

Computação Distribuída

Odorico Machado Mendizabal



Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE

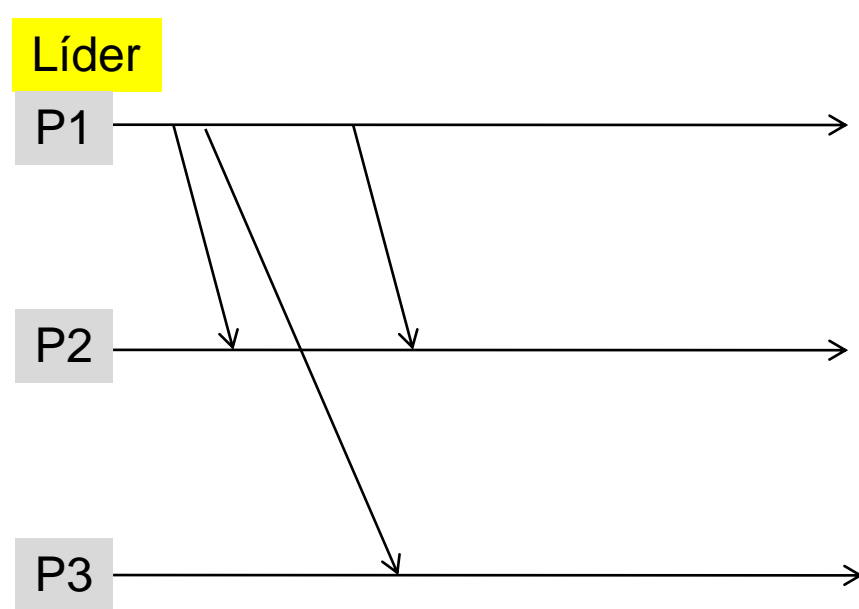


Eleição de Líder

Líder em Sistemas Distribuídos

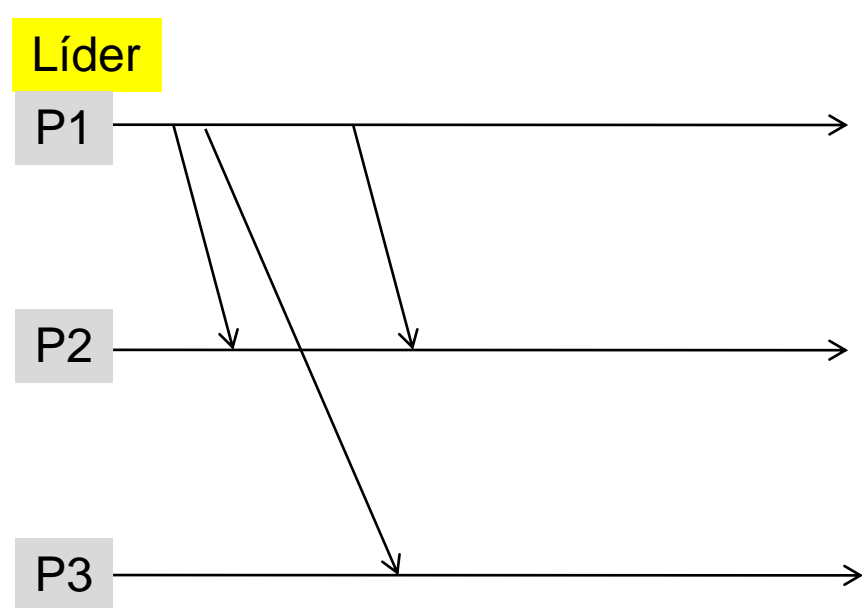
- Em sistemas distribuídos há situações em que um processo deve apresentar um comportamento diferenciado dos demais:
 - Sequenciador de mensagens
 - Árbitro para acesso à região compartilhada
 - Escalonador de tarefas
 - Validador de transações
 - Primário (no modelo de replicação passiva)
 - ...
- Este papel normalmente é atribuído a um líder (coordenador, mestre)

Líder em Sistemas Distribuídos



Protocolos distribuídos podem adotar o uso de um líder para distribuir ou coordenar tarefas entre os processos

Líder em Sistemas Distribuídos

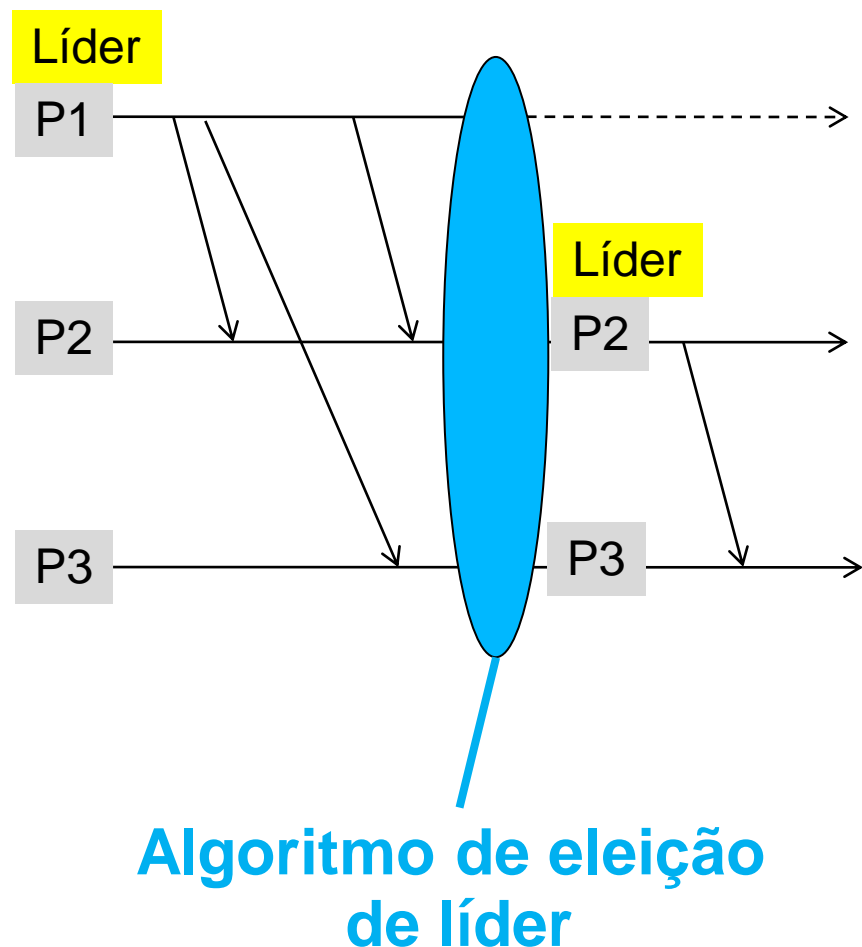


Protocolos distribuídos podem adotar o uso de um líder para distribuir ou coordenar tarefas entre os processos

Alguns desafios:

- O líder pode falhar
- O sistema pode ser dinâmico, permitindo a entrada ou saída de processos

Líder em Sistemas Distribuídos



Protocolos distribuídos podem adotar o uso de um líder para distribuir ou coordenar tarefas entre os processos

Alguns desafios:

- O líder pode falhar
- O sistema pode ser dinâmico, permitindo a entrada ou saída de processos

Nestes casos, pode haver necessidade de mudança de líder

Escolha do líder

- Normalmente a escolha de líder é feita através de uma eleição
- Processos trocam mensagens com o objetivo de alcançar um consenso sobre qual processo será o líder
- Aspectos como a possibilidade de falhas e o modelo do sistema devem implicar diretamente na forma como o algoritmo deve ser desenvolvido
 - Modelo do sistema síncrono ou assíncrono
 - Tipos de falha (parada, colapso, omissão, bizantinas)

Algoritmo de Chang e Roberts

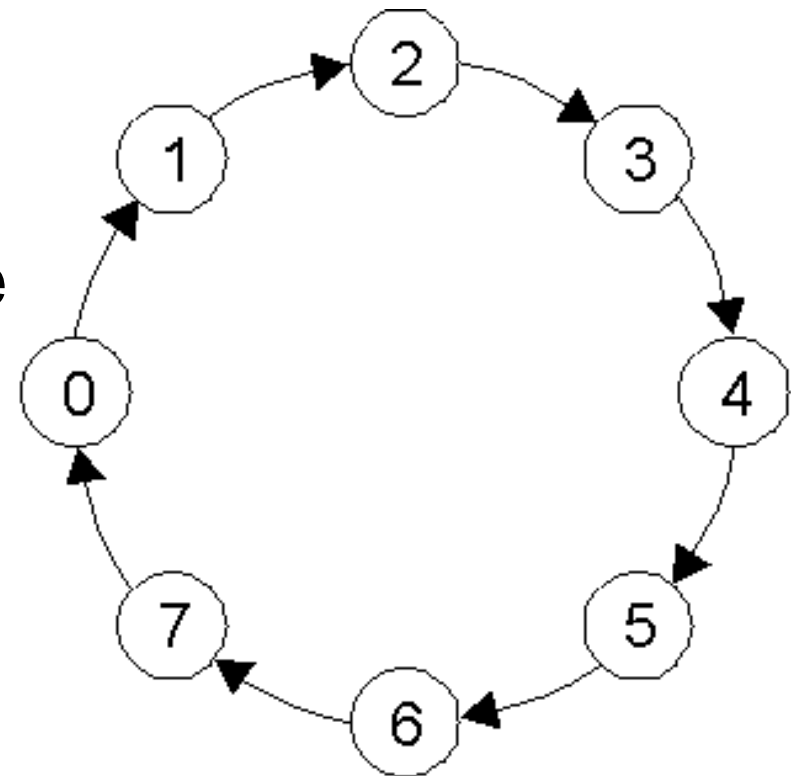
- Proposto em 1979, adaptando uma ideia do algoritmo de Gerard Le Lann (1977)
- Algoritmo descentralizado e assíncrono
- Cada processo executa o mesmo algoritmo localmente
- Sem relógio global
- Cada processo tem uma identidade única
- Número finito de processos

Algoritmo de Chang e Roberts

- Um processo não convoca mais de uma eleição enquanto não obtiver resposta
- N processos podem convocar até N eleições concorrentemente
- Apenas um líder é eleito

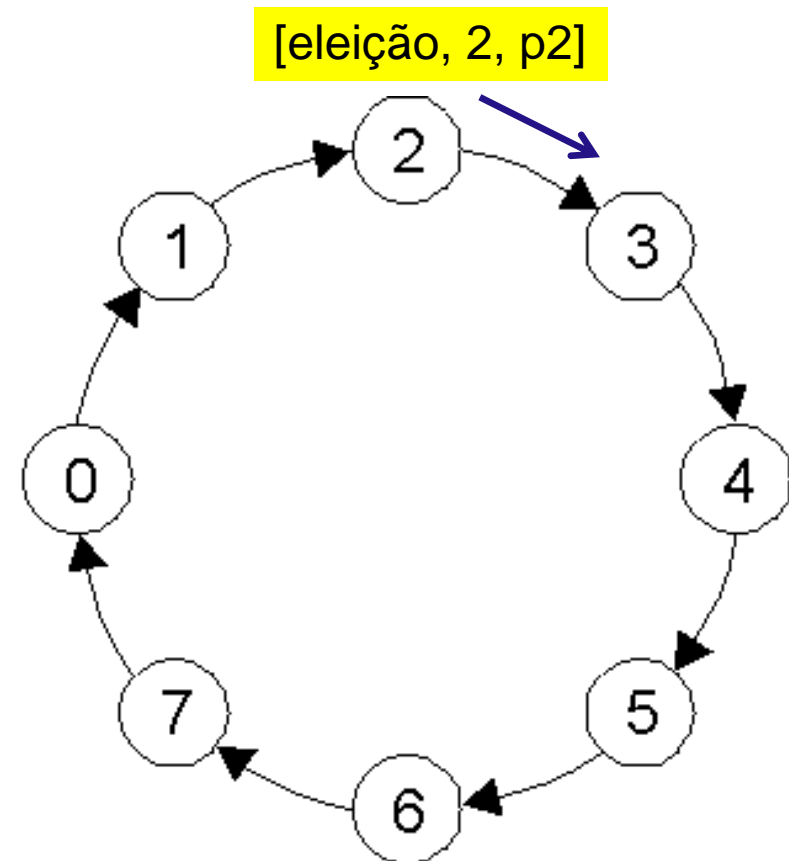
Algoritmo de Chang e Roberts

- Processos são organizados em anel unidirecional
- As mensagens são enviadas em sentido horário
- Algoritmo em duas rodadas:
 - Eleição: Primeira volta define-se o processo de maior ID
 - Vencedor: Segunda volta anuncia-se o vencedor



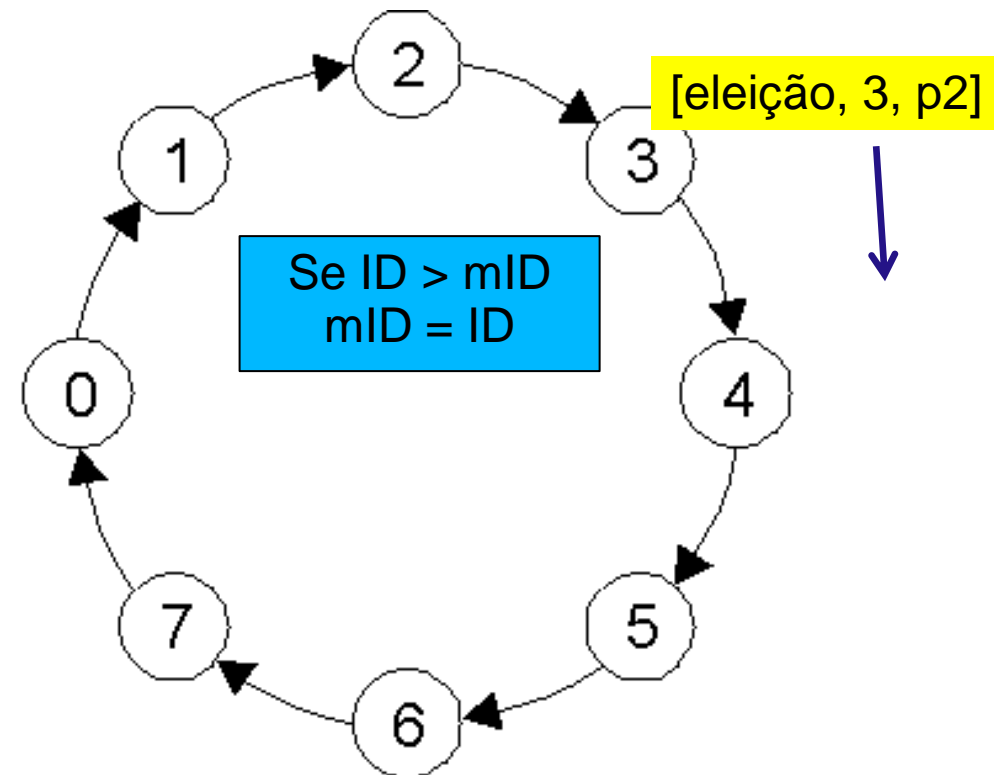
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição



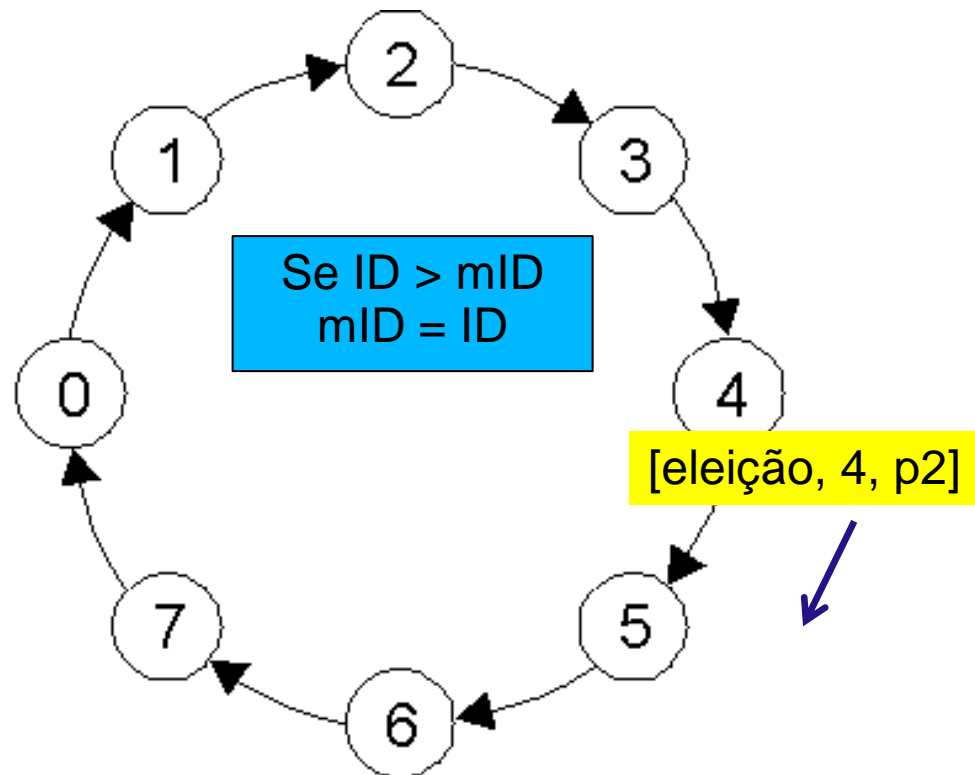
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)



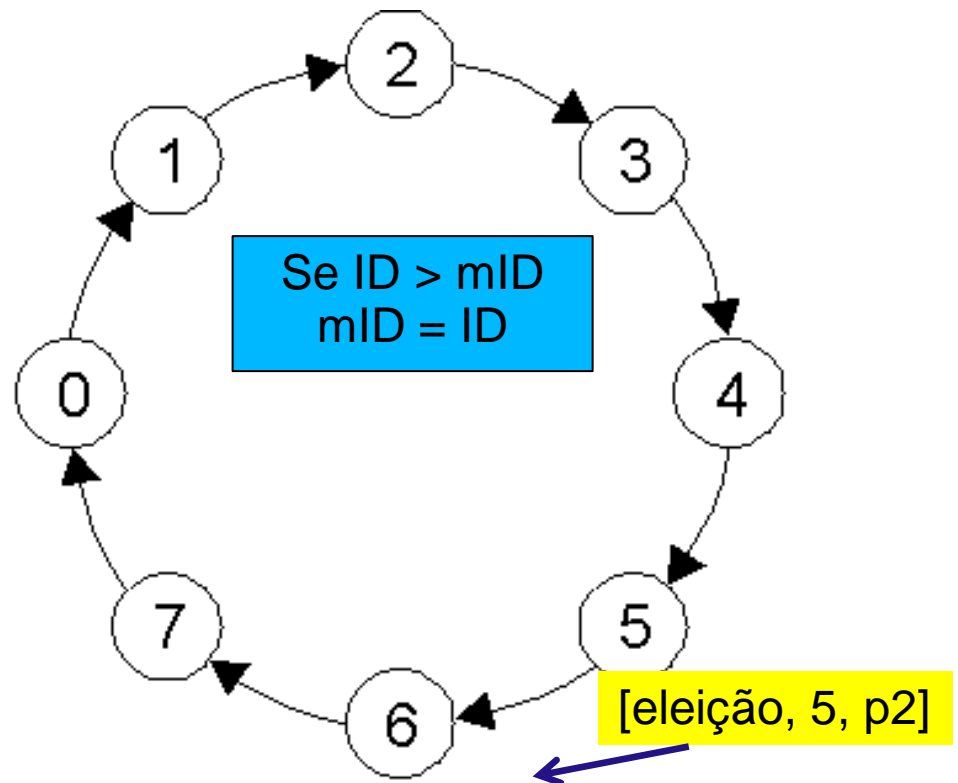
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)



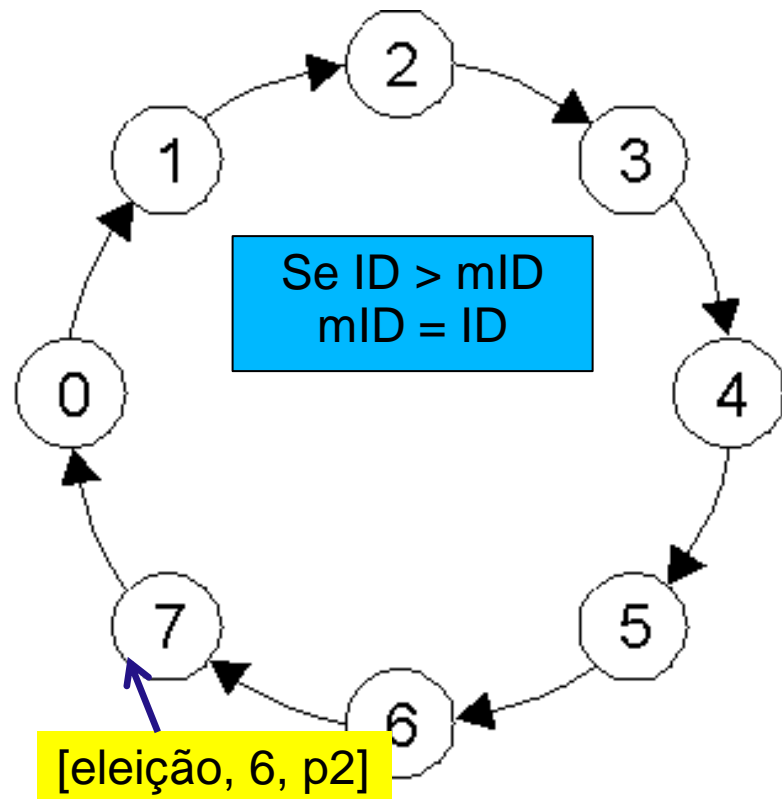
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)
- P5 substitui o ID da mensagem (P5 é o novo preferido)



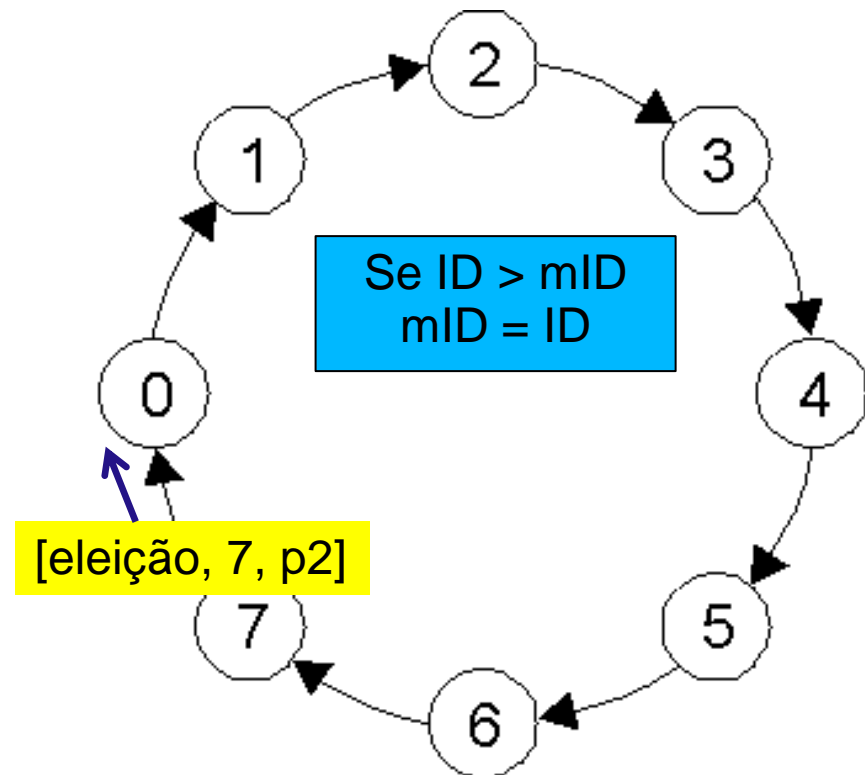
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)
- P5 substitui o ID da mensagem (P5 é o novo preferido)
- P6 substitui o ID da mensagem (P6 é o novo preferido)



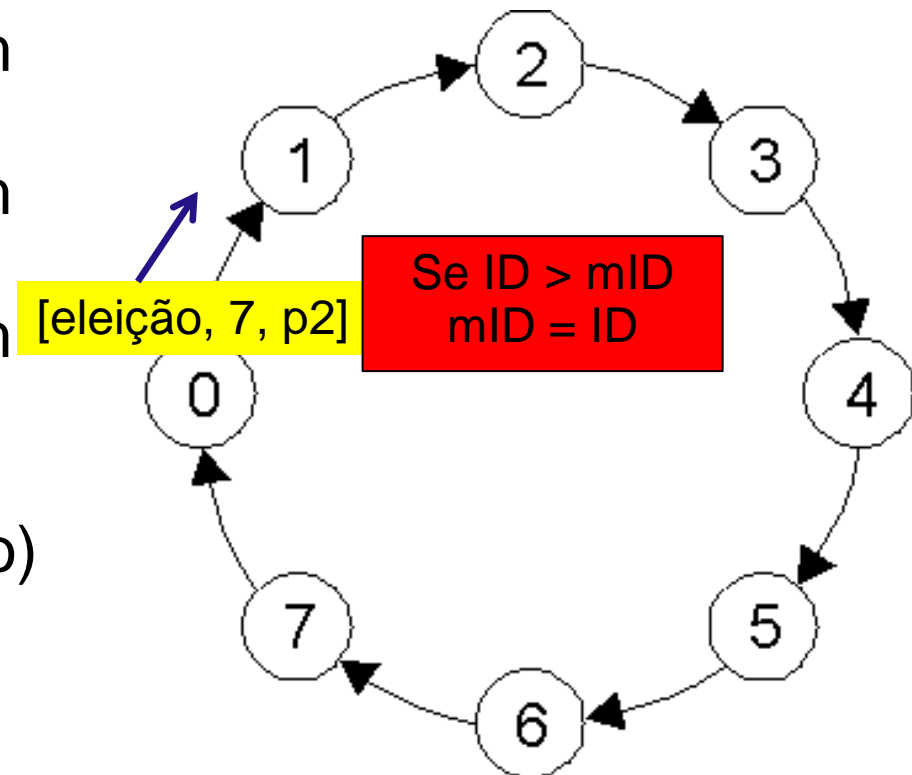
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)
- P5 substitui o ID da mensagem (P5 é o novo preferido)
- P6 substitui o ID da mensagem (P6 é o novo preferido)
- P7 substitui o ID da mensagem (P7 é o novo preferido)



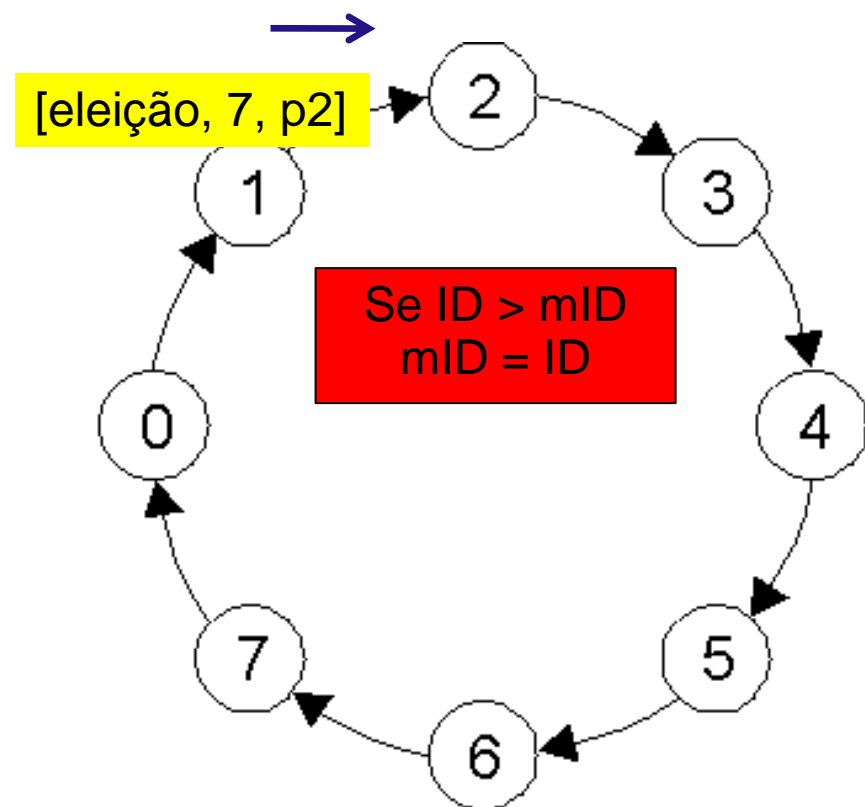
Algoritmo de Chang e Roberts – Round 1

- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)
- P5 substitui o ID da mensagem (P5 é o novo preferido)
- P6 substitui o ID da mensagem (P6 é o novo preferido)
- P7 substitui o ID da mensagem (P7 é o novo preferido)
- P0 não substitui o ID da mensagem (P7 ainda é preferido)



Algoritmo de Chang e Roberts – Round 1

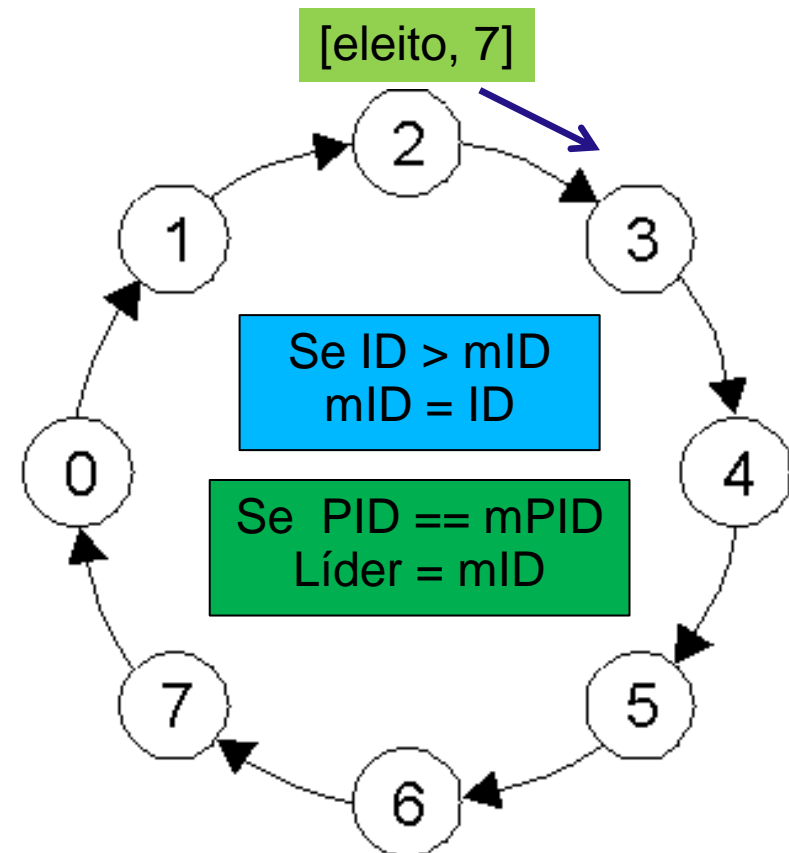
- P2 inicia eleição
- P3 substitui o ID da mensagem (P3 é o novo preferido)
- P4 substitui o ID da mensagem (P4 é o novo preferido)
- P5 substitui o ID da mensagem (P5 é o novo preferido)
- P6 substitui o ID da mensagem (P6 é o novo preferido)
- P7 substitui o ID da mensagem (P7 é o novo preferido)
- P0 não substitui o ID da mensagem (P7 ainda é preferido)
- P1 não substitui o ID da mensagem (P7 ainda é preferido)



Algoritmo de Chang e Roberts – Round 2

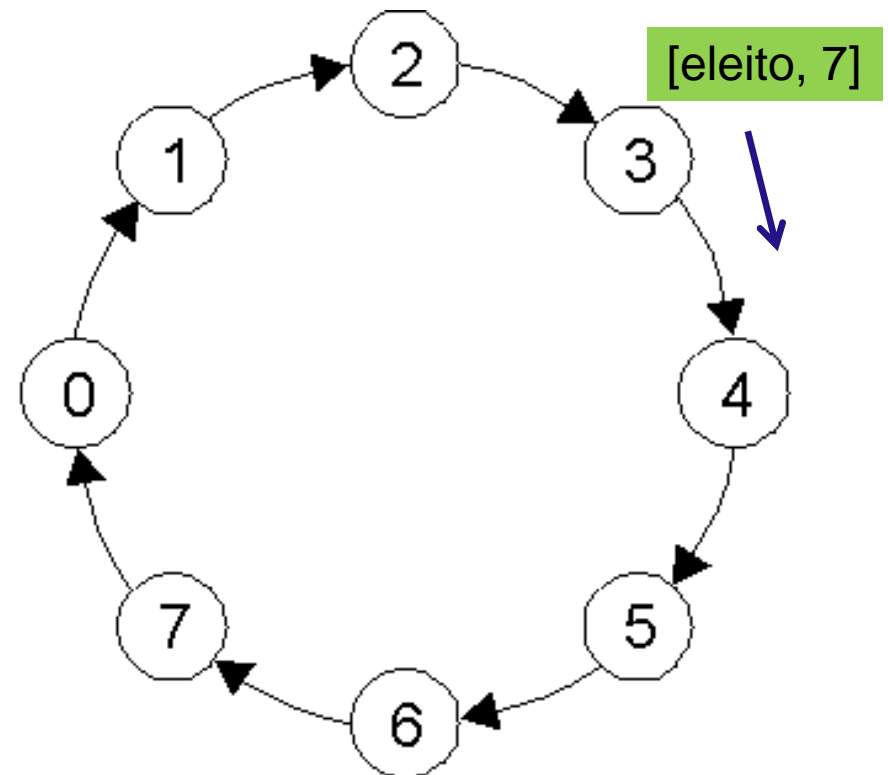
- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição

Líder decidido: P7



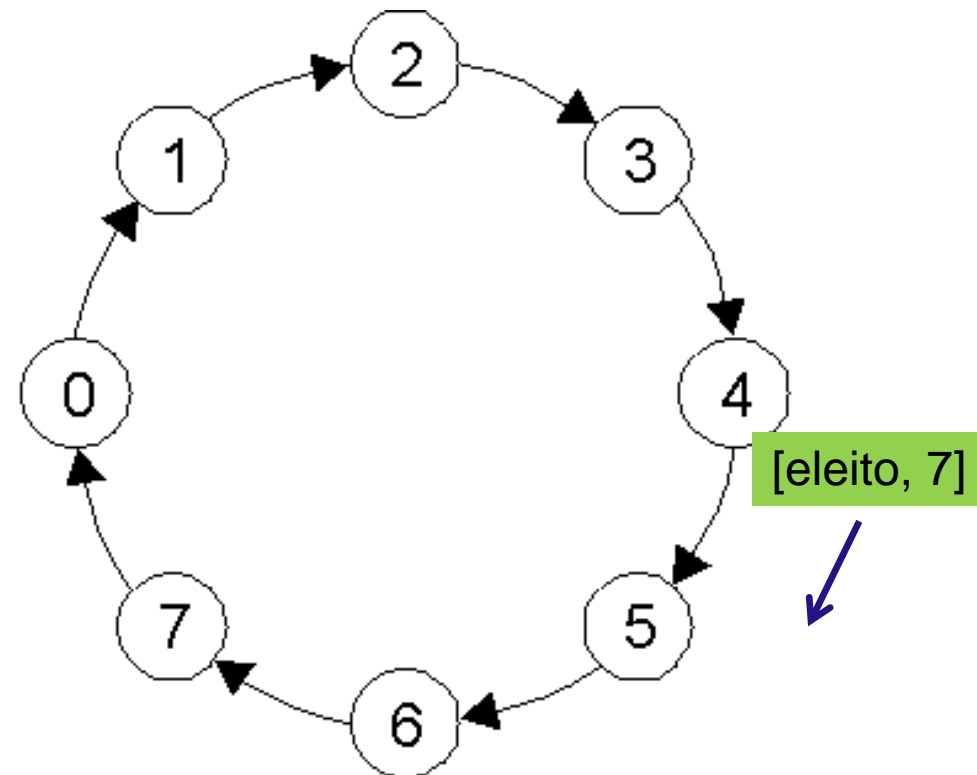
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição



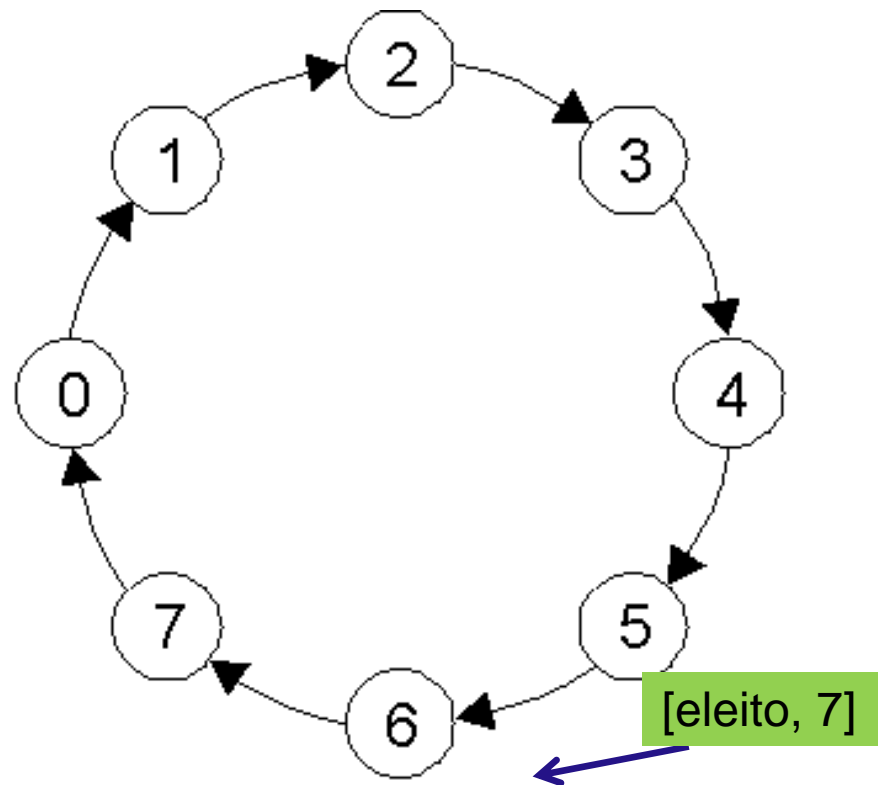
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição
- P4 repassa resultado da eleição



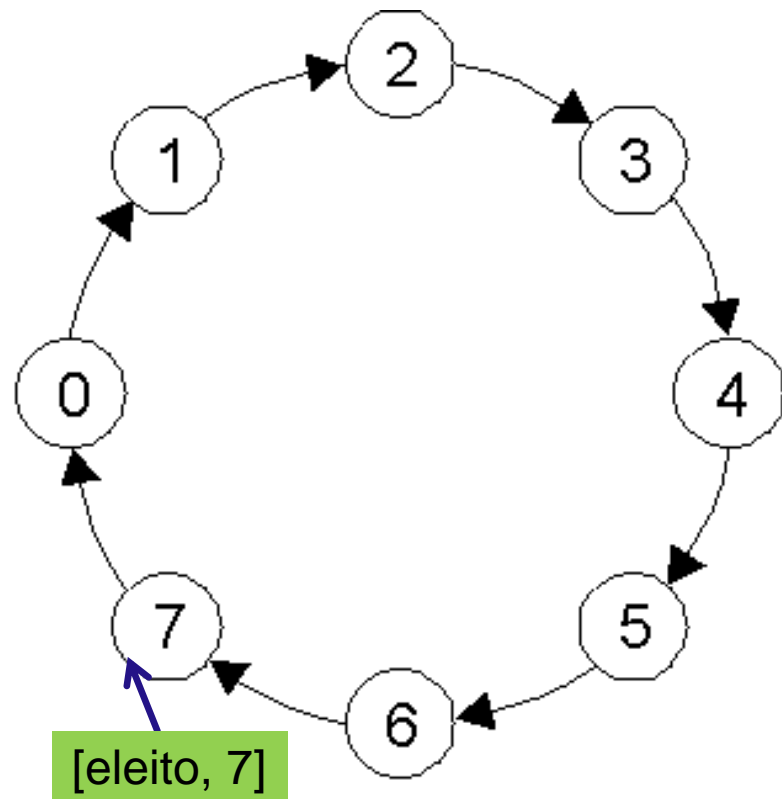
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição
- P4 repassa resultado da eleição
- P5 repassa resultado da eleição



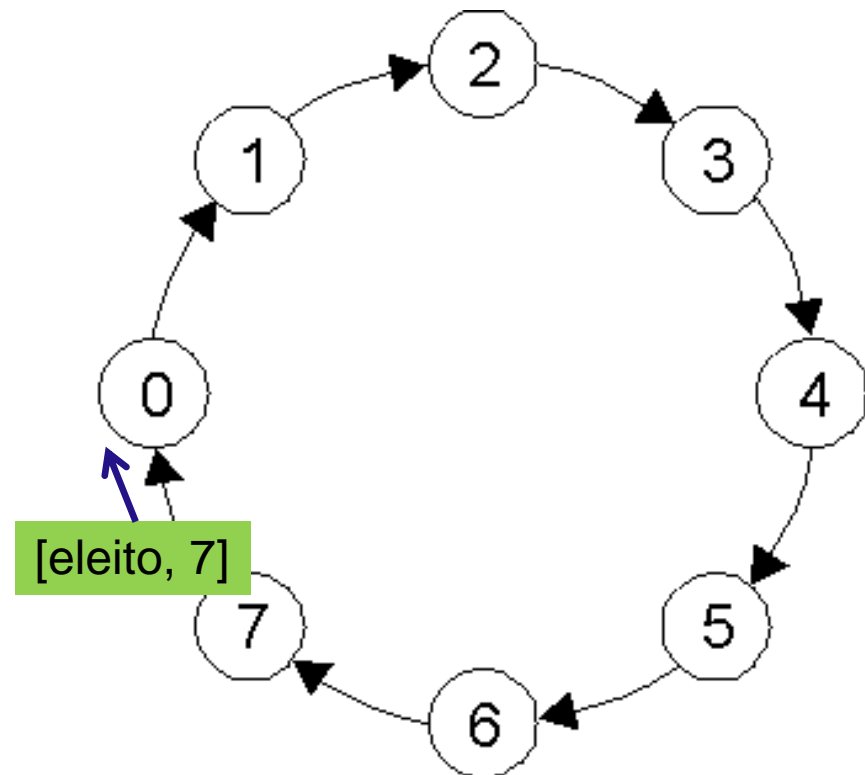
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição
- P4 repassa resultado da eleição
- P5 repassa resultado da eleição
- P6 repassa resultado da eleição



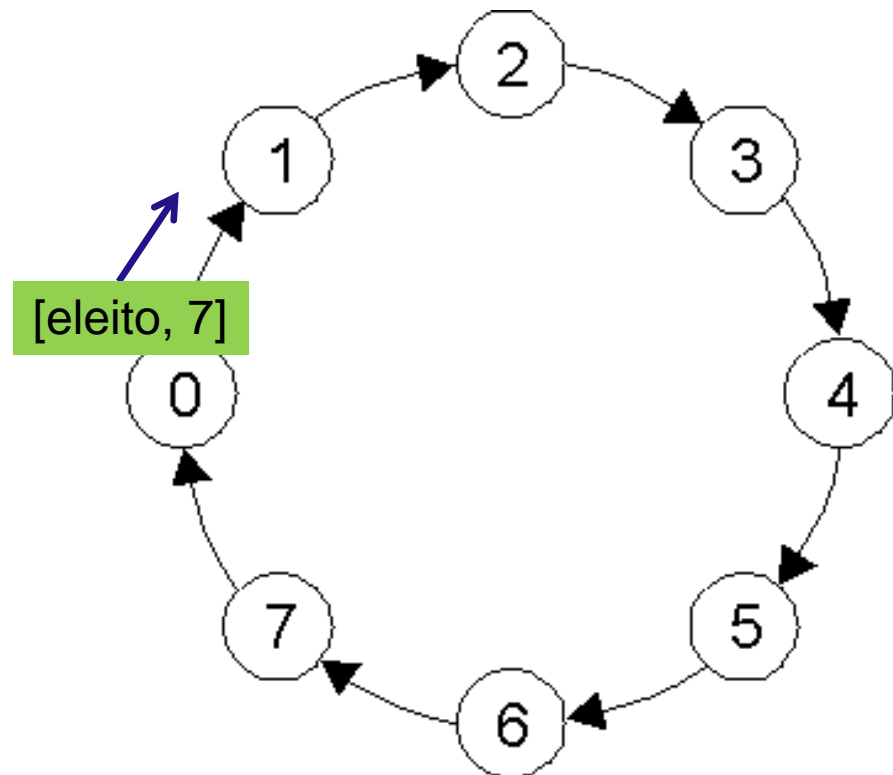
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição
- P4 repassa resultado da eleição
- P5 repassa resultado da eleição
- P6 repassa resultado da eleição
- P7 repassa resultado da eleição



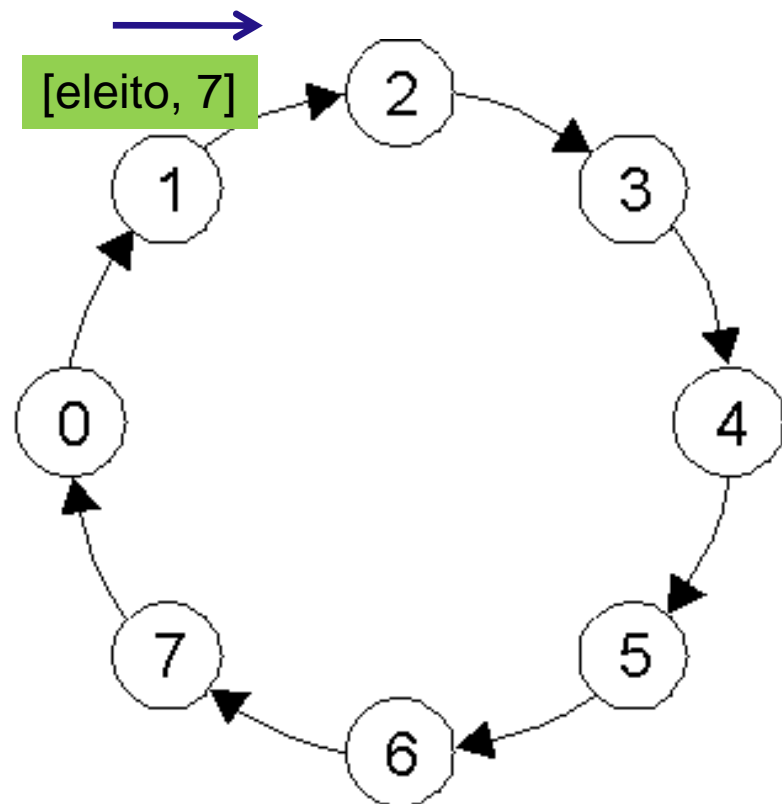
Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
- P2 repassa resultado da eleição
- P3 repassa resultado da eleição
- P4 repassa resultado da eleição
- P5 repassa resultado da eleição
- P6 repassa resultado da eleição
- P7 repassa resultado da eleição
- P1 repassa resultado da eleição



Algoritmo de Chang e Roberts – Round 2

- A mensagem circulou por toda a rede e o maior ID foi descoberto
 - P2 repassa resultado da eleição
 - P3 repassa resultado da eleição
 - P4 repassa resultado da eleição
 - P5 repassa resultado da eleição
 - P6 repassa resultado da eleição
 - P7 repassa resultado da eleição
 - P1 repassa resultado da eleição
 - Todos os processos sabem quem é o líder
- FIM



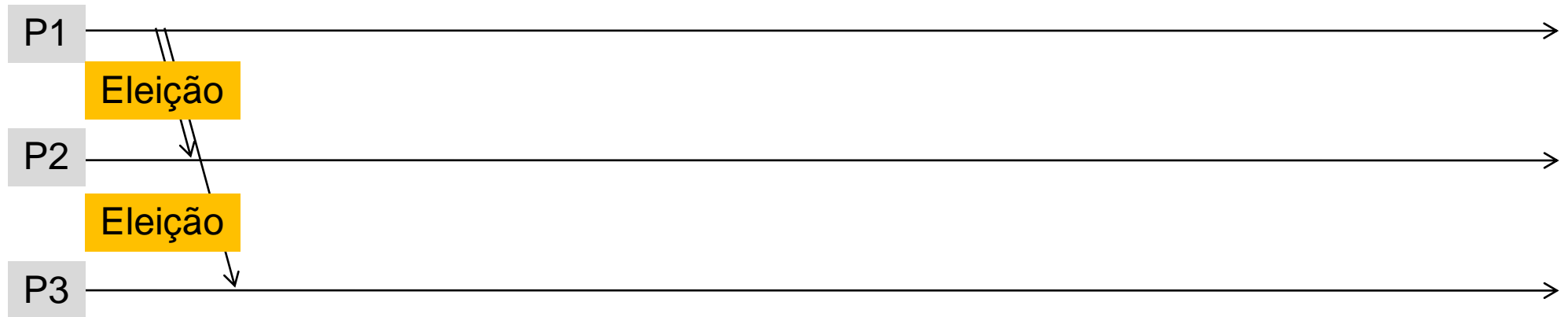
Algoritmo de Chang e Roberts

- Assume-se que processos não falham ou no caso de falhas o anel precisa ser restaurado
- Funciona em sistemas assíncronos
- Comunicação confiável (mensagens são entregues)

Algoritmo do Valentão (Bully)

- Proposto por Garcia-Molina, em 1982
- **Premissas:**
 - Processos possuem um identificador único
 - O sistema é síncrono
 - Processos são equipados com um detector de falhas
 - Canais de comunicação são confiáveis
 - mensagens enviadas por processos corretos não são perdidas
 - Até f processos podem falhar em um sistema com n processos, onde $n = f+1$
 - Processos podem falhar e se recuperar

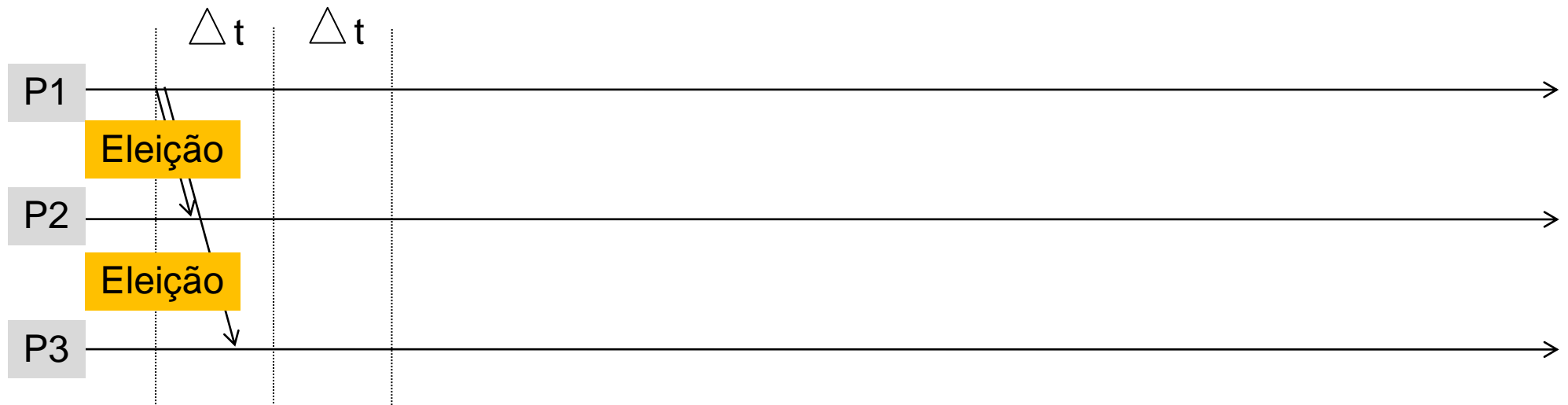
Algoritmo do Valentão (Bully)



Qualquer processo inicia a eleição (por exemplo P1)

- Envia mensagem Eleição para processos com ID maior que o seu

Algoritmo do Valentão (Bully)

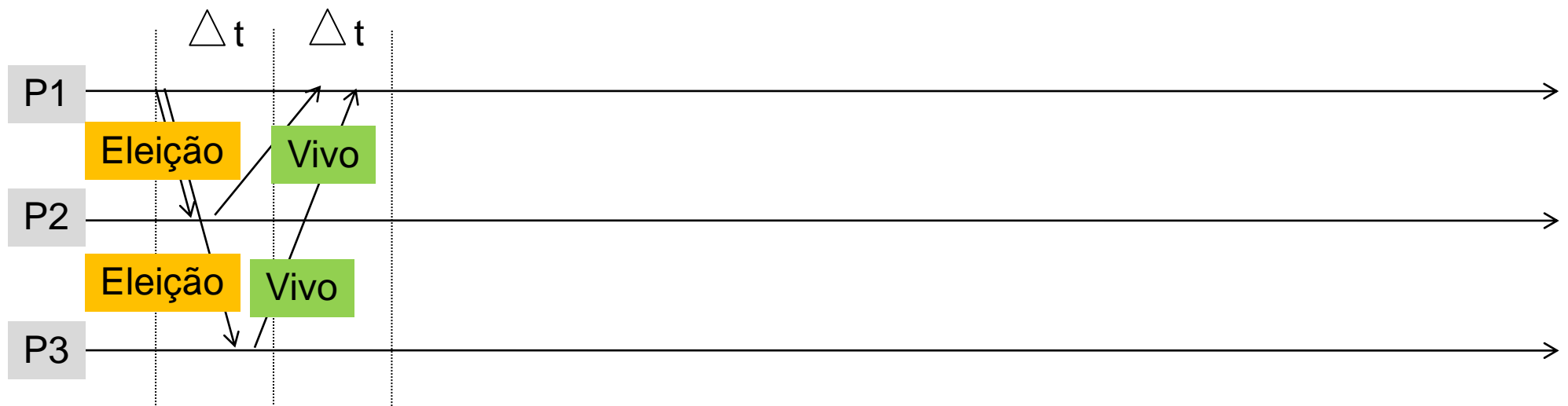


Qualquer processo inicia a eleição (por exemplo P1)

- Envia mensagem Eleição para processos com ID maior que o seu

- Sistema é síncrono, então processo aguarda respostas em até 2 rodadas desde o envio da mensagem

Algoritmo do Valentão (Bully)



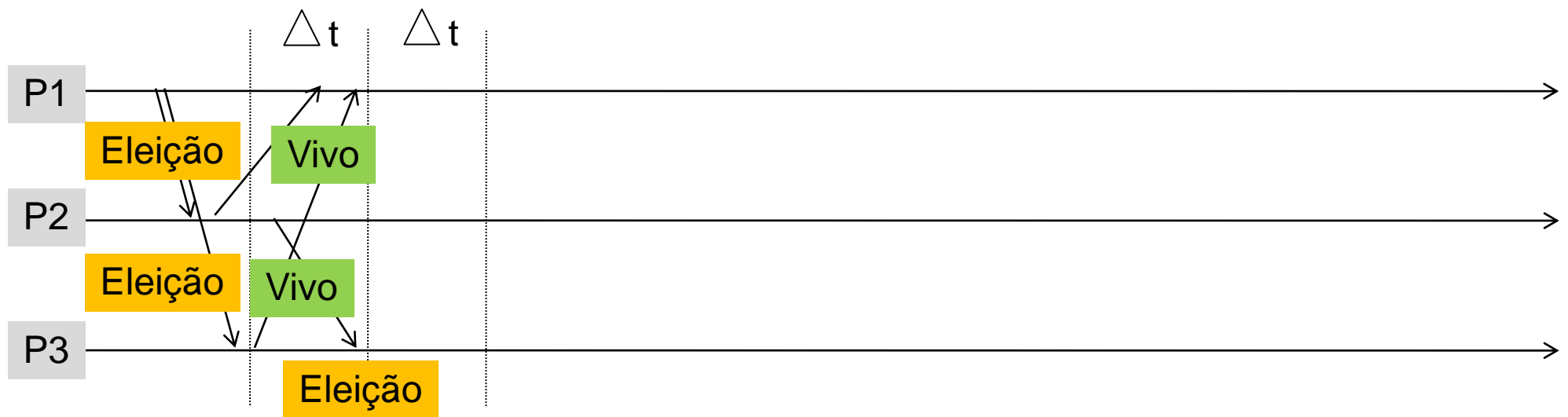
Qualquer processo inicia a eleição (por exemplo P1)

- Envia mensagem Eleição para processos com ID maior que o seu

- Sistema é síncrono, então processo aguarda respostas em até 2 rodadas desde o envio da mensagem

- Processo que recebe mensagem Eleição responde com mensagem indicando que está vivo

Algoritmo do Valentão (Bully)



Qualquer processo inicia a eleição (por exemplo P1)

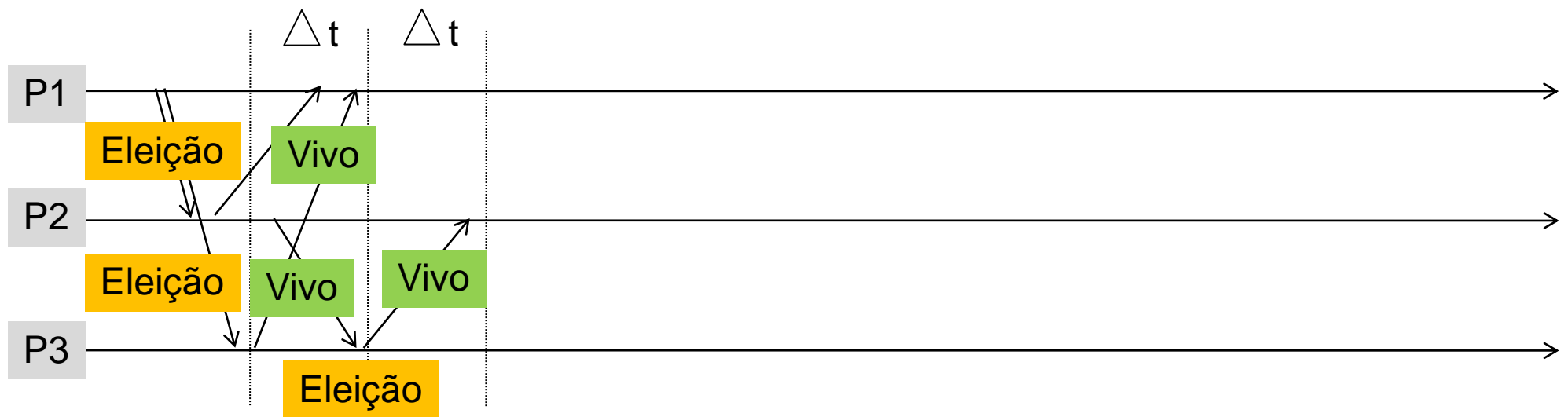
- Envia mensagem Eleição para processos com ID maior que o seu

- Sistema é síncrono, então processo aguarda respostas em até 2 rodadas desde o envio da mensagem

- Processo que recebe mensagem Eleição responde com mensagem indicando que está vivo

- Processo que recebe a mensagem Eleição encaminha a mensagem para processos com ID maior que o seu

Algoritmo do Valentão (Bully)



Qualquer processo inicia a eleição (por exemplo P1)

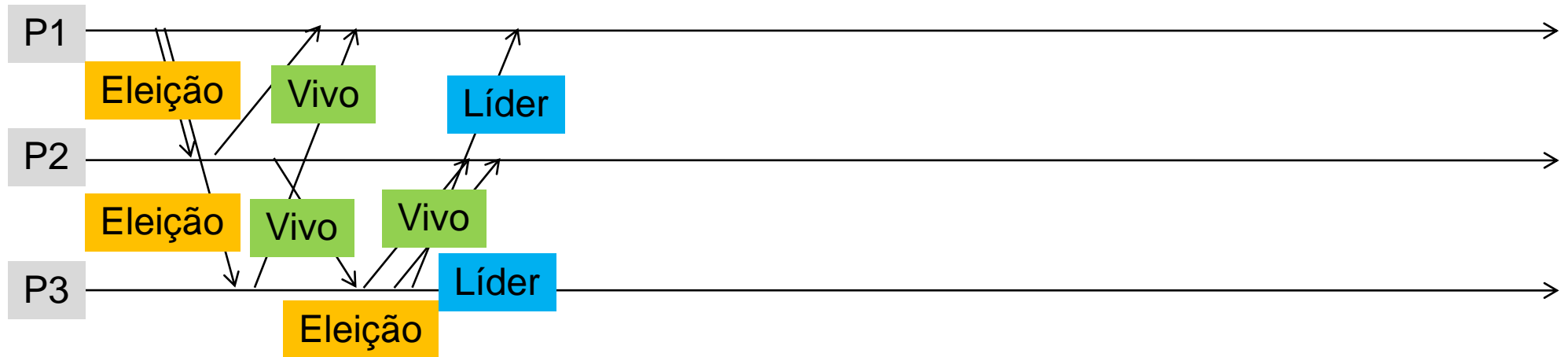
- Envia mensagem Eleição para processos com ID maior que o seu

- Sistema é síncrono, então processo aguarda respostas em até 2 rodadas desde o envio da mensagem

- Processo que recebe mensagem Eleição responde com mensagem indicando que está vivo

- Processo que recebe a mensagem Eleição encaminha a mensagem para processos com ID maior que o seu

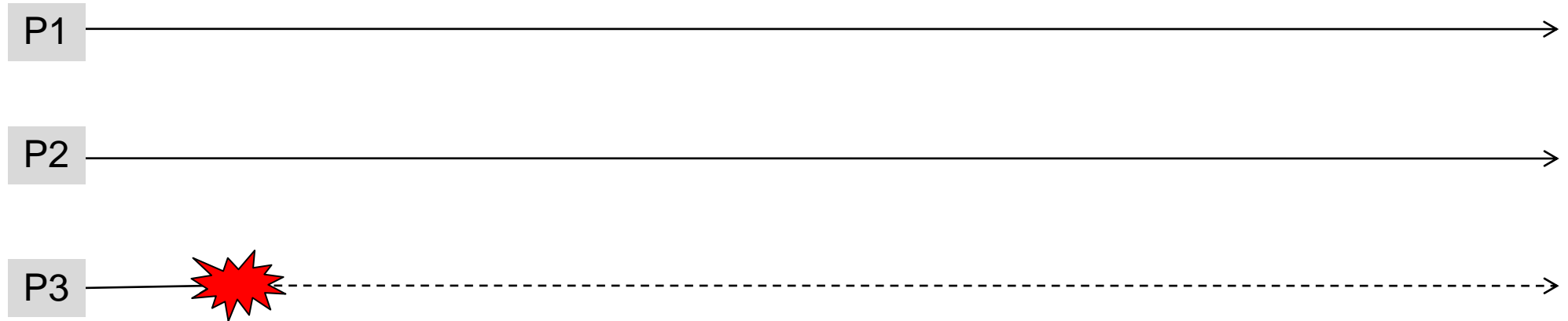
Algoritmo do Valentão (Bully)



- Não havendo processo com maior ID, o processo P3 proclama-se líder e envia mensagem para todos os processos informando que é o líder

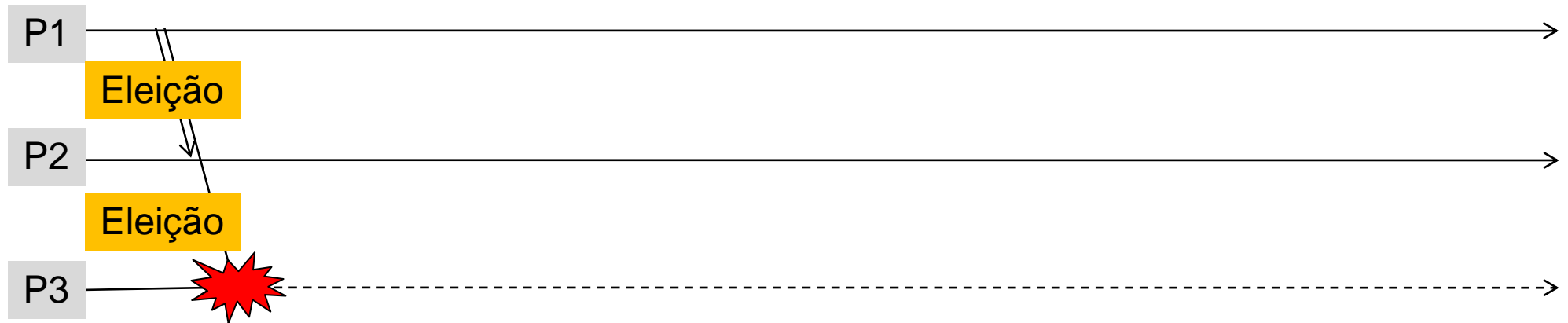
Algoritmo do Valentão (Bully) – Caso com processo falho (1)

Processo P3 falhou



Algoritmo do Valentão (Bully) – Caso com processo falho (1)

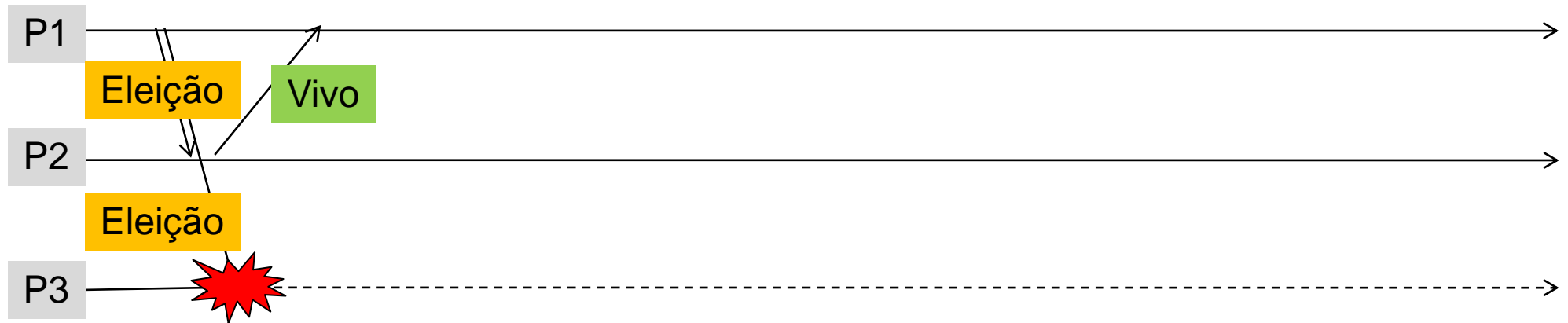
Processo P3 falhou



- Não há líder no sistema, P1 inicia eleição

Algoritmo do Valentão (Bully) – Caso com processo falho (1)

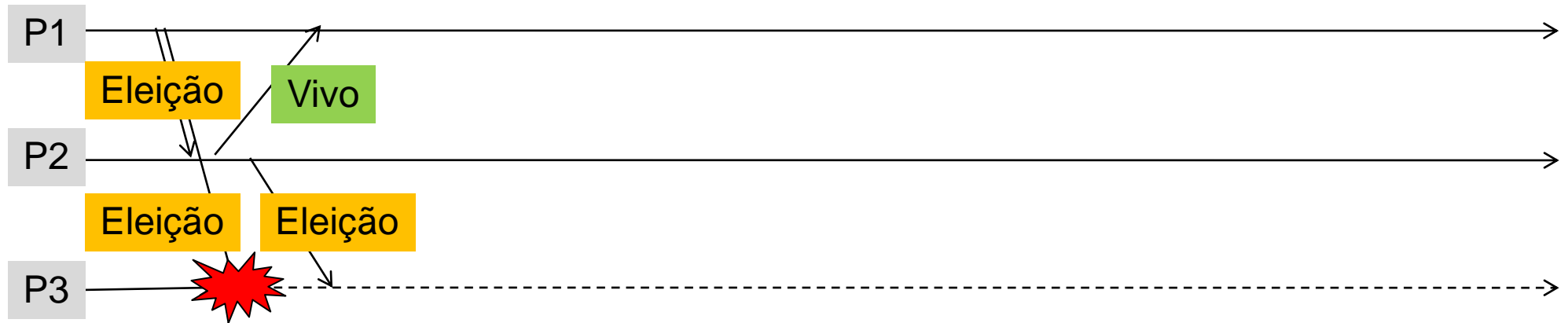
Processo P3 falhou



- P3 está falho, portanto não responde

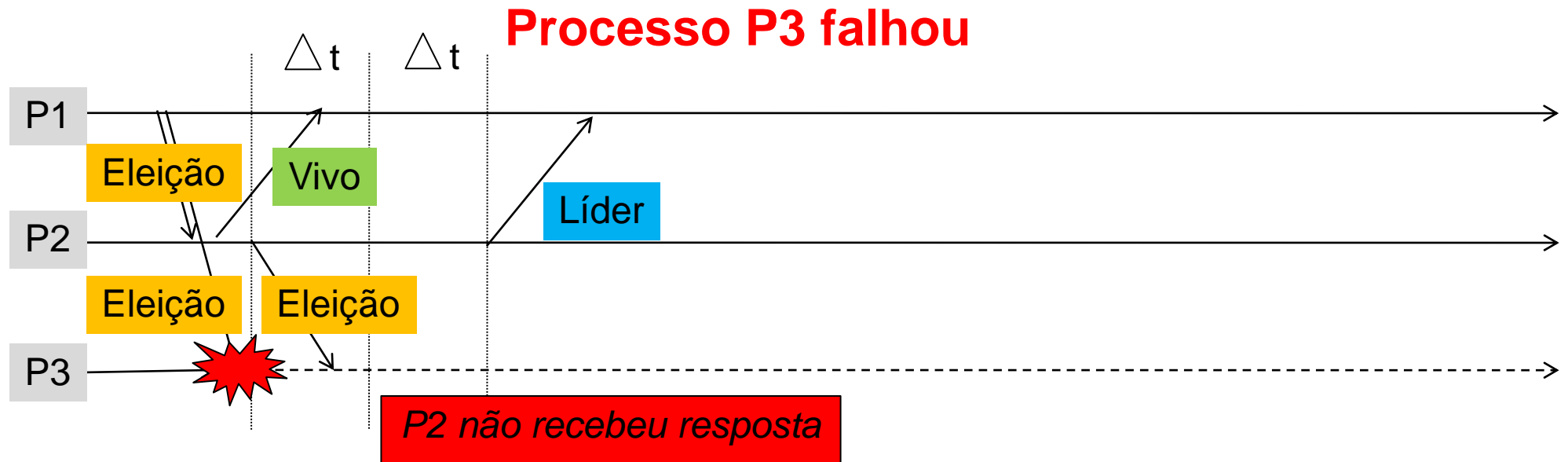
Algoritmo do Valentão (Bully) – Caso com processo falho (1)

Processo P3 falhou



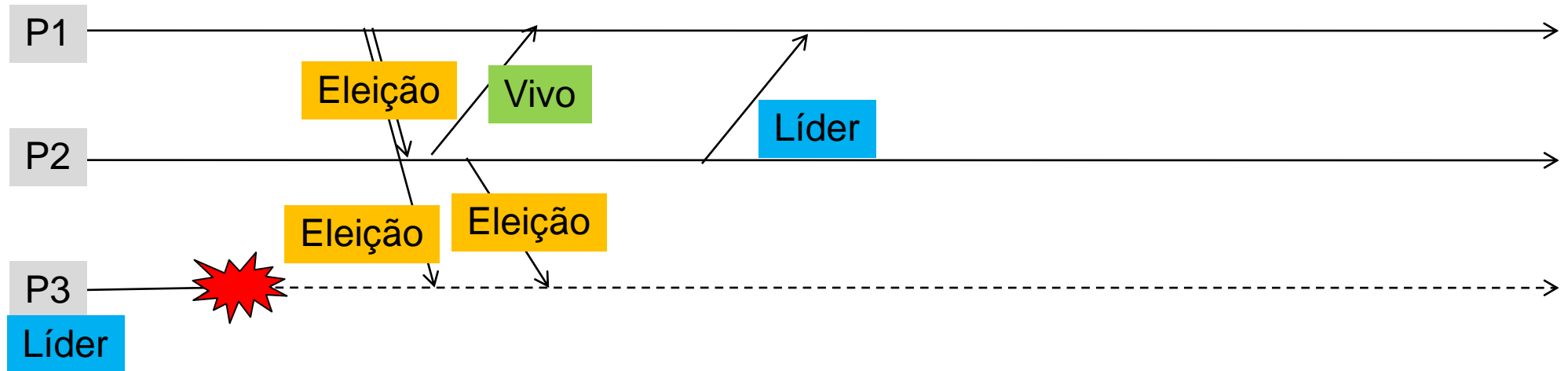
- P2 envia mensagem de eleição para processos com ID maior

Algoritmo do Valentão (Bully) – Caso com processo falho (1)



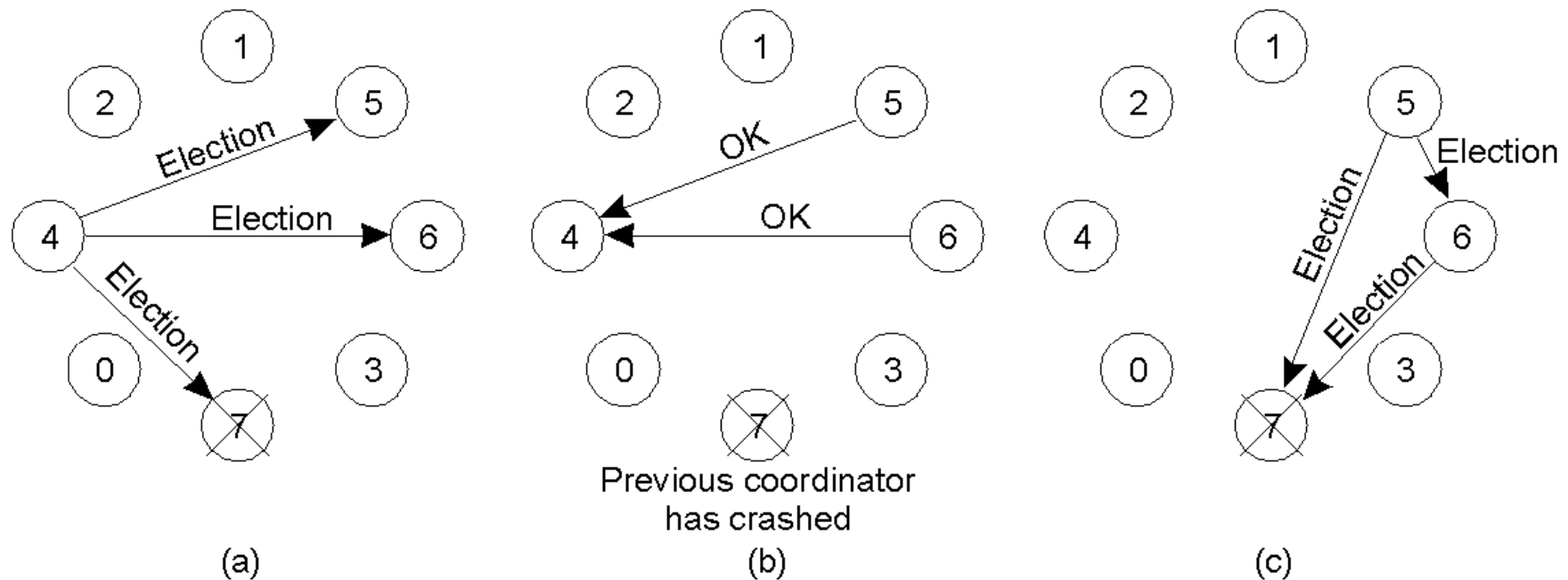
-Não havendo resposta do processo P3, o processo P2 proclama-se líder e envia mensagem informando que é o líder

Algoritmo do Valentão (Bully) – Caso de falha no líder



- Líder atual (processo P3) falha
- Processos eventualmente detectam falha de P3 e iniciam algoritmo de eleição
- Processo de maior ID que não receber mensagens "Vivo" será o novo líder

Algoritmo do Valentão (Bully)



Exemplo extraído do livro do Tanembaun:

- Processo 4 inicia a eleição
- Processos 5 e 6 avisam que estão vivos
- Processos 5 e 6 iniciam eleição
- Após iniciar eleição, processo 6 não recebe mensagem de nenhum outro processo e declara-se líder

Algoritmo do Valentão (Bully) – Considerações

- Processos precisam conhecer outros processos do sistema e seus identificadores
- **Premissa:** Funciona em ambientes síncronos
 - Caso latência das mensagens for desconhecido (sistemas assíncronos) não há garantia que o protocolo atinja o consenso (eleja um líder)
- Algoritmo requer a espera por respostas
 - o desempenho do algoritmo é limitado pelo tempo de espera por mensagens e pelo número de processos no sistema

Análise de complexidade

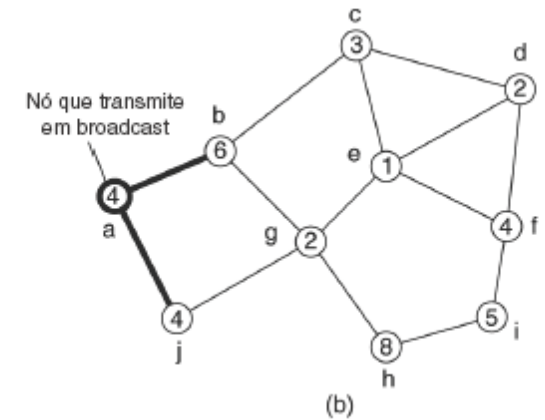
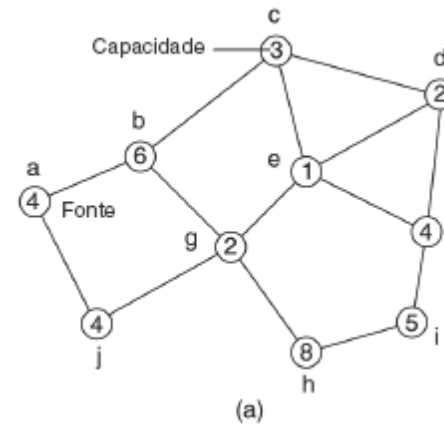
- Algoritmo do valentão
 - Dados n processos:
 - Se o processo com menor prioridade inicia a eleição
 - Então $n-1$ eleições são necessárias (Pradeep diz ser necessárias $n-2$)
 - Cada eleição requer $O(n^2)$ no pior caso
 - Se o processo com maior prioridade inicia a eleição
 - Então requer apenas $n-2$ mensagens
- Algoritmo do anel
 - Eleição sempre requer $2(n-1)$ mensagens
 - $(n-1)$ para votação
 - $(n-1)$ para divulgação do líder

Algoritmo de Eleição em redes sem fio

- Nestes cenários não é realista supor que a entrega de mensagens é confiável ou que a topologia é fixa
- Vasudevan, em 2004 propôs um protocolo para eleição de líder nestes cenários
 - Implemente um protocolo de Árvore de Abrangência (*Spanning Tree*)
 - O líder é eleito baseado em suas características (tempo de bateria, conectividade, etc.)
 - O melhor nodo é escolhido como líder

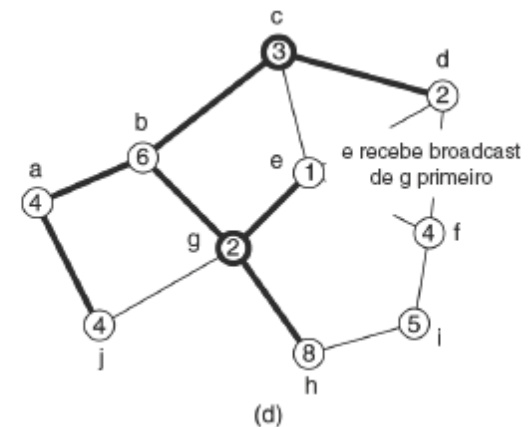
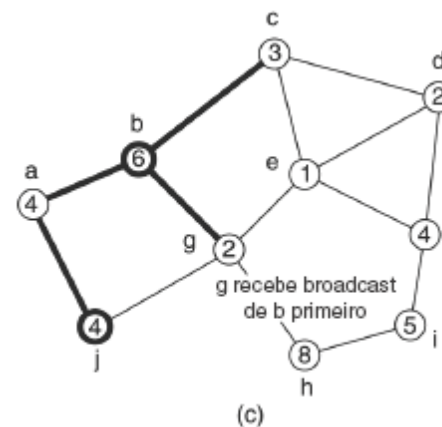
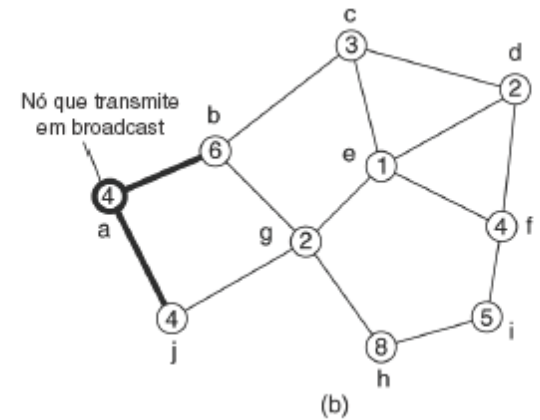
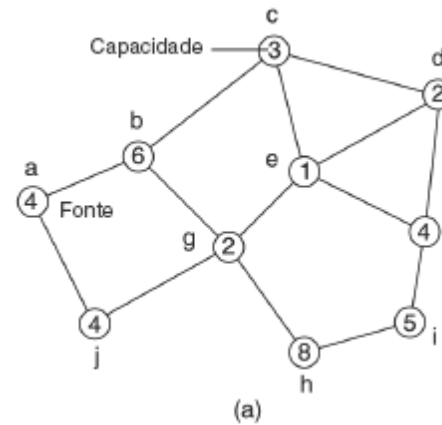
Algoritmo de Eleição em redes sem fio

- Um processo inicia a eleição (Processo **a**)
- **a** envia mensagem ELEIÇÃO a todos os seus vizinhos
- os vizinhos marcam **a** como processo nó pai e propagam a mensagem eleição aos seus vizinhos



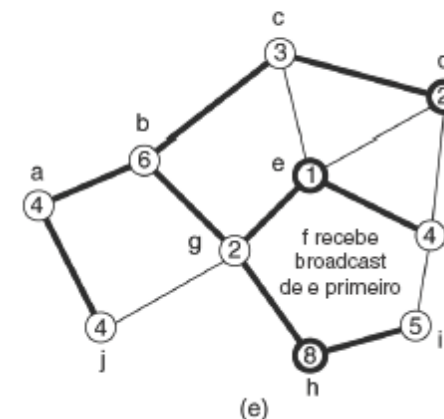
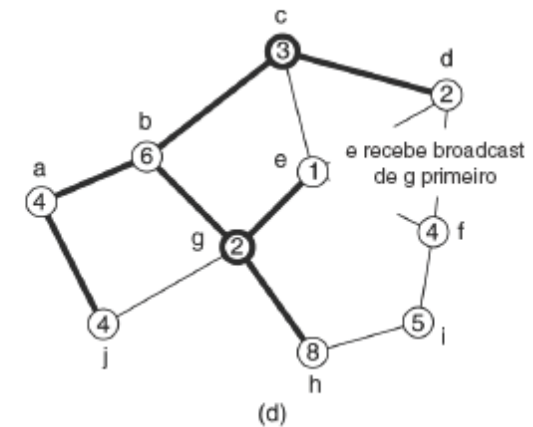
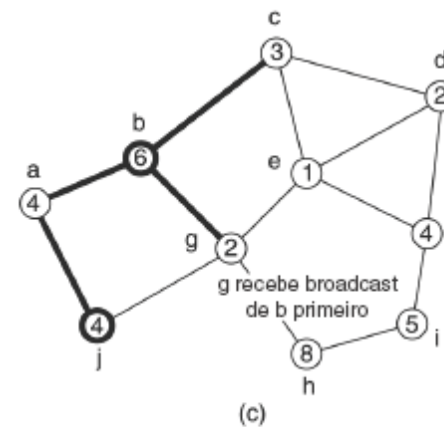
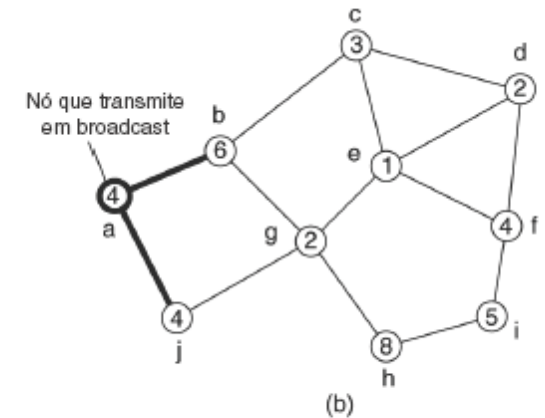
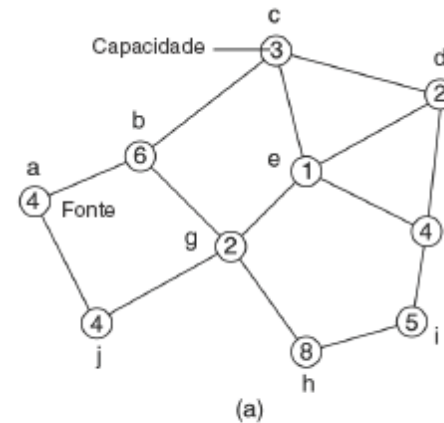
Algoritmo de Eleição em redes sem fio

- Um processo inicia a eleição (Processo **a**)
- **a** envia mensagem ELEIÇÃO a todos os seus vizinhos
- os vizinhos marcam **a** como processo nó pai e propagam a mensagem eleição aos seus vizinhos
- processos que já tenham um pai, ao receberem uma mensagem ELEIÇÃO, apenas informam que já possuem um processo pai



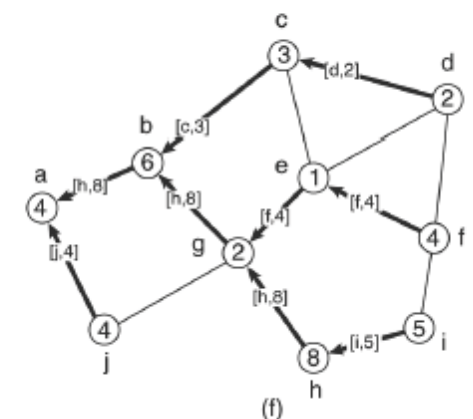
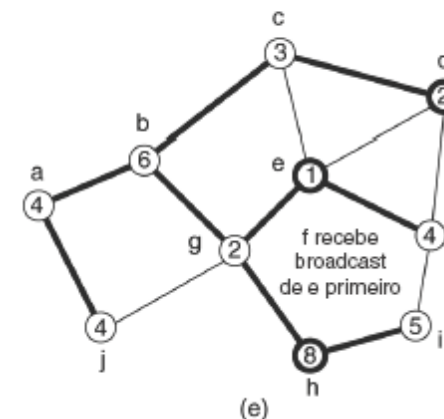
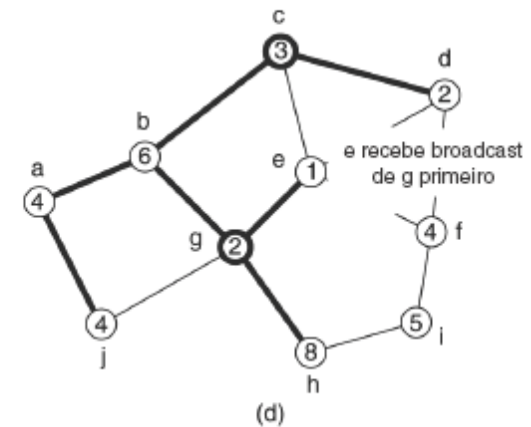
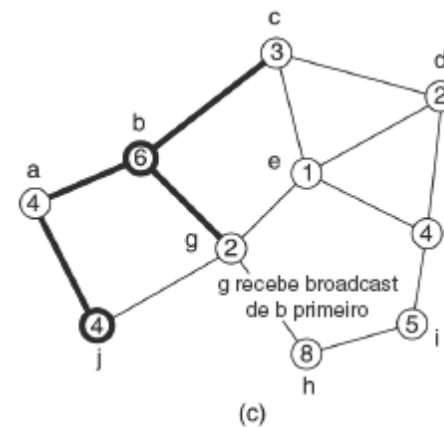
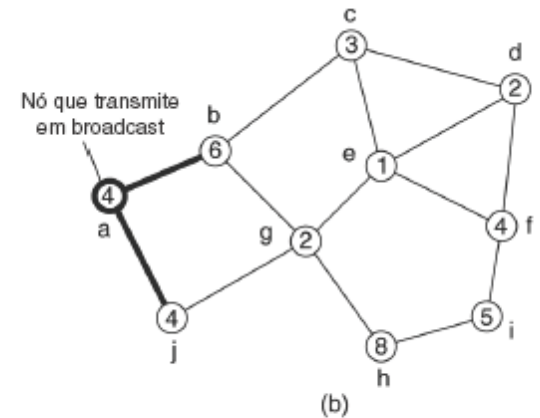
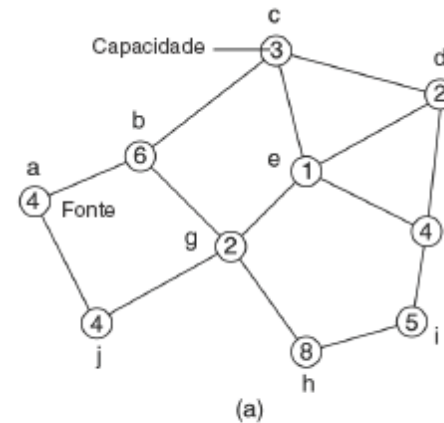
Algoritmo de Eleição em redes sem fio

- Um processo inicia a eleição (Processo **a**)
- **a** envia mensagem ELEIÇÃO a todos os seus vizinhos
- os vizinhos marcam **a** como processo nó pai e propagam a mensagem eleição aos seus vizinhos
- processos que já tenham um pai, ao receberem uma mensagem ELEIÇÃO, apenas informam que já possuem um processo pai
- Uma árvore é criada



Algoritmo de Eleição em redes sem fio

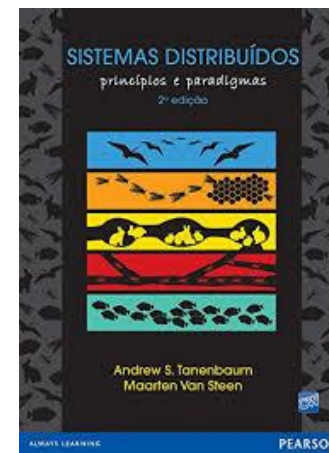
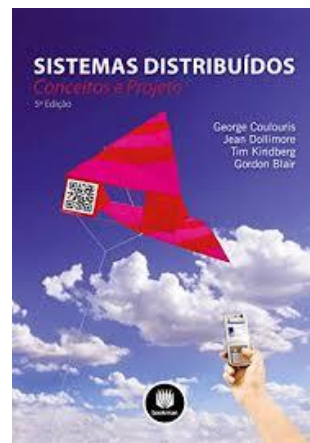
- Um processo inicia a eleição (Processo **a**)
- **a** envia mensagem ELEIÇÃO a todos os seus vizinhos
- os vizinhos marcam **a** como processo nó pai e propagam a mensagem eleição aos seus vizinhos
- processos que já tenham um pai, ao receberem uma mensagem ELEIÇÃO, apenas informam que já possuem um processo pai
- Uma árvore é criada
- Partindo das folhas, cada nodo informa ao seu pai suas informações de capacidades
- O nodo pai compara as informações recebidas pelos seus filhos e repassa para seu pai apenas o melhor nodo
- Finalmente o nodo raiz (pai de todos) indica através de difusão pela rede qual será o líder



Referências

Parte destes slides são baseadas em material de aula dos livros:

- *Coulouris, George; Dollimore, Jean; Kindberg, Tim; Blair, Gordon. Sistemas Distribuídos: Conceitos e Projetos. Bookman; 5ª edição. 2013.*
- *Tanenbaum, Andrew S.; Van Steen, Maarten. Sistemas Distribuídos: Princípios e Paradigmas. 2007. Pearson Universidades; 2ª edição.*



- *Imagens e clip arts diversos: <https://free-icon-rainbow.com/> , <https://www.gratispng.com/>*