Itai Rodeh Leader Election Algorithm

Oskar Krygier

December 2024

1 Introduction

Itai Rodeh is a probabilistic algorithm for leader election. We consider asynchronous, directed ring, where each process is anonymous, but the size of the ring is known. The model must be reliable and buffers between processes should be FIFO.

2 The algorithm

The algorithm consists of two phases:

1. *Selection phase*
   Each active process flips a coin and sends it to the next active process. If the process receive 1 while being 0, it becomes inactive. We repeat this phase at most $c = 5 \log n$

2. *Verification phase*
   Each active process sends a counter towards the cycle to check whether it is the only active process. If the process receives the message with counter equal to the size of the cycle, it becomes a leader.

2.1 Implementation

An implementation of the algorithm is pretty straightforward and convenient due to the fact that the process know how many messages should receive and send in the selection phase. The processes communicate between each other by well structured and lightweight messages. They also have their own states used in order to recognise their function. Since the algorithm is randomised, it can fail sometimes, but the reinvocation is implemented.

2.1.1 Message structure

Messages structure is implemented as follows:

- MessageType

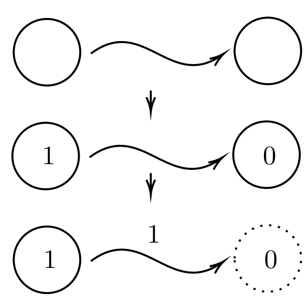

Figure 1: Visualization of the selection phase

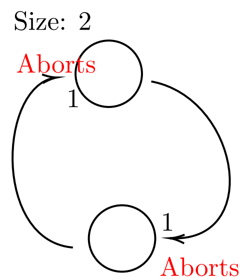

Figure 2: Visualization of the verification phase

- DATA (`[]byte`)

  Where MESSAGETYPE can be one of the following:

- Elimination – message in the selection phase of the algorithm
- Counter – message in the verification phase of the algorithm
- Elected – just to confirm that the leader is elected, used to stop

2.1.2 Process state

Each process has one of the following states.

- Unknown – the process is active, but not yet leader
- Nonleader – the process is inactive
- Leader – the process is a leader

Initially every process has *Unknown* state

3 Correctness and complexity analysis

A leader election algorithm may return incorrect answer in two cases: if it elected more than one leader or no process were elected. However, in our algorithm the following happen:

- It is highly likely that after $c = 5 \log n$ rounds of selection phase there will be no more than one active process. In case of the bad luck, we can just reinvoke it.
- Always at least one process is active; If any process becomes inactive, then it must be at least one which got bit 1.

3.1 Bit complexity analysis

We assume that we flip fair coin here (1 and 0 with prob. $\frac{1}{2}$). The probability that an active process becomes inactive in a round is $\frac{1}{4}$. Therefore, by linearity of expectations, expected number of processes which become inactive in round 1 is $\frac{n}{4}$. So the expected number of rounds until only one process remains active is $\log_{4/3} n$. If we choose $c > 2 \log_{4/3} n \approx 4.8188 \log n$ then the probability that more than one process remains active after c rounds is small[1]. The total bit complexity of the selection phase is $cn = \mathcal{O}(n \log n)$. In the verification phase it is $a\mathcal{O}(n \log n)$, where $a = 1$ is expected number of active processes after c rounds.

3.2 Probabilistic analysis

Let X_t - number of active processes which became inactive in stage t
Also

$$D_t = \sum_{i=1}^{t} X_i$$

$$N_t = n - D_t$$

$$Q_t(z) = \sum_{d \geq 0} \Pr(D_t = d) z^d$$

We can calculate probability of $X_1 = k$

$$\Pr(X_1 = k) = 2^{-n+1} \binom{n}{2k}$$

$$Q_1(z) = \sum_{k \geq 0} \Pr(X_1 = k) z^k = 2^{-n}[(1 + \sqrt{z})^n + (z - \sqrt{z})^n]$$

$$E(X_1) = Q_1'(1) = \frac{n}{4}$$

$$E(X_1^2) = Q_1''(1) + E(X_1) = \frac{n(n+1)}{16}$$

We define $\tilde{N}$ such that

$$E(\tilde{X} | \tilde{D}_{t-1} = d) = \frac{n-d}{4}$$

Thus

$$N_t = \begin{cases} N_t, & N_t > 1 \\ 1, & \text{otherwise} \end{cases}$$

Lemma 3.1.

$$E(\tilde{D}_t) = n(1 - (3/4)^t)$$

$$V(\tilde{D}_t) = \frac{n}{3} \left[\left(\frac{3}{4} \right)^t - \left(\frac{3}{4} \right)^{2t} \right]$$

Lemma 3.2.

$$\Pr(N_t > 1) \leq (n/3)(3/4)^t + n((3n-1)/3)(3/4)^{2t}$$

Proof. By Chebychev's inequality

$$\Pr(N_t > 1) \leq E(N_t^2) = E((n - \tilde{D}_t)^2) = \frac{n}{3} \left(\frac{3}{4} \right)^t + n \frac{3n-1}{3} \left(\frac{3}{4} \right)^{2t}$$

$\square$

Corollary 3.2.1. *For $c(n)/\log n > 2/\log_2 4/3 \approx 4.8188417$ the expected communication complexity of the algorithm is $\mathcal{O}(n \log n)$ bits*

References

[1] Itai A. and Rodeh M. "Symmetry breaking in distributed networks". In: *Information and Computation* 88.1 (1990), pp. 60–87.