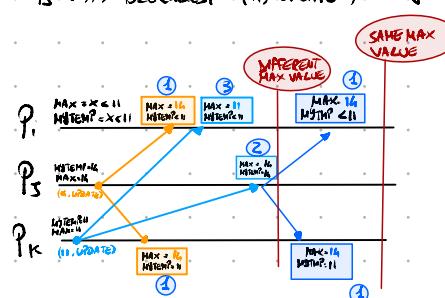
Because of the NON-UNIFORM configuration: Processes doesn't know the total number of nodes; A process won't know how many messages to wait in order to compute which is the leader (the one who measured the highest temperature). We have also the problem that, if the leader leaves the network during the election, he won't notify all processes about the max temperature, to avoid this a timeout must be implemented, but because of asynchronicity, this will add a non-negligible 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 $(i, UPDATE)$ from P_3 before $(i, 'UPDATE')$ from P_k , by the 1st statement it will set $MAX = 14$, then by the 3rd statement P_i will set $MAX = 11$.

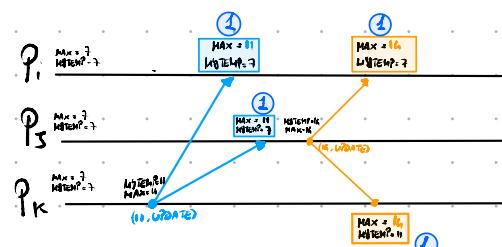
In conclusion P_i , P_3 and P_k will know different value: $MAX_i = 11$ AND $MAX_3 = MAX_k = 14$. Then thanks to the 2nd statement P_3 will broadcast $(i, 'UPDATE')$ setting all MAX equal.



{ A MSG IS ARRIVED, TACKLED WITH "UPDATE" }

- ① IF ($J > MAX$)
 $MAX = J$;
- ② ELSE IF ($J \leq MAX$ AND $MYTEMP > J$)
 $MAX = MYTEMP$
 BROADCAST ($MAX, UPDATE$)
- ③ ELSE ($J \leq MAX$ AND $MYTEMP \leq J$)
 $MAX = J$

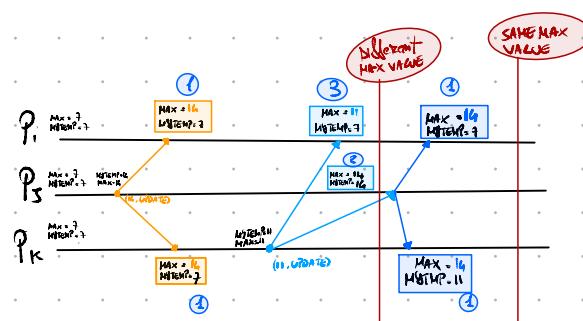
③ CASE 11 THEN 14 : In the first case the max temperature is correctly set by all processor, we won't end up in the situation described in the question ②, because P_i , thanks to the TOTAL ORDERING PROPERTY, will set MAX to 11 and then to 14.



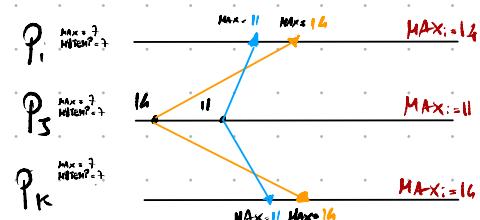
{ A MSG IS ARRIVED, TACKLED WITH "UPDATE" }

- ① IF ($J > MAX$)
 $MAX = J$;
- ② ELSE IF ($J \leq MAX$ AND $MYTEMP > J$)
 $MAX = MYTEMP$
 BROADCAST ($MAX, UPDATE$)
- ③ ELSE ($J \leq MAX$ AND $MYTEMP \leq J$)
 $MAX = J$

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 MAX temperature.



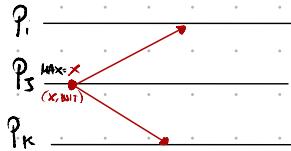
To avoid it we must assume that the sending order is the same of the deliverying (if P_3 broadcast 14 then 11, P_i must receive 14 then 11)

5 We can easily remove the 'INIT TAG' and treat it as an 'UPDATE' message.

The only difference is that it can trigger the End Statement generating another broadcast, increasing in this case the message complexity.

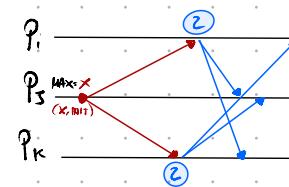
In the worst case, it would trigger a $(n-1)$ broadcast each one generating $(n-1)$ -messages, so the message complexity is increased of $O((n-1)(n-1)) = O(n^2)$ messages.

with INIT TAG



{ A MSG s RECEIVES, TAGGED WITH "UPDATE"
IF ($s > \text{MAX}$)
 $\text{MAX} = s$;

without INIT TAG



{ A MSG s ARRIVES, TAGGED WITH "UPDATE"
① IF ($s > \text{MAX}$)
 $\text{MAX} = s$;
② ELSE IF ($s \leq \text{MAX}$ AND $\text{MYTEMP} > s$)
 $\text{MAX} = \text{MYTEMP}$
BROADCAST(MAX , UPDATE)
③ ELSE ($s \leq \text{MAX}$ AND $\text{MYTEMP} \leq s$)
 $\text{MAX} = s$