

Higham-Przytycka algorithm for leader election in directed rings

The goal of the algorithm is, given a directed ring of n machines with unique ids , to select a single machine as a leader and make all other machines agree as to who the leader is.

BASIC algorithm

We will begin by analyzing a simpler version of the algorithm.

Each message will consist of two fields. rnd - specifying the round number, and id - a unique label. The only state that nodes will keep track of is what the previous message sent by them looked like ($node.id$ and $node.rnd$). If a message received by a node looks the same as the message previously sent by it, then the node will be able to safely assume that it is now the leader. At the beginning each node sends a message with $rnd = 0$ and id set to its unique identifier.

If a node receives a message with different rnd than its own then it simply forwards the message.

If a node receives a message with the same rnd as its own and rnd is odd then if $message.id < node.id$ the message is destroyed (nothing is sent to the next node), otherwise $message.rnd$ is bumped by 1 and message gets passed to the next node (we will call this action *promotion*).

If a node receives a message with the same rnd as its own and rnd is even then if $message.id > node.id$ the message is destroyed, otherwise the message gets promoted.

Correctness

For the sake of simplicity we can assume that messages are processed in an order such that all alive messages have the same rnd . Correctness of the algorithm stems from the following facts:

The algorithm never deletes all messages — In odd rounds the message with highest id will never be destroyed. In even rounds the message with lowest id will never be destroyed.

In every round at least one message gets deleted — In odd rounds the message with lowest id will be destroyed. In even rounds the message with highest id will be destroyed.

This means that at some point only one message will remain and it will return back to its sender unchanged. The sender then becomes a leader.

Improvements

Note that in BASIC there are exactly n messages sent in each round (each message has to travel to the next sender on the ring before being destroyed or promoted). This is not necessary for the correctness of the algorithm. A message could get promoted before reaching the next sender on the ring (the node promoting the message is then considered its sender in the next round). The improved algorithm makes heavy use of that fact.

Promotion by witness

The first improvement is to promote a message if it can be verified that the message would get promoted by BASIC further along the way. Suppose a node encounters a message such that $message.rnd = node.rnd + 1$ and $message.rnd$ is even. Then in previous round the node has sent a message with $id = node.id$. The previous message got either promoted or destroyed by some node z . If it got promoted then $z.id = node.id$. If it got destroyed, then $node.id < z.id$ and $node.rnd = z.rnd$. We can repeat this reasoning with the message sent by z , finding $z2$ that either promoted or destroyed the message such that $z.id \leq z2.id$. After repeating this multiple times we will always reach a first node w that promoted a message in previous round. We know that $node.id \leq w.id$ and $w.rnd = node.rnd + 1$, so if $message.id < node.id$ we can assume that it will get promoted at w anyway, so the node can promote it now.

Promotion by distance

The second improvement involves adding a *counter* field to the message. Each time the message gets promoted to round i , *counter* is set to F_{i+2} where F_t is the t th Fibonacci number. Each time a message is forwarded *counter* is decreased by 1. If *counter* reaches 0 then the message is promoted without checking any other conditions. This could potentially allow a round to pass without destroying any messages, however counters quickly become large enough to allow messages to traverse a full ring. So the algorithm remains correct.

ELECT algorithm

The final algorithm (named ELECT in the original paper) consists of BASIC augmented with promotion by distance in odd numbered rounds and promotion by witness in even numbered rounds. It is correct since it only destroys messages that BASIC would destroy and the number of messages decreases to 1 (promotion by distance becomes insignificant after logarithmic number of rounds and promotion by witness behaves the same way as BASIC).

Complexity

Let $\text{host}_i(a)$ denote the node that promoted the message with label a to round i .

Let $\text{destroyer}_i(a)$ denote the node that eliminated the message with label a in round i .

Let $\delta(x, y)$ denote the distance between nodes x and y in the direction of the ring.

Lemma 1. *If message with label a reaches round $i + 1$, and i is odd, then $\delta(\text{host}_i(a), \text{host}_{i+1}(a)) \geq F_{i+1}$*

Lemma 2. *If message with label a is destroyed in round i , and i is odd, then $\delta(\text{host}_i(a), \text{destroyer}_i(a)) \geq F_i$*

Proof. The proof of these lemmas is quite involved, so I will only show the main ideas.

We begin by induction on odd i , assuming that both lemmas are true for $i - 2$.

A node that promoted a message to round i also promoted some message to round $i - 1$.

For a message b that is promoted to round i , $\delta(\text{host}_{i-2}(b), \text{host}_i(b)) \geq F_i$. If b travelled less than F_i steps in round $i - 2$ then it was promoted by some $\text{host}_{i-1}(b) = \text{host}_{i-2}(c)$ and $c < b$. By induction $\delta(\text{host}_{i-2}(b), \text{host}_{i-1}(b)) \geq F_{i-1}$. In order to be promoted, b must reach some node with label $d > b$. $d > b > c$ so $d \neq c$ therefore c had to be destroyed between b and d . Hence $\delta(\text{host}_{i-1}(b), \text{host}_i(b)) \geq \delta(\text{host}_{i-2}(c), \text{destroyer}_{i-2}(c)) \geq F_{i-2}$ by induction. The distance is therefore at least $F_{i-1} + F_{i-2} = F_i$.

Let a and b be labels of two consecutive messages in round i such that b travels in front of a .

We can now prove Lemma 1. Assume a survives round i . If a was promoted by distance, then it travelled F_{i+2} steps and Lemma 1 holds. Otherwise a travels to $\text{host}_i(b)$ and is promoted ($a < b$). Since $i - 1$ is even, $\text{host}_{i-1}(a) = \text{host}_{i-2}(g)$ for some $g > a$. Consider first processor (that is a host for round $i - 1$) $\text{host}_{i-1}(h)$ after $\text{host}_{i-2}(g)$. $h \geq g$. So $h \neq b$. Thus

$$\begin{aligned} \delta(\text{host}_i(a), \text{host}_{i+1}(a)) &= \delta(\text{host}_{i-2}(g), \text{host}_{i-1}(h)) + \delta(\text{host}_{i-1}(h), \text{host}_i(b)) \\ &\geq \delta(\text{host}_{i-2}(h), \text{host}_{i-1}(h)) + \delta(\text{host}_{i-2}(b), \text{host}_i(b)) \\ &\geq F_{i-1} + F_i = F_{i+1} \end{aligned}$$

Now let us prove Lemma 2. Suppose a is eliminated in round i . Since all eliminations happen the same way as in BASIC, $a < b$ and a has been eliminated by $\text{host}_i(b)$. a must have been promoted to round i by $\text{host}_{i-2}(b)$. Thus

$$\begin{aligned} \delta(\text{host}_i(a), \text{destroyer}_i(a)) &= \delta(\text{host}_i(a), \text{host}_i(b)) \\ &\geq \delta(\text{host}_{i-2}(b), \text{host}_i(b)) \\ &\geq F_i \end{aligned}$$

□

Lemma 3. *Every message promoted in round i where i is even travels at least F_i steps less than it would in BASIC.*

Proof. Let a be a label of surviving message, let b be the first message in front of a . $a < b$ because a survives an even round. If not promoted beforehand, a will be promoted by witness at $\text{host}_{i-1}(b)$ (the node that promoted b will have $id = b > a$ and $rnd = i$), so it will stop at least $\delta(\text{host}_{i-1}(b), \text{host}_i(b)) \geq F_i$ (Lemma 1) steps before $\text{host}_i(b)$. \square

Theorem 1. *Messages in ELECT traverse fewer than $1.271n \log(n) + O(n)$ edges in total.*

Proof. Every message in round i where i is odd travels at least F_i steps and no two messages travel through the same edge, so number of messages alive in round i is at most n/F_i . Therefore ELECT uses at most $F^{-1}(n) + O(1)$ rounds.

Consider even round i followed by odd round $i + 1$. If there are x messages in round $i + 1$ then number of steps travelled by messages in round i is at most $n - xF_i$ (Lemma 3). In round $i + 1$ each envelope travels at most F_{i+1+2} steps (promotion by distance), therefore number of steps in round $i + 1$ is at most $\min(xF_{i+3}, n)$.

Total number of steps in rounds i and $i + 1$ is bounded above by $\min(n + x(F_{i+3} - F_i), 2n - xF_i) \leq 2n - nF_i/F_{i+3} \leq n(2 - \phi^{-3} + \phi^{-2i}) = n(4 - \sqrt{5} + \phi^{-2i})$.

Hence the total number of steps is bounded by

$$\sum_{\text{even}(i), 2 \leq i \leq F^{-1}(n)} n(4 - \sqrt{5} + \phi^{-2i}) + O(n) = \frac{4 - \sqrt{5}}{2} n \log_{\phi} n + O(n) < 1.271n \log n + O(n)$$

\square

References

- [1] Lisa Higham, Teresa Przytycka (1995) *A simple, efficient algorithm for maximum finding on rings*