

# Sistemas Distribuídos



## Comunicação em Grupo

## Referência



- “Sistemas operacionais modernos” Andrew S. TANENBAUM Prentice-Hall, 1995
  - Seção 10.4 pág. 304 - 311

## Comunicação em Grupo

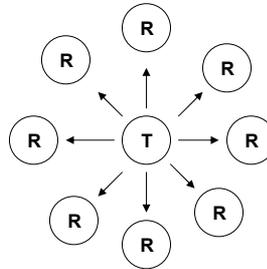
- Suponha que se deseja um serviço de arquivos único e tolerante a falha → implementado com um grupo de servidores de arquivo
- Cliente deve enviar mensagens a TODOS os servidores para ter certeza que a requisição foi executada
- Com RPC deveríamos repetir a chamada para cada um dos servidores

## Comunicação em Grupo

- Grupo: conjunto de processos que agem juntos.
- Quando uma mensagem é enviada ao grupo TODOS os membros do grupo devem recebê-la
  - comunicação um-para-muitos



Comunicação  
ponto-a-ponto



Comunicação  
um-para-muitos

## Comunicação em Grupo

- Os grupos são dinâmicos:
  - novos grupos podem ser criados
  - grupos antigos podem ser destruídos
  - um processo pode se juntar a um grupo
  - um processo pode sair de um grupo
  - um processo pode ser membro de vários grupos ao mesmo tempo
- São necessários mecanismos para gerenciar os grupos

## Implementação

- Implementação dependente do hardware:
  - Multicasting: atribui-se um endereço especial ao grupo que é enxergado por várias máquinas. Quando o pacote é enviado para este endereço ele é automaticamente enviado a todas as máquinas com acesso ao endereço.
  - Broadcasting: os pacotes contendo um endereço especial são enviados para todas as máquinas. Cada máquina deve verificar os pacotes de broadcast para determinar se o pacote é efetivamente destinado a ela (sobrecarga).

7

## Implementação (cont)

- Unicasting: o transmissor envia um pacote para cada membro do grupo. Se existem  $N$  membros no grupo são enviados  $N$  pacotes.
  - Funciona para grupos pequenos.

8

## Aspectos de projeto

- A comunicação em grupo é semelhante em alguns aspectos a troca de mensagens comum (bufferização, bloqueio etc)
- Outros aspectos importantes a serem considerados na comunicação em grupo:
  - Grupos fechados x grupos abertos
  - Grupos igualitários x grupos hierárquicos
  - Controle de membros de um grupo
  - Endereçamento de grupo
  - Atomicidade
  - Ordenação das mensagens
  - Grupos sobrepostos
  - Escalabilidade

9

## Grupos fechados x grupos abertos

- Grupos fechados: somente os membros do grupo podem enviar mensagens para o grupo. Ex.: processamento paralelo
  - Processos que não pertençam ao grupo podem enviar mensagens aos membros do grupo individualmente.
- Grupos abertos: qualquer processo pode enviar mensagens ao grupo. Ex.: servidores replicados.

10

## Grupos igualitários x grupos hierárquicos

- Grupos igualitários: todos os processos são iguais, nenhum deles é superior. Todas as decisões são tomadas coletivamente.
  - Vantagem?
    - é simétrico. Não existe um único ponto de falha. Se um dos processos falhar o grupo fica menor mas continua trabalhando.
  - Desvantagem?
    - a tomada de decisão é complexa. É necessária uma votação (sobrecarga)

11

## Grupos igualitários x grupos hierárquicos

- Grupos hierárquicos: existe uma hierarquia, por exemplo, um processo é o coordenador e dirige o trabalho dos demais.
  - Vantagem?
    - a tomada de decisão executada pelo coordenador é mais simples.
  - Desvantagem?
    - se o coordenador falhar o grupo pára o trabalho.

12

## Controle de membros de um grupo

- Gerência de grupo: controla a criação e eliminação de grupos e inclusão e exclusão de membros no grupo.
- Servidor de grupo: processo que mantém uma base de dados com todos os grupos e seus membros.
  - Solução centralizada: único ponto de falha

13

## Controle de membros de um grupo

- Solução distribuída:
  - Grupo aberto: um novo processo anuncia sua presença no grupo.
  - Grupo fechado: deve funcionar como grupos abertos para adesão de novos membros.
  - Para deixar o grupo basta o processo anunciar sua saída.

14

## Controle de membros de um grupo

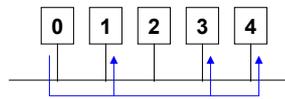
- Problemas:
  - Se um membro do grupo sai do ar? Ele deixa o grupo mas não anuncia sua saída. Os membros devem descobrir que o membro não está respondendo.
  - O ato de deixar ou se juntar a um grupo deve ser síncrono com as mensagens que estão sendo enviadas.
  - Se um grande número de máquinas sair do ar?
    - O grupo deve ser reconstituído. Alguém deve iniciar o processo de reconstituição do grupo. E se 2 processos tentarem ao mesmo tempo?

15

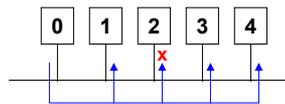
## Endereçamento de grupo

- Para enviar uma mensagem a um grupo o processo transmissor deve especificar qual é o grupo a que está se referindo.
  - Multicasting: o endereço do grupo é associado a um endereço de multicast.
  - Broadcasting: a mensagem é enviada em broadcast e o kernel se encarrega de descartar a mensagem se não houver um membro do grupo.
  - O kernel da máquina do processo transmissor se encarrega de fazer unicasts a cada um dos processos do grupo

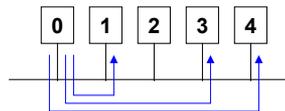
16



**1 multicast**



**1 broadcast**



**3 unicasts**

## Endereçamento de grupo

- Outra maneira de endereçar: o transmissor fornece uma lista do processos do grupo.
- Terceira solução: endereçamento com predicado. A mensagem é enviada contendo uma expressão booleana (predicado) a ser avaliada (ex.: número da máquina, valor de variável etc). Se for verdadeiro a mensagem é aceita. Caso contrário é descartada.
  - Ex.: enviar mensagens para máquinas que tenham no mínimo 4M de memória livre.

## Atomicidade

- Ou broadcast atômico
- Tudo-ou-nada: as mensagens devem chegar a TODOS os membros do grupo ou a nenhum → atomicidade
- Implementação não é simples: pode haver falha na comunicação.
  - Solução: mensagem de confirmação.
- E se uma máquina falhar?

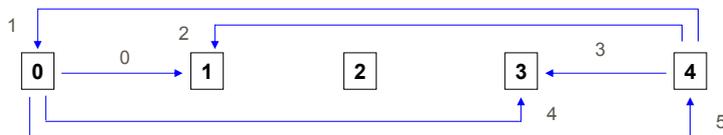
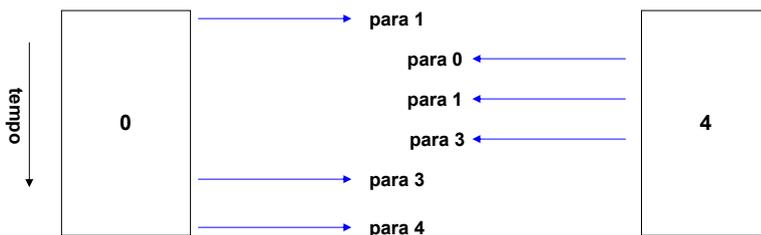
## Atomicidade

- Algoritmo de Joseph e Birman:
  - O transmissor envia a mensagem a todos os membros do grupo e inicia um temporizador para permitir que as retransmissões sejam enviadas se necessário
  - Quando o receptor receber uma mensagem nova deve retransmiti-la a todos os membros do grupo. Se já conhecer a mensagem é só descartá-la.

## Ordenação de mensagens

- A comunicação em grupo deve atender a propriedade de ordenação de mensagens.
- Considere que:
  - Existam 5 máquinas
  - Os processos 0, 1, 3 e 4 pertençam ao mesmo grupo
  - Os processos 0 e 4 desejam simultaneamente enviar mensagem.
  - Não há multicast nem broadcast
  - Os dois processos competem pela rede local

21

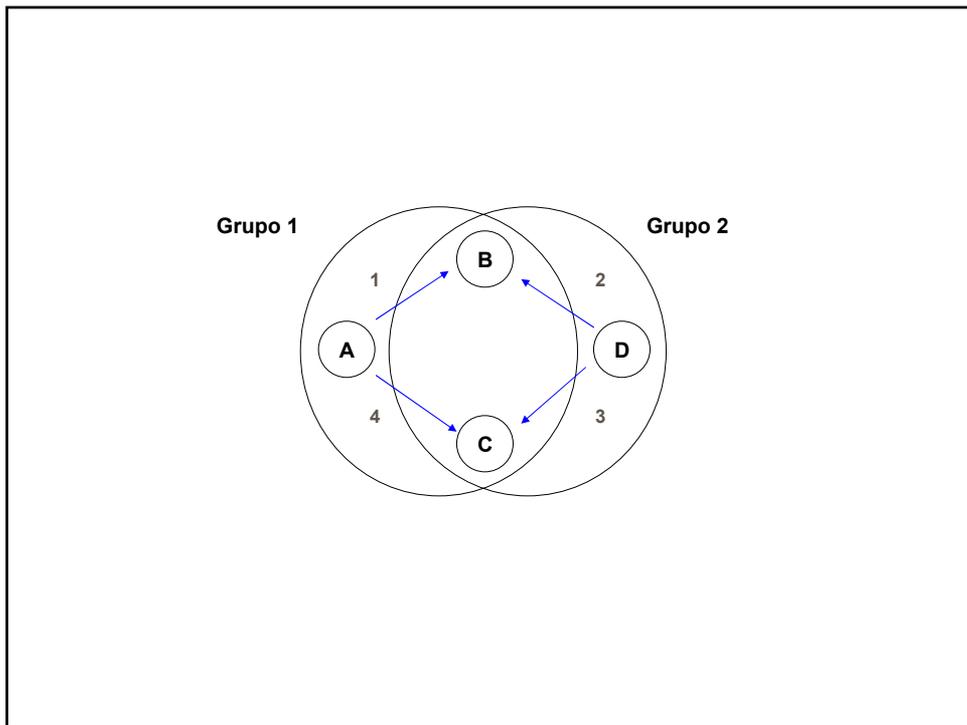


## Ordenação de mensagens

- Ordenação de tempo global: as mensagens são expedidas a todos os membros do grupo na ordem exata em que foram enviadas.
- Ordenação consistente no tempo: se duas mensagens foram enviadas quase ao mesmo tempo, o sistema considera uma delas como sendo a primeira e envia ao grupo.

## Grupos sobrepostos

- Um processo pode ser membro de vários grupos ao mesmo tempo
- Considere:
  - Dois grupos:
    - Grupo 1: A, B e C
    - Grupo 2: B, C e D
  - A e D tentam enviar uma mensagem simultaneamente a seus respectivos grupos

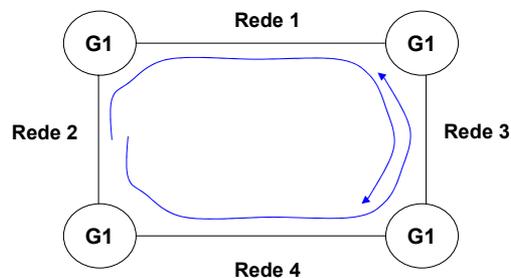


## Grupos sobrepostos

- Apesar de haver ordenação dentro do grupo não há uma coordenação entre grupos.
- Implementação da ordenação no tempo entre grupos diferentes é complexa.

# Escalabilidade

- Se os grupos possuem muitos elementos?
- E se eles estiverem distribuídos em várias redes com gateways entre eles?
- Os gateways devem ou não passar um multicast adiante?
- Podem existir pacotes nas diferentes redes ao mesmo tempo.



# Exercícios



- “Sistemas operacionais modernos” Andrew S. TANENBAUM Prentice-Hall, 1995
  - Capítulo 10 Exercícios 16 - 18 (pág. 315)