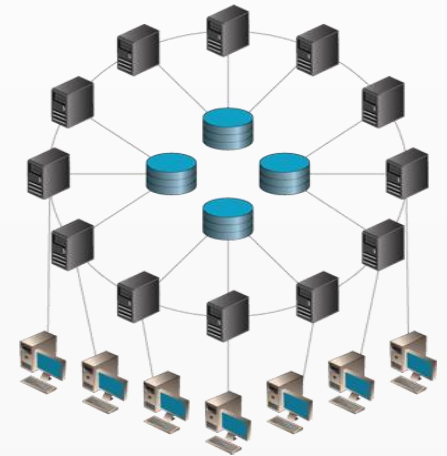


Algoritmos de Eleição

# SMD0050 - SISTEMAS DISTRIBUÍDOS

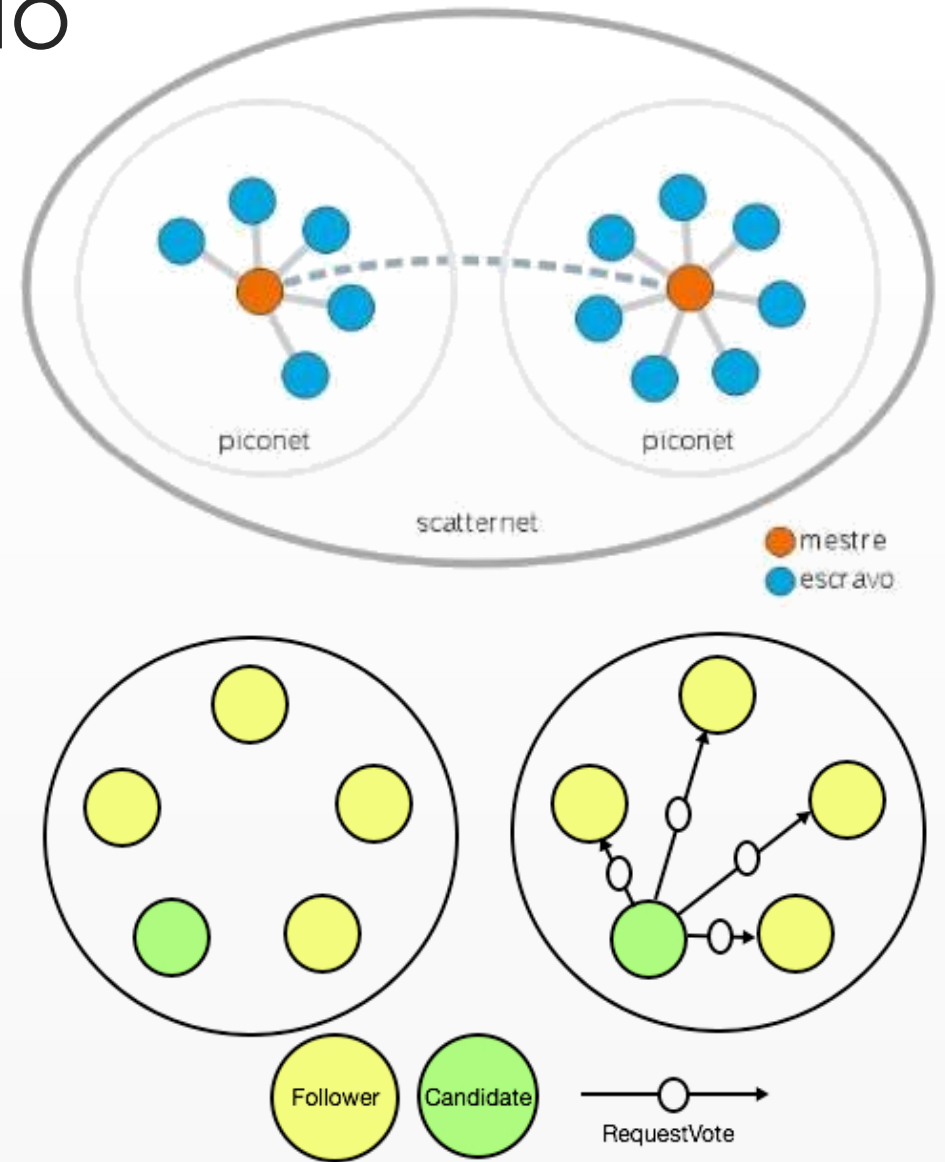
1



Slides são baseados nos slides do Coulouris e Tanenbaum

# Algoritmos de Eleição

- Usados quando há necessidade de um ou mais nós devem agir como coordenador.
- Coordenar a replicação, difusão, agregação
  - Formação de Scartternet em Bluetooth
- Iniciar/coordenar um processo de resolução de consenso
  - Paxos, RAFT



# Exemplos de Algoritmos de Eleição

- De propósito geral
  - Algoritmo do valentão
  - Algoritmo de anel
- Soluções para ambientes específicos
  - Algoritmos para Ad Hoc (Bluetooth Topology Construction)
  - Algoritmos para sistemas de grande escala (Blockchain)



# Requisitos importantes

- **Safety** - Todos os participantes precisam chegar a um **consenso** ou serem informados de quem foi o escolhido
- **Liveness** – Em algum momento, deve-se definir o coordenador
- **Stability** – Devem ser evitadas reeleições desnecessárias
- Nem sempre é possível saber o número total de processos
  - inundação, topologia em anel



**requisitos**

# Requisitos importantes

- A eleição é iniciada como reação a uma **detecção de falha** do antigo coordenador
- Pode haver falha durante a escolha do novo coordenador
- Múltiplas eleições podem ocorrer em paralelo mas devem chegar a um mesmo resultado



**requisitos**

# Tarefa 1- Vamos tentar criar o nosso!

Não vale pescar!

Processos podem ser identificados pelo seu IP e PID



Processos tem a lista de IPs de todos os membros

O processo de PID 2 detectou que o coordenador não está mais ativo



Tô pensando em ser o coordenador!



Todas as máquinas ativas são capazes de serem o coordenador

# Algoritmo do Valentão (Bully)

- Inventado por Garcia-Molina (1982)
- Todos nós possuem um identificador
- Sistema síncrono com falhas tipo fail-stop\*, baseado na difusão de mensagens



É eleito o nó com maior identificador que está ativo

\*fail-stop – processo “cai” e isso é detectável por parceiros



# Algoritmo do Valentão

- Sempre que um nó qualquer  $P$  nota que o coordenador não responde,  $P$  inicia uma eleição:
  1.  $P$  envia uma mensagem ELEIÇÃO a todos os processos de números mais altos;
  2. Se nenhum responder (sistema síncrono),  $P$  vence a eleição e se torna o coordenador;
  3. Se um dos processos de número mais alto responder, ele toma o poder e o trabalho de  $P$  está concluído.

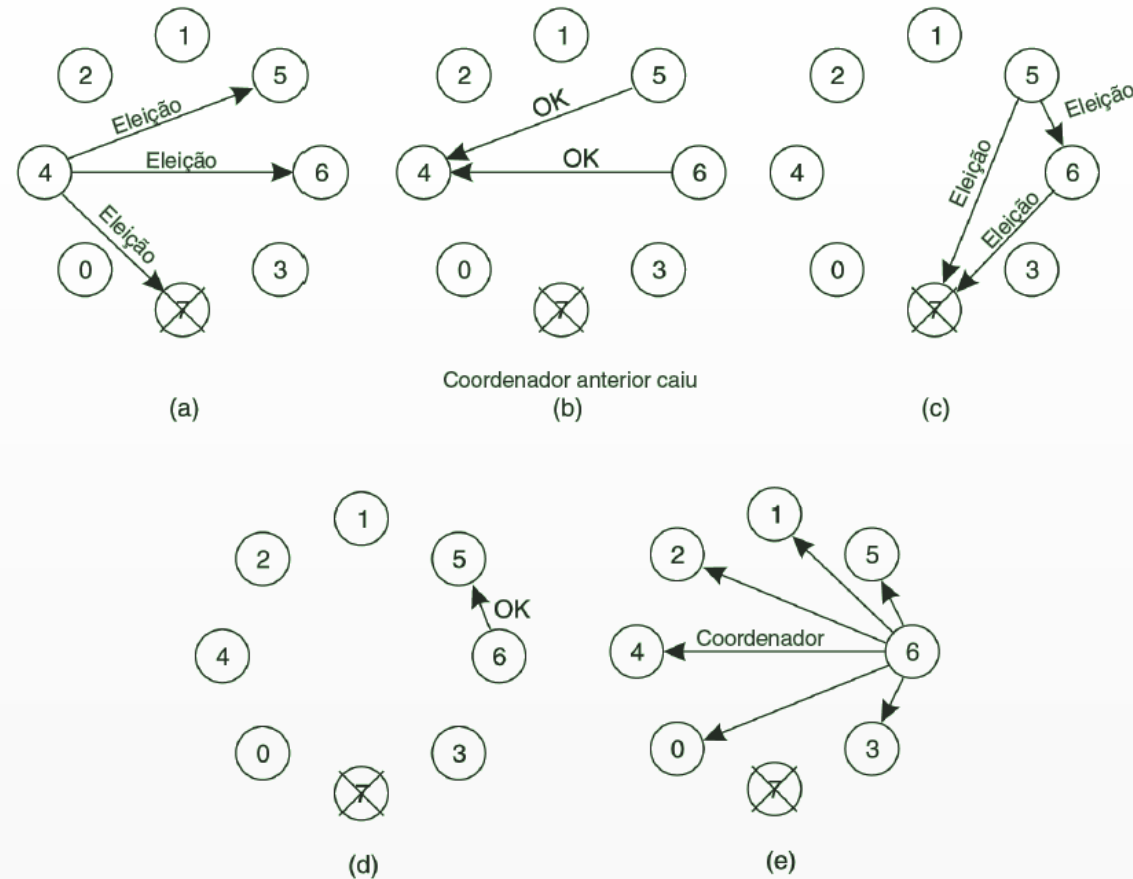


# Algoritmo do Valentão - Suposições

- Sistema Síncrono - Toda mensagem é entregue em  $T_m$  unidades de tempo após o seu envio
- Todos os processos não falhos respondem a todas as mensagens recebidas em  $T_p$  unidades de tempo
- Definição de um detector de falhas confiável:

Se um processo não responde em  $2T_m + T_p$  unidades de tempo, ele falhou

# Algoritmo do Valentão



**Figura 6.19** Algoritmo de eleição do valentão. (a) O processo 4 convoca uma eleição. (b) Os processos 5 e 6 respondem e mandam 4 parar. (c) Agora, cada um, 5 e 6, convoca uma eleição. (d) O processo 6 manda 5 parar. (e) O processo 6 vence e informa a todos.

# Vantagens e Desvantagens?

# Vídeo

- [https://www.youtube.com/watch?v=K44x\\_VQmUs8](https://www.youtube.com/watch?v=K44x_VQmUs8)





## Algoritmo do Anel - LCR


- Proposto por Le Lann, Chang e Roberts
- Baseado na utilização de anel (físico ou lógico)

Não usa ficha!

- Quando qualquer processo nota que o coordenador não está funcionando, monta uma mensagem ELEIÇÃO com seu próprio número (ex: PID+IP) e o envia a seu sucessor ou ao próximo que esteja em funcionamento



# Algoritmo do Anel

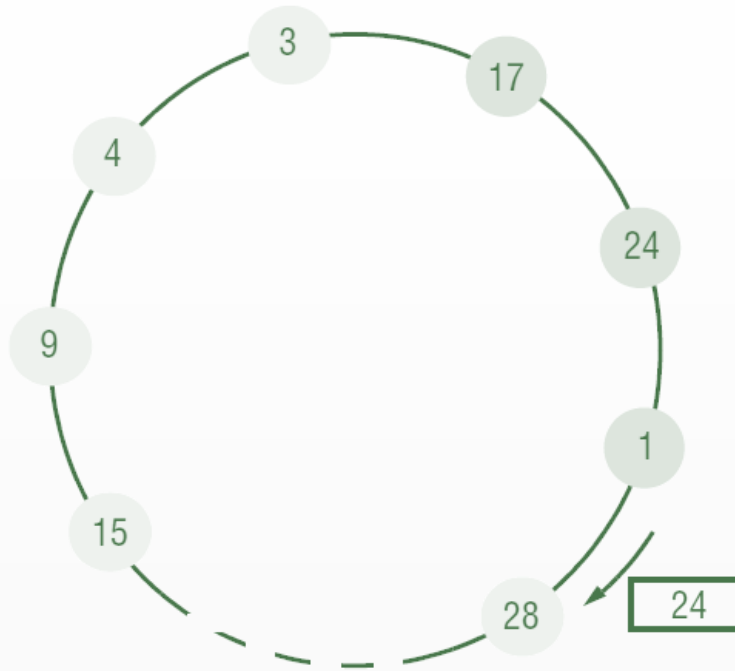
- Se o nó que recebe a mensagem de eleição tem um identificador maior que o informado na mensagem que recebeu, passa uma mensagem de eleição para seu vizinho da direita com seu próprio identificador.
  - Caso contrário aceita que o nó que tem o identificador contido na mensagem será o líder e repassa ao seu vizinho da direita.
- 



# Algoritmo do Anel

- A eleição termina quando
- Se o nó recebe uma mensagem com o identificador idêntico ao seu, ele se declara LÍDER
  - envia essa mensagem pelo Anel
- Este evento só ocorre quando a mensagem contendo o maior identificador circulou por todo o anel tornando todos os seus membros cientes do resultado.

# Algoritmo de eleição por Anel



*Note:* The election was started by process 17. The highest process identifier encountered so far is 24. Participant processes are shown in a darker tint.



## Tarefa 2

- Existem tem processos P que a cada 30s imprimem o nome do coordenador
- Vamos implementar um algoritmo de eleição que eleja qual será o coordenador
- Vamos usar RMI na implementação
  - Crie uma Interface Remote chamada P com o método startElection() e setLeader()
- Crie uma variável PID em cada processo
  - Use  
ManagementFactory.getRuntimeMXBean().getName()

**In Java 9** the new process API can be used:

```
long pid = ProcessHandle.current().pid();
```



## Tarefa 2

- **Fase de descoberta:** Existem 3 instâncias de P que alguns instantes após as suas inicializações devem encontrar seus pares numa rede
  - Usem o RMI Registry para tal
  - Ao final da busca, cada P tem um Array com as 3 instâncias de P e seus respectivos PIDs
- Em seguida, usando um algoritmo de eleição do Valentão, escolha qual deles é o coordenador
  - Cada processo inicia uma eleição em um tempo aleatório (30s a 60s)
  - Use os métodos `setElection` para comunicar a eleição e `setLeader` para se declarar como vencedores

# Tarefa de Casa – 18/05/2018

- Modifique o algoritmo anterior para que cada participante cheque se o coordenador está vivo, caso não esteja uma nova eleição é iniciada
- Escreva um pequeno artigo de 2 páginas descrevendo a solução de eleição do RAFT e compare sua implementação

