

1 Leader election in asynchronous undirected ring

The goal is to elect a leader in an undirected ring with n nodes, that communicate asynchronously. We assume that the processors are identical, except for unique identifiers. The size of the network is unknown. Each node has a list of its neighbours in an arbitrary order.

The most straightforward solution is to make every node send its identifier to both neighbours. By passing the identifiers around processors can choose the node with the highest identifier to be the leader. This algorithm requires only $O(n)$ rounds, but it needs $O(n^2)$ messages to be sent. Here we analyze an algorithm presented in 1980 by Hirschberg and Sinclair [1], which only requires $O(n \log n)$ messages.

2 The algorithm

The idea is that nodes initiate messages that are passed around in both directions along paths of predetermined length (which are successive powers of 2). At the beginning every processor sends a message containing its identifier, length 1 and an information that the message is going out. Then after every received message there are six main cases:

1. If the identifier in the message is higher than the processor's identifier:
 - (a) If the message already visited the predetermined number of nodes, then we send it back, with the information that it's coming back.
 - (b) If the message has not yet visited the predetermined number of nodes or is already coming back, then we pass it to the next processor lowering its length by 1.
2. If the identifier in the message is lower than the processor's identifier, then the message is ignored.
3. If the identifier in the message is equal to the processor's identifier and the message is going out, then we win the election and now need to send an ending message to both neighbours.
4. If the identifier in the message is equal to the processor's identifier and the message is coming back and it is first come back message, then we do nothing.
5. If the identifier in the message is equal to the processor's identifier and the message is coming back and it is second come back message then we initiate a new message with double the length of the previously initiated message and send it to both neighbours.
6. If the message is an end message, we pass it to the next processor.

3 Correctness

Let v be the node with the highest identifier. Any node that receives a message from v must have a lower identifier. That means it will have to pass this message further or send it back to the source. It follows that every message with a length lower than n will come back to v , so after $\lceil \log n \rceil$ initiated messages v will send one with a length at least n . This message will visit v again without being sent back at any point, which means v will be declared the leader.

On the other hand for any node $w \neq v$ it is impossible to be declared the leader, because there exists $k < n$ such that a message with length k sent from w will visit v , where it will be ignored. Every such node will be relaying messages until it receives a stop message originating from the leader.

4 Complexity analysis

Theorem 1. *Number of messages is $O(n \log n)$.*

Proof. A processor initiates a message with length 2^i only if it is not defeated by a processor within distance 2^{i-1} , so within any group of $2^{i-1} + 1$ processors only one will initiate a message with length 2^i . As a result of that initiation at most $4 \cdot 2^i$ messages will be passed, because every message is sent in two directions and will either travel 2^i edges and return or will be ignored within the first 2^i steps. No message will have a length higher than $2^{\lceil \log n \rceil}$ since, once a processor initiates paths of at least n length and the message is acceptable all the way around the circle, the processor wins and stops initiating messages. So the total number of messages is bounded by:

$$4(1 \cdot n + 2 \cdot \lceil \frac{n}{2} \rceil + 4 \cdot \lceil \frac{n}{3} \rceil + \dots + 2^i \cdot \lceil \frac{n}{2^{i-1} + 1} \rceil + \dots + 2^{\lceil \log n \rceil} \cdot \lceil \frac{n}{2^{\lceil \log n \rceil - 1} + 1} \rceil)$$

Each of the $1 + \lceil \log n \rceil$ terms is bounded by $2n$, so the number of messages is not greater than $8(n + n \log n) \in O(n \log n)$. \square

References

- [1] D. S. Hirschberg, J. B. Sinclair, “Decentralized Extrema-Finding in Circular Configurations of Processors”, *Commun. ACM* **1980**, 23, 627–628, <https://doi.org/10.1145/359024.359029>.