ELEIÇÃO DO LÍDER

- **Processo Líder**:
  - Muitas aplicações distribuídas requerem que um processo aja como coordenador, iniciador ou faça algum papel especial
  - Caso seja designado um nó específico para esta função
    - Aplicação é paralisada se houver falha do nó
  - **Escolha do Líder**
    - Deve ser feita automaticamente
    - Em geral, Líder é o nó que tem maior ID
    - Em caso de falha, um novo nó deve ser o Líder

ALGORITMO BULLY

- Algoritmo **bully** (valentão) foi proposto por Garcia-Molina (1982):
  - Quando qualquer nó *P* nota que o **líder não responde**, ele convoca uma eleição:
    - *P* envia uma mensagem **ELEIÇÃO** a todos os nós com ID mais alto do que ele
    - Se **nenhum responder**, ***P* vence** a eleição e se torna líder
      - *P* envia uma mensagem, indicando ser líder

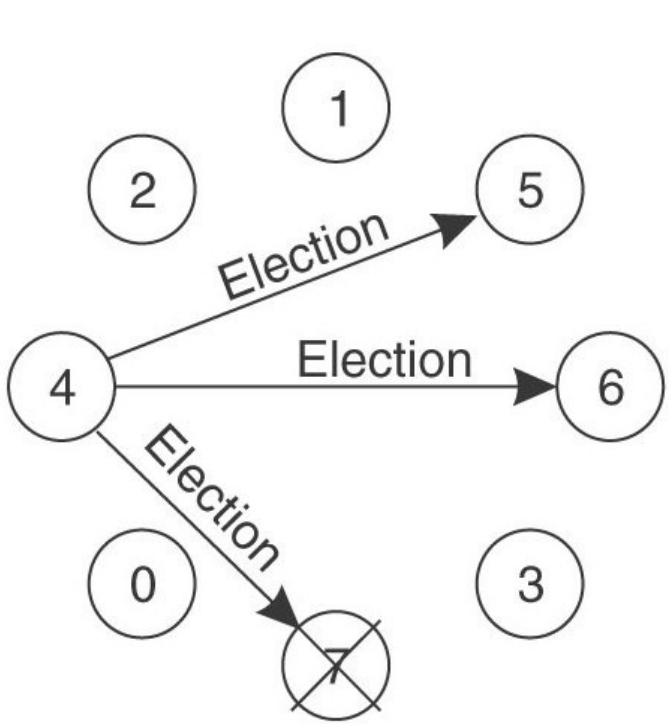
- Se um nó *Q* **responder**:
  - A eleição convocada por *P* termina
  - O nó *Q* convoca uma eleição enviando para a mensagem para os nós mais altos.

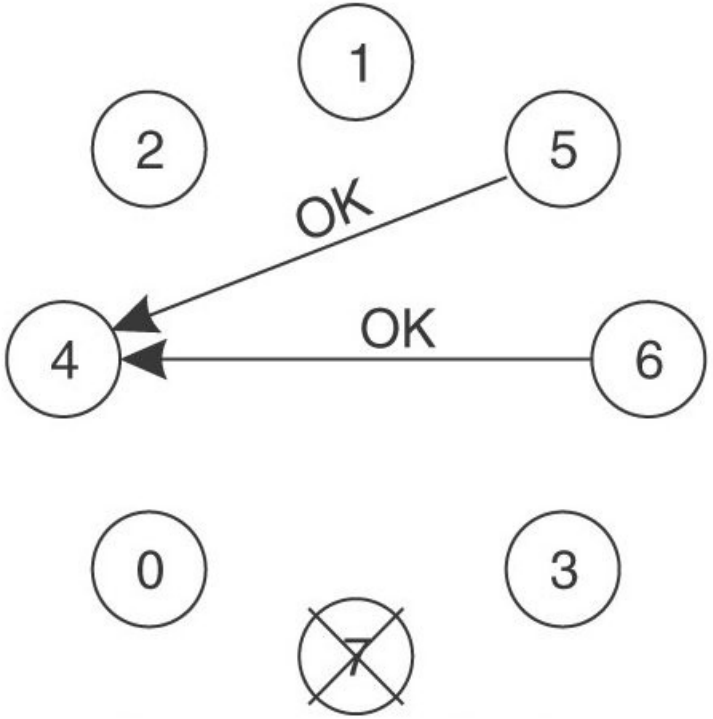
- A eleição é sucessivamente chamada até que algum nó não receba mais respostas
  - O nó que não recebe resposta é o nó com o identificador mais alto em atividade

- Caso um nó *R* volte a atividade:
  - Ele convoca uma eleição e assume o comando, se tiver ID maior

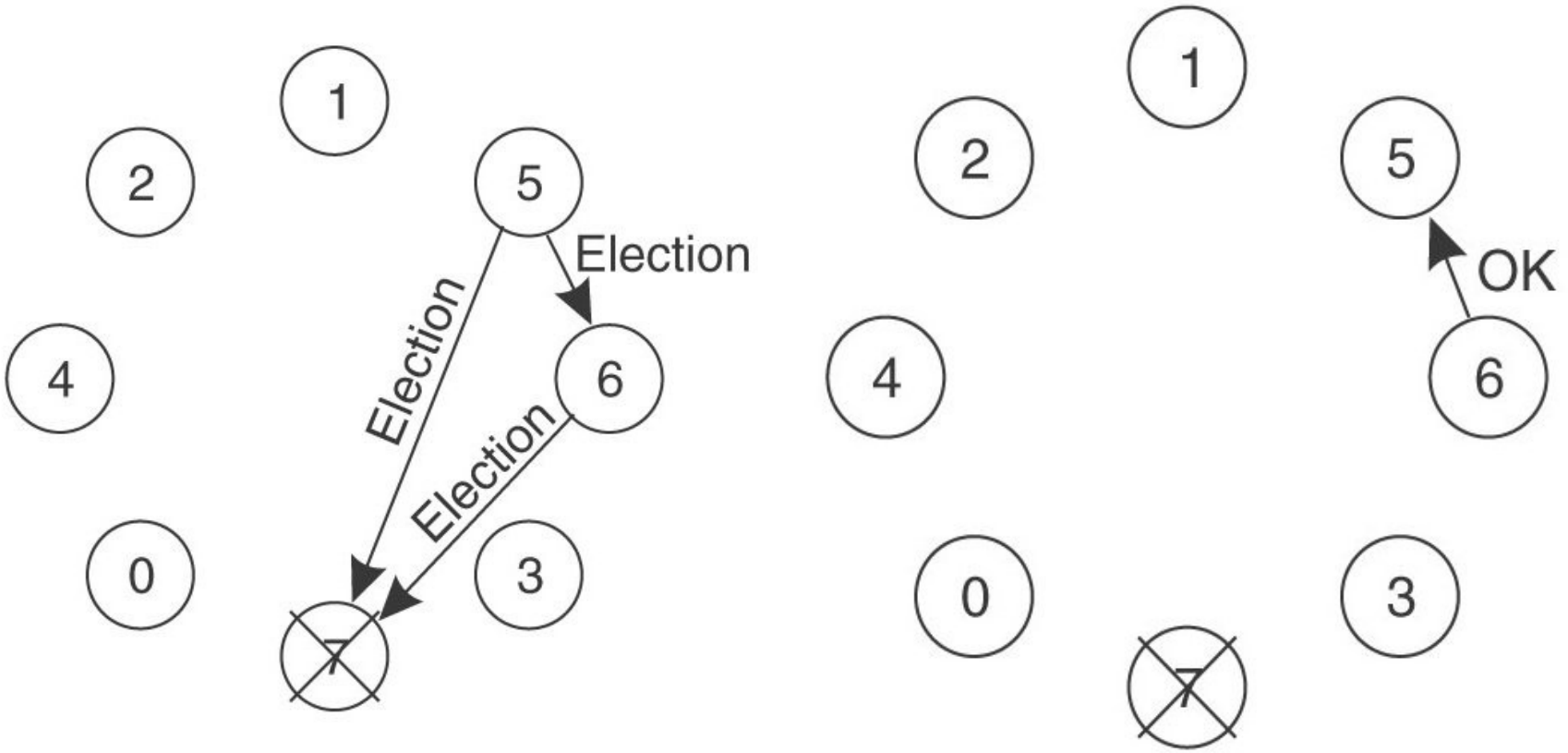
Exemplo: Um sistema com 7 nós. O líder 7 falha:

1. O nó 4 detecta que 7 falhou e convoca eleição
   Nós 5 e 6 respondem

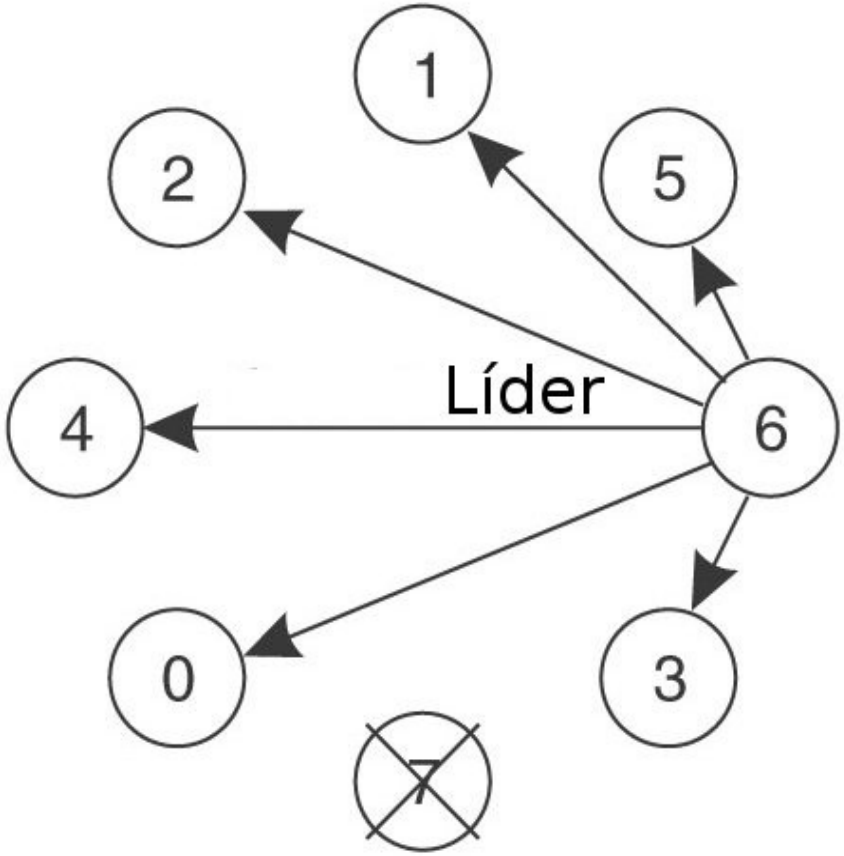




2. Nós 5 e 6 convocam eleição para os nós maiores
   - Nó 6 responde a eleição de 5



3. O nó 6 se declara líder para demais nós



ALGORITMO DE ANEL

- O Algoritmo *bully* admite que os nós estão totalmente interconectados

- Outros algoritmos são mais eficientes em topologias específicas da rede
  - **No pior caso**, o algoritmo bully gasta $O(n^3)$ mensagens

Algoritmo de Chang-Roberts

- Considera uma topologia em anel unidirecional
- Um nó que detecta a falha do líder inicia uma eleição:
  - Envia mensagem para o próximo nó com o seu ID
  - Se um **nó recebe uma mensagem** de eleição:
    - Com **valor menor** do que o seu próprio ID
      - Substitui o valor pelo seu ID
      - Envia para o próximo nó
    - Com **valor maior** do que o seu próprio ID
      - Repassa a mensagem para o próximo nó
    - Com **valor igual** ao seu próprio ID
      - O nó é o líder. Envia mensagem avisando.
- **No pior caso**, o algoritmo gasta $O(n^2)$ mensagens

Algoritmo de Franklin

- Considera uma topologia em anel bidirecional
- Os nós estão dispostos em qualquer ordem no anel
- Cada nó possui associado um cor:
  - Nó com **cor vermelha**: apto a ser líder e iniciar eleição
  - Nó com **cor preta**: apenas repassa a mensagem de eleição
- Inicialmente, todos os nós têm cor vermelha

- Na **eleição**, cada nó:
  - Manda mensagem de eleição para os seu dois vizinhos, com o seu ID
  - Aguarda mensagem de seus dois vizinhos
  - Efetua a **análise das mensagens**:
    - Se houver um ID maior que o seu, torna-se preto
    - Se os IDs forem menores, passa para a **próxima rodada**, repetindo as mensagens para os vizinhos
    - Se os ID forem iguais ao seu, torna-se líder

- Complexidade do algoritmo de Franklin: $O(n \log n)$ mensagens

Exemplo: Após a primeira rodada, restam os nós 2, 7 e 9.
Na segunda rodada resta apenas o nó 9, que se declara líder