Leader election in undirected ring - stages with feedback

Our goal is to elect leader in a undirected ring with n nodes an d asynchronous first-in-first-out communication. We assume that every node has a unique identifier, and that all identifiers are totally ordered. Initially, no node needs any knowledge about the identities of the nodes at the other ends of its incident edges. Each node recognizes the directions of its edges, but there is no global sense of orientation in the ring.

Here we describe algorithm presented by Rotem et al. [1987], which achieves the above while using at most $3n \log_3 n + O(n)$ messages in the worst case.

The algorithm

The algorithm will consist of electoral rounds.
We will distinguish 4 states a node can be in:
passive - node does not participate in election anymore and only passes messages
(node that is not "*passive*" will be also regarded as "*active*")
E-candidate - node is taking part in this round election
A-candidate - node awaits feedback from its active neighbours
Leader - node is the leader

Messages sent will consist of pairs of "type" and "value" where values will be from set of unique identifiers of nodes participating, while types can be either of *E-msg*, *A-msg* and *T-msg*.
By *left*, *right* and *both* we will denote directions in witch node sends or from which it receives a message.
By i we will denote node's unique identifier and by $n(i)$ node whose unique identifier is i.
Upon initialization each node starts with status *E-candidate*.

Each round starts with each *E-candidate* sending message (*E-msg*,i) in both of its directions.

Than, each *E-candidate* awaits to receive a message from both of its direction, lets denote them as $(E\text{-}msg, vl)$ recieved from *left* and $(E\text{-}msg, vr)$ received from *right*, let $y = max(i, vl, vr)$.
If $y > i$ then $n(i)$ will send $(A\text{-}msg, y)$ toward direction from witch message with value$=y$ was received.
Than regardless if any message was sent from $n(i)$, it will change it's state to *A-candidate*.
If $i = vl = vr$ than $n(i)$ is the only *E-candidate* in the ring, it should become *Leader* and send *T-msg* in one of it's directions.

Each *A-candidate*, awaits approval from both of it's active neighbours as such it waits for message $(A\text{-}msg, i)$ from both sides than it changes status to *E-candidate* to participate in the next round.
If *A-candidate* receives $(E\text{-}msg, i)$ it should treat it as negative feedback from an active neighbour, it than becomes *passive* and passes the received message.

passive nodes pass every message except for potential $(A\text{-}msg, i)$ being no longer needed.
If it receives $(T\text{-}msg, val)$ it learns that $n(val)$ is the leader, passes the message and ends algorithm.

Leader ends when it receives $(T\text{-}msg, i)$.

It shall be noted that receiving *T-msg* can be used to obtain common sense of direction in the ring.

Correctness

Let $j = max\{i|n(i)\text{participates in the algorithm}\}$, than from the voting and checking procedure follows that at each electoral round $n(j)$ will defeat both of it's immediate active neighbours; hence the number of *passive* nodes increases monotonically. At the same time $n(j)$ will never be defeated (receive negative feedback). This implies that at some round $n(j)$ will be the only active node and both of it's *E-msg*es will return to it making it the leader.
Having being elected $n(j)$ can send termination order *T-msg* around the ring along with it's identifier to notify other nodes of it's election.

Complexity analysis

By m_i let's denote set of nodes participating in i-th election (number of *E-candidates*).
Let $r_j(n(i))$ where $n(i) \in m_j$ the immediate neighbour to the left from $n(i)$ that $r_j(n(i) \in m_j$, symmetrically we define $l_j(n(i) \in m_j$.

Lemma 1. - $|m_j| \geq 3|m_{j+1}|\ \forall j$

If a node $n(i) \in m_j$ survives the j-th round than it means that it must have received positive acknowledgment from both $r_j(n(i)$ and $l_j(n(i)$. Therefore because of the voting procedure, a negative feedback will be received by $l_j^2(n(i)$, $l_j(n(i)$, $r_j(n(i)$ and $r_j^2(n(i)$. Therefore for each surviving node at least 2 were eliminated. Hence, at most one third of the processors in m_j survives the j-th stage, and the lemma is proved.

Lemma 2. - number of electoral rounds $\leq \lceil \log_3 n \rceil$

The proof follows from Lemma 1.

We can now notice that each round every 2 immediate active nodes will send between them at most 3 messages being, exchange of 2 *E-msg*es and no more than one *A-msg* from which follows;

$|\mathbf{Messages}| \leq 3n\lceil \log_3 n + O(n) \approx 1.89n \log_2 n + O(n)$

References

D. Rotem, E. Korach, and N. Santoro. Analysis of a distributed algorithm for extrema finding in a ring. *Journal of Parallel and Distributed Computing*, 4(6):575–591, 1987. ISSN 0743-7315. doi: https://doi.org/10.1016/0743-7315(87)90031-1. URL https://www.sciencedirect.com/science/article/pii/0743731587900311.