# Leader election in an undirected ring

Our goal is to elect a leader in an undirected ring with $n$ nodes and 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 an algorithm presented by Rotem et al. [1987], which achieves the above and uses at most $3n \log_3 n + O(n)$ messages in the worst case.

## The algorithm

### Definitions and assumptions

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

***E-candidate*** — node is taking part in this electoral round

***A-candidate*** — node awaits feedback from its active neighbours

***Leader*** — node is the leader

(node that is not "*passive*" will be also regarded as "*active*")

Messages sent will consist of pairs of "type" and "value". Values will be from the set of unique identifiers of the nodes participating, while type can be either *E-msg*, *A-msg* or *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 the node's unique identifier and by $n(i)$ the node whose unique identifier is $i$.

Upon initialization each node starts with status *E-candidate*.

### Procedures

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

Afterwards, each *E-candidate* awaits to receive a message from both of its directions.

Let's denote them as ($E\text{-}msg, vl$) received from *left* and ($E\text{-}msg, vr$) received from *right*.

Let $y = max(i, vl, vr)$.

If $y > i$, then $n(i)$ will send $(A\text{-}msg, y)$ to left if $y = vr$ or to right if $y = vl$. Then regardless if any message was sent from $n(i)$, it will change its state to *A-candidate*.

If $i = vl = vr$ then $n(i)$ is the only *E-candidate* in the ring. It should become the *Leader* and send *T-msg* in one of its directions.

Each *A-candidate* awaits approval from both of its active neighbours. Meaning, it waits for message $(A\text{-}msg, i)$ from both sides. Afterwards, it changes its status to *E-candidate* to participate in the next round.

If *A-candidate* receives $(E\text{-}msg, i)$, it should treat it as a negative feedback, it should become *passive* and pass the received message.

*passive* node passes every message, except for potential $(A\text{-}msg, i)$ which is no longer needed.

If it receives $(T\text{-}msg, val)$, it learns that $n(val)$ is the leader, passes the message and finishes.

*Leader* finishes 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 $j = max\{i | n(i) \text{ participates in the algorithm}\}$.

Then from the voting and checking procedure follows that at each electoral round $n(j)$ will defeat both of its 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 its *E-msg* messages will return to it, making it the leader.

Having been elected $n(j)$ can send a termination order *T-msg* around the ring along with its identifier to notify other nodes of its 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 $n(i) \in m_j$ be the immediate neighbour to the left of $n(i)$ who is still active in $j$-th round, symmetrically we define $l_j(n(i)$.

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

*Proof.* If a node $n(i) \in m_j$ survives the $j$-th round, then 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, negative feedback will be received by $l_j^2(n(i)$, $l_j(n(i)$, $r_j(n(i)$ and $r_j^2(n(i)$. Thus, for each surviving node, at least 2 were eliminated. Hence, at most a third of the processors from $m_j$ survive the $j$-th round. □

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

*Proof.* The proof follows from Lemma 1. □

We can now notice that each round every 2 immediate active nodes will exchange at most 3 messages. That is, 2 *E-msg* messages and no more than one *A-msg*.

Hence, follows:

**Theorem 1.** $|Messages| \leq 3n\lceil \log_3 n + O(n) \approx 1.89n \log_2 n + O(n)$

## 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.