ALGORITMOS DISTRIBUÍDOS
Algoritmos de eleição

Algoritmos Distribuídos (eleição)

- Em sistemas distribuídos, diversos algoritmos necessitam que um processo funcione como coordenador, inicializador, sequenciador, enfim, ter um papel especial
- exemplos.:
  - coordenador de exclusão mútua com controle centralizado
  - coordenador para detecção de deadlock distribuído
  - seqüenciador de eventos para ordenação consistente centralizada
  - etc.
- falha do coordenador compromete serviço para vários processos
- novo coordenador deve assumir - eleição!
  - Objetivo: eleger ***um*** processo, entre os ativos, para desempenhar função especial

Algoritmos Distribuídos (eleição)

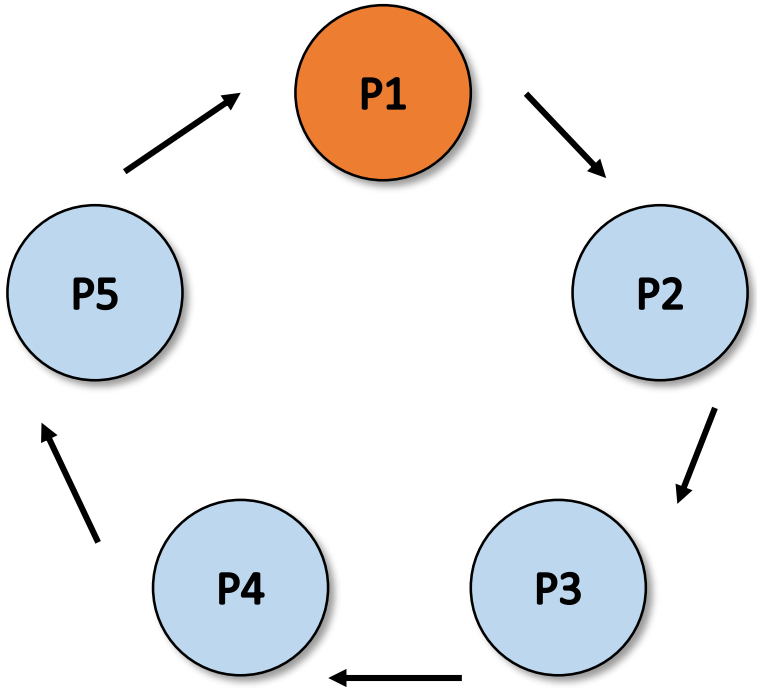
- Para os algoritmos, assume-se que: (*general assumptions*)
  - 1) Todo processo no sistema tem uma *prioridade única*
  - 2) Quando eleição acontece, o processo com maior prioridade *entre os processos ativos* é eleito como coordenador
  - 3) Na recuperação (volta à atividade), um processo falho pode tomar ações para juntar-se ao grupo de processos ativos

Algoritmos Distribuídos (eleição)

- ***Ring algorithm***
  - Baseado no uso de um anel lógico, sem uso de *token*
  - Cada processo conhece o anel inteiro, mas manda mensagens somente para o próximo processo ativo na direção do anel
  - Quando processo detecta que o coord. não está ativo, ele constrói uma msg ELEIÇÃO contendo seu id., e manda para o próximo do anel
  - A cada passo, o processo que recebe a msg inclui seu id na msg e envia para o próximo do anel
  - No final o processo que iniciou a eleição recebe a msg e escolhe aquele que tem maior id.
  - Nova msg é enviada novamente através do anel para todos contendo o novo coordenador
  - Uma vez que a msg passou por todos processos e chegou no originador, ela é retirada do sistema

Caso Real

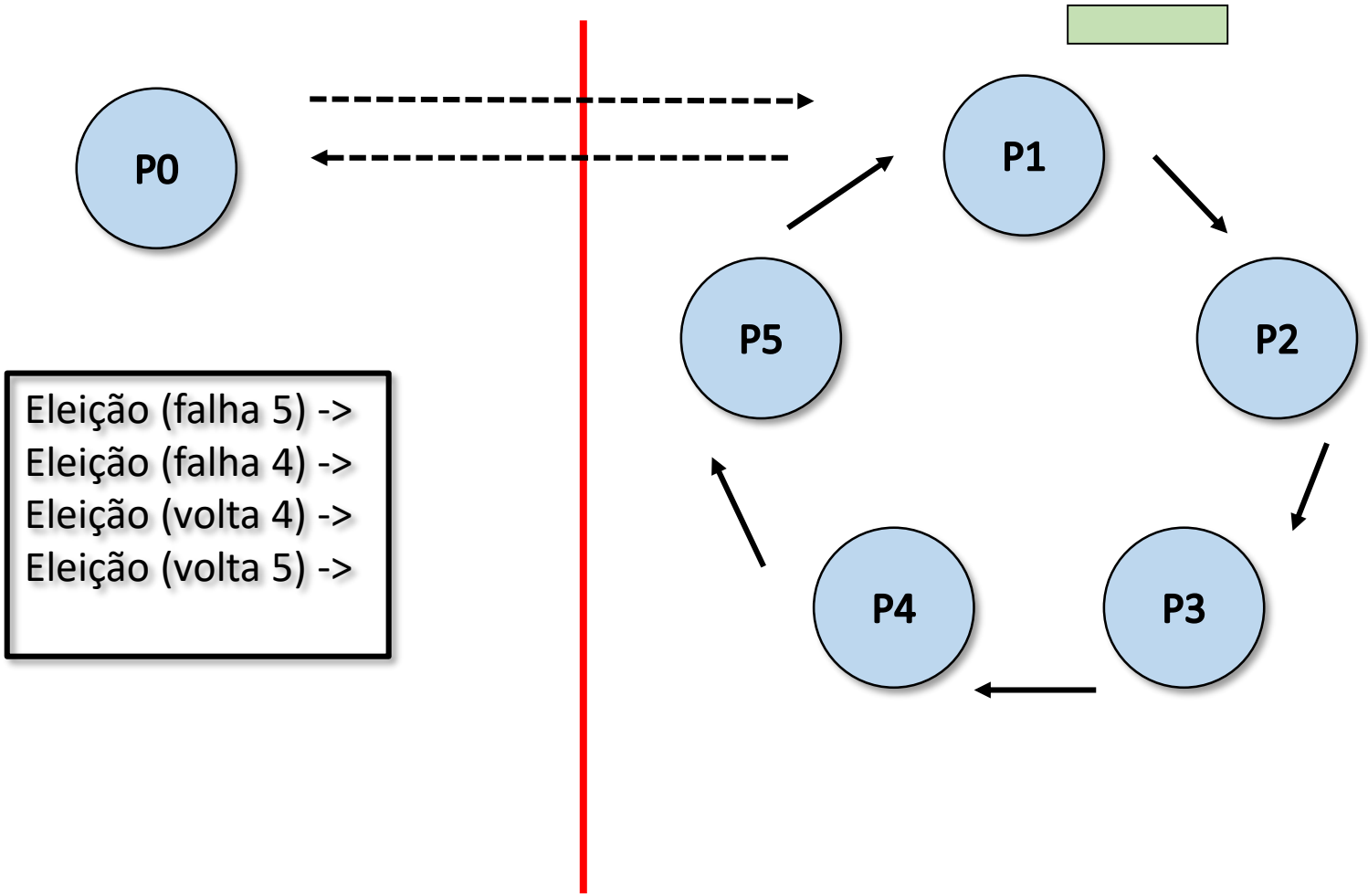
- *Processos falham e não respondem*
- *Todos que não estão eleitos (azuis) controlam o atual eleito (laranja)*
  - *Fazendo ping regularmente*
- *Quem descobrir falha primeiro dispara eleição*
- *Quando processo que falhou voltar dispara eleição*
- *Múltilplas eleições podem ocorrer ao mesmo tempo*



No nosso caso

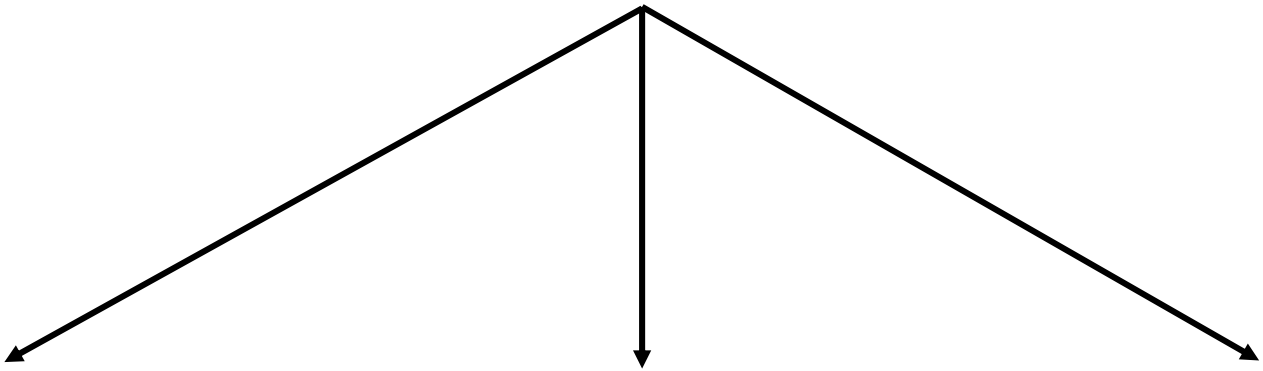
- *Emular eleição em máquina local*
- *Processos não falham naturalmente*
  - *Precisamos simular (forçar) falhas*
- *Sugestão: uso de um processo de controle externo ao anel*
  - *Gera eventos para testar o algoritmo distribuído*
  - *Funciona como um script de teste*
  - *Já loga os eventos que acontecem (para ser usado no relatório)*

Sugestão de Modelagem



Máquina de estados (em cada nó do anel de possíveis eleitos)

Receive (bloqueante)
- Dependendo do tipo da mensagem:



Dispara Eleição
1)Construo pacote eleição (com nó falho)
2)Coloco no Anel (próximo)
3)Recebo resultado (anterior)
4) Faço pacote confirmação
5) Coloco no anel (próximo)
6) Recebo confirmação (anterior)
7) Aviso controle (P0)

Votação eleição
1) Coloco meu ID
2) Coloco no Anel (próximo)
3) Se próximo falho pulo

Confirmação vencedor eleição
1) Anoto eleito
2) Coloco no Anel (próximo)
3) Se próximo falho pulo