Implementação do algoritmo de Itai-Rodeh

Alex Davis Neuwiem da Silva (21202103)
Luan Diniz Moraes (21204000)
Lucas Castro Truppel Machado (22100632)

Introdução I

- A eleição de líderes é o problema de eleger um único líder em uma rede, no sentido de que o processo líder sabe que foi eleito e os demais processos sabem que não foram eleitos.
- Os algoritmos de eleição de líder exigem que todos os processos tenham o mesmo algoritmo local e que cada cálculo termine com um processo eleito como líder.

Introdução II

- O algoritmo de Itai-Rodeh é um algoritmo probabilístico de eleição de líderes para anéis unidirecionais anônimos, baseado no algoritmo de Chang-Roberts.
- Cada processo seleciona uma identidade aleatória de um domínio finito e os processos com a maior identidade iniciam um novo turno eleitoral se detectarem um conflito de nomes.
- Supõe-se que o tamanho do anel é conhecido por todos os processos.
- Cada processo consegue reconhecer sua própria mensagem por meio de um contador de saltos que faz parte da mensagem.

Suposições iniciais I

- Consideramos um anel assíncrono, anônimo e unidirecional composto por $n \geq 2$ processos, sendo eles denominados de p_0 até p_{n-1}.
- Os processos comunicam-se de forma assíncrona, enviando e recebendo mensagens através de canais que são considerados confiáveis e possuem capacidade n.
- Os canais são unidirecionais: uma mensagem enviada pelo processo p_i é adicionada à fila de mensagens do processo $p_{(i+1) \bmod n}$.

Suposições iniciais II

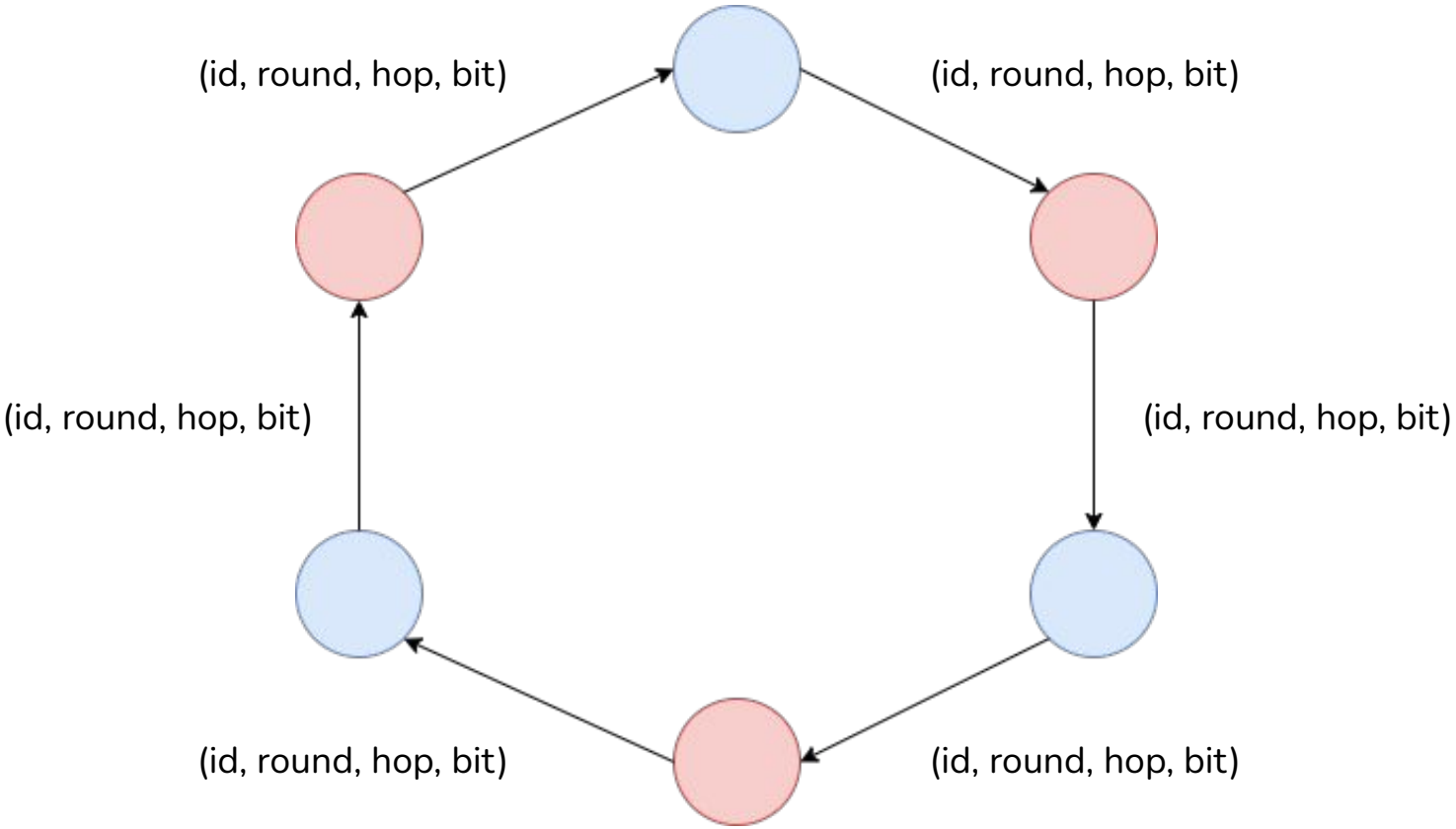
- Supõe-se que receber uma mensagem, processá-la e possivelmente enviar uma mensagem subsequente leva tempo zero (ou seja, é instantâneo).
- As filas de mensagens são guiadas por um escalonador justo, o que significa que em cada sequência infinita de execução, cada mensagem enviada eventualmente chega ao seu destino.
- Os processos são anônimos, portanto não possuem identidades únicas.

Atributos do Processo

- ***n*** é o número total de processos.
- ***k*** é o número total de ***id***s disponíveis.
- ***id*** é um valor gerado aleatoriamente (os ***id***s podem repetir).
- ***state*** varia entre {***ACTIVE***, ***PASSIVE***, ***LEADER***}.
- ***round*** representa o número do round de eleição atual.
- Cada processo possui sua fila de mensagens.
- Todo processo envia uma mensagem para o próximo do ciclo.

Atributos da Mensagem

- ***id*** e ***round*** são obtidos do processo que originou a mensagem.
- ***hop*** é um contador inicializado em 1 e é incrementado toda vez que um processo repassa a mensagem.
- ***bit*** é um booleano inicializado como ***true*** e que recebe o valor ***false*** quando a mensagem visita um processo com o mesmo ***id*** que seu emissor.


(id, round, hop, bit)
(id, round, hop, bit)
(id, round, hop, bit)
(id, round, hop, bit)
(id, round, hop, bit)
(id, round, hop, bit)

Implementação do algoritmo

- Os processos se comportam com base em seu parâmetro ***state*** e nos parâmetros da mensagem recebida.
- A execução só termina quando todo processo estiver no estado ***PASSIVE*** ou ter sido eleito como líder e quando não restarem mais mensagens em sua fila de mensagens.

The Itai-Rodeh algorithm.

- Initially, all processes are active, and each process p_i randomly selects its identity $id_i \in \{1, \ldots, k\}$ and sends the message $(id_i, 1, 1, \mathit{true})$.

- Upon receipt of a message $(id, \mathit{round}, \mathit{hop}, \mathit{bit})$, a passive process p_i $(\mathit{state}_i = \mathit{passive})$ passes on the message, increasing the counter hop by one; an active process p_i $(\mathit{state}_i = \mathit{active})$ behaves according to one of the following steps:

  - if $\mathit{hop} = n$ and $\mathit{bit} = \mathit{true}$, then p_i becomes the leader $(\mathit{state}'_i = \mathit{leader})$;

  - if $\mathit{hop} = n$ and $\mathit{bit} = \mathit{false}$, then p_i selects a new random identity $id'_i \in \{1, \ldots, k\}$, moves to the next round $(\mathit{round}'_i = \mathit{round}_i + 1)$, and sends the message $(id'_i, \mathit{round}'_i, 1, \mathit{true})$;

  - if $(\mathit{round}, id) = (\mathit{round}_i, id_i)$ and $\mathit{hop} < n$, then p_i passes on the message $(id, \mathit{round}, \mathit{hop} + 1, \mathit{false})$;

  - if $(\mathit{round}, id) > (\mathit{round}_i, id_i)$ (where (round, id) and (round_i, id_i) are compared lexicographically), then p_i becomes passive $(\mathit{state}'_i = \mathit{passive})$ and passes on the message $(id, \mathit{round}, \mathit{hop} + 1, \mathit{bit})$;

  - if $(\mathit{round}, id) < (\mathit{round}_i, id_i)$, then p_i purges the message.

Referências

- https://satoss.uni.lu/members/jun/papers/JUCS06.pdf
- https://www.cse.msu.edu/~borzoo/teaching/15/CAS769/lectures/week4.pdf
- https://www.lix.polytechnique.fr/comete/seminar/180105.pdf