




Sistemas Distribuídos

Grupos

Edeyson Andrade Gomes

www.edeyson.com.br



Roteiro da Aula

Roteiro da Aula

- ▶ Definição de Grupos
- ▶ Tipos
- ▶ Atomicidade
- ▶ Ordenamento

Comunicação em Grupo

- ▶ **RPC**
 - ▶ Comunicação entre Pares
 - ▶ Cliente - Servidor
- ▶ **Comunicação entre vários processos?**
 - ▶ Coleção de servidores de arquivos
 - ▶ Visão única
 - ▶ Tolerância a Falhas
- ▶ **Grupo de Processos**
 - ▶ RPC não suporta
 - ▶ RPC separado para cada PAR
 - ▶ Comunicação ponto-a-ponto

Comunicação em Grupo

- Mensagens enviadas para o grupo são recebidas por todos os membros do grupo
 - Comunicação de um para muitos
- Grupos são entidades dinâmicas
 - Novos grupos podem ser criados e grupos existentes podem ser destruídos
 - Um processo pode entrar e sair de um grupo
 - Inscrição
 - Um processo pode pertencer a mais de um grupo ao mesmo tempo
 - Necessidade de gerência de grupos e membros
 - Abstração
 - Encapsula detalhes do grupo

Comunicação em Grupo

○ Implementação de comunicação em grupo

- Depende do hardware e da rede de comunicação

○ Multicast

- Endereço de rede Especial
- Máquinas **escutam** endereço especial
- Fácil de implementar
 - Cada grupo com um endereço especial

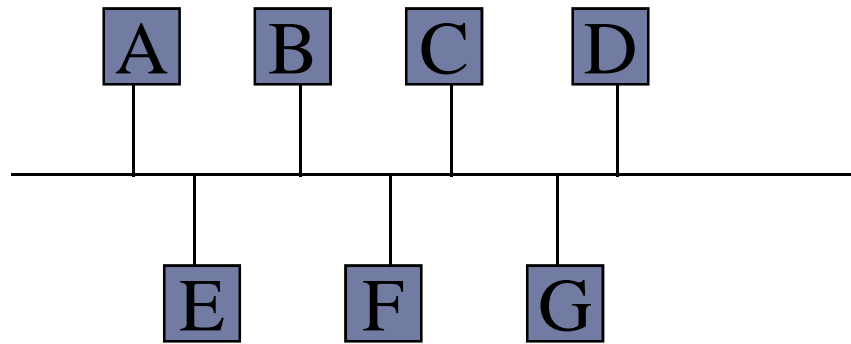
○ Broadcast

- Envio a todas as máquinas
- Máquinas devem validar o destino da mensagem

○ Unicast

- Pacote para máquina específica
- Usado em grupos pequenos

Comunicação em Grupo

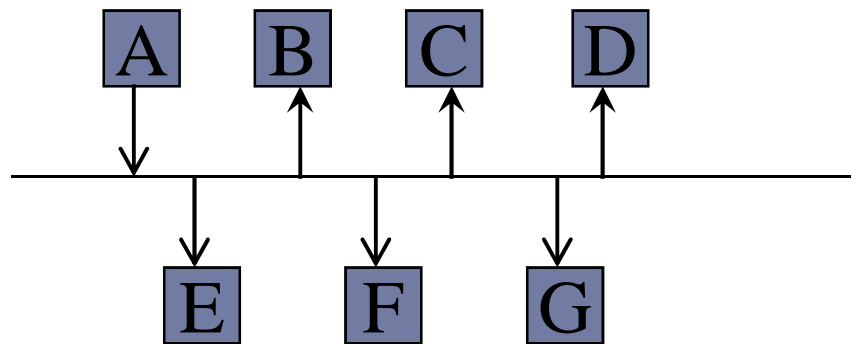


G1: A, C, E

G2: A, D, F

G3: A, B, E, F, G

Comunicação em Grupo



G1: A, C, E

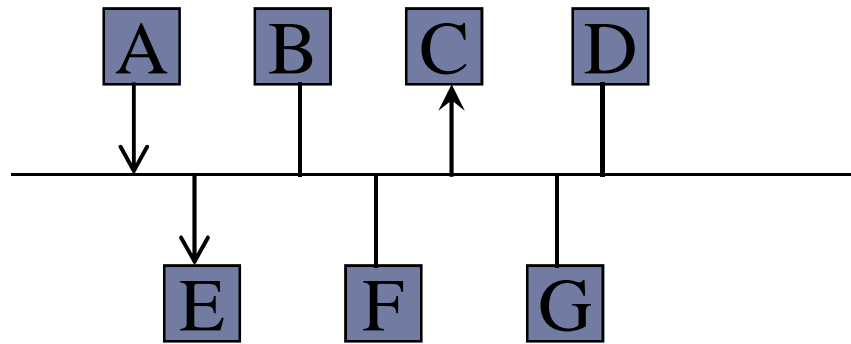
G2: A, D, F

G3: A, B, E, F, G

Broadcast

A envia MSG para G1

Comunicação em Grupo



G1: A, C, E

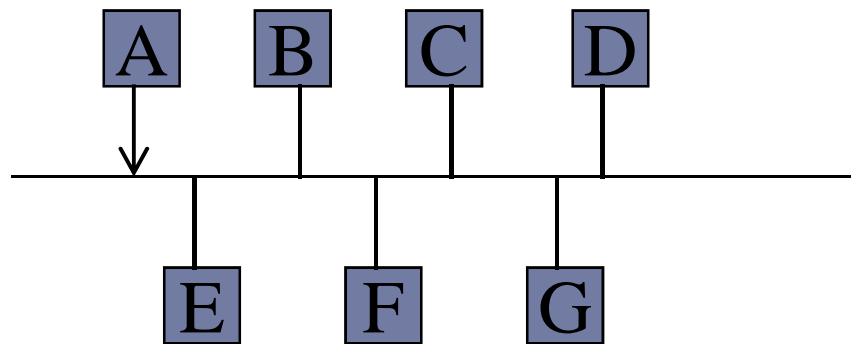
G2: A, D, F

G3: A, B, E, F, G

Multicast

A envia MSG para G1

Comunicação em Grupo



G1: A, C, E

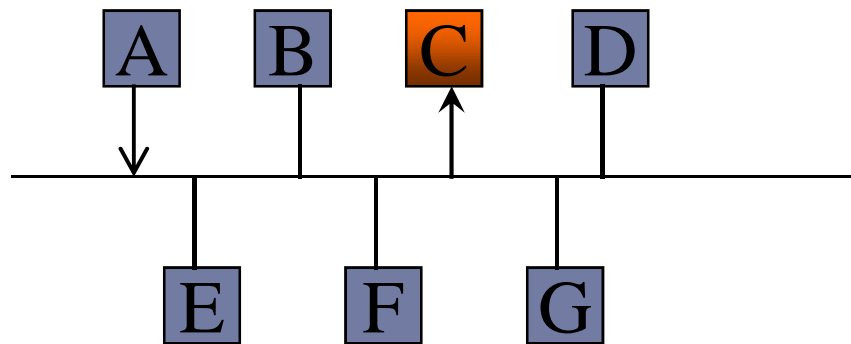
G2: A, D, F

G3: A, B, E, F, G

Unicast

A envia MSG para G1

Comunicação em Grupo



G1: A, C, E

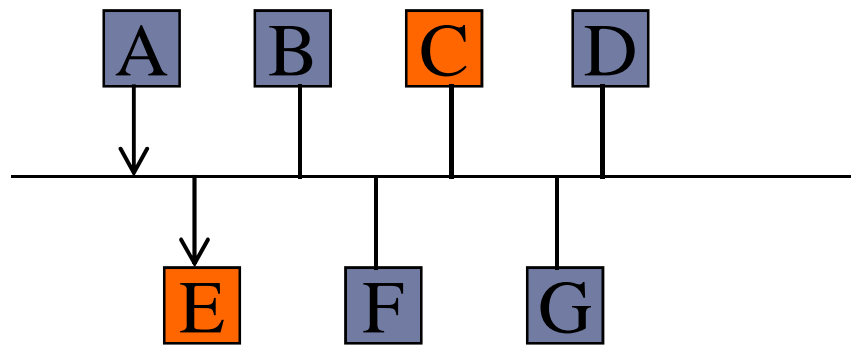
G2: A, D, F

G3: A, B, E, F, G

Unicast

A envia MSG para G1

Comunicação em Grupo



G1: A, C, E

G2: A, D, F

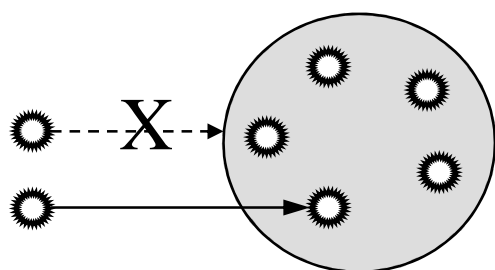
G3: A, B, E, F, G

Unicast

A envia MSG para G1

Comunicação em Grupo

- Organização de grupos:
 - Grupos Fechados X Grupos Abertos

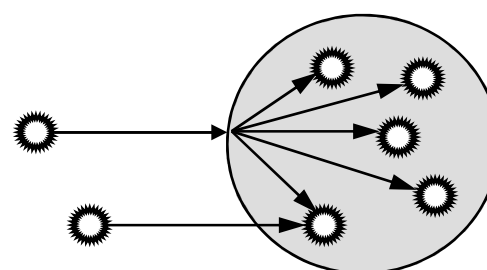


GRUPO FECHADO

Apenas membros do Grupo podem enviar mensagens ao Grupo.

Processo externo pode apenas enviar diretamente para membro do grupo.

**Usado para implementar processamento paralelo.
(Jogo de Xadrez)**



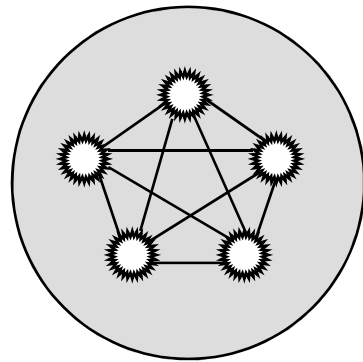
GRUPO ABERTO

Processos fora do grupo podem enviar mensagens para o Grupo ou para integrantes desse.

**Usado para implementar grupos de servidores.
(Replicação)**

Comunicação em Grupo

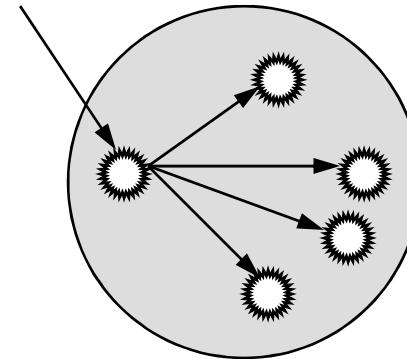
- ▶ Estrutura interna do grupo para tomada de decisões



GRUPO IGUALITÁRIO

Tomada de decisão por votação
50% dos votos + 1

coordenador



GRUPO HIERÁRQUICO

Processo Coordenador

Comunicação em Grupo

- ▶ **Grupo Iguatário**

- ▶ Simétrico

- ▶ Se um processo falhar, o grupo pode continuar seu trabalho.

- ▶ Todas as decisões são tomadas coletivamente, através de votação

- ▶ Complexidade e *overhead* (sobrecarga).

Comunicação em Grupo

▶ Grupo Hierárquico

- ▶ Uma falha de um coordenador provoca uma interrupção temporária do serviço, até que um novo coordenador seja eleito.
- ▶ É preciso haver um mecanismo de detecção de falha do coordenador.
- ▶ Todas as decisões são tomadas pelo coordenador.
- ▶ Jogo de Xadrez paralelo
 - ▶ Coordenador controla a estratégia
 - Trabalhadores avaliam jogadas

Comunicação em Grupo

- ▶ **Endereçamento de grupos**
 - ▶ Semelhante ao endereçamento de processos
 - ▶ Endereço único de grupo
 - ▶ Endereços de multicast
 - ▶ Endereços de broadcast
 - ▶ Unicast
 - O kernel transmissor deve manter uma lista das máquinas que possuem processos pertencentes ao grupo

Comunicação em Grupo

- ▶ **Endereçamento de grupos**
 - ▶ Um endereço para cada processo
 - ▶ O cliente mantém uma lista de endereços dos processos filiados ao grupo, que deve ser sempre atualizada

Comunicação em Grupo

- ▶ **Primitivas de comunicação Send e Receive**
 - ▶ Não é fácil encapsular as primitivas de comunicação em grupo em RPCs
 - ▶ Usar as mesmas primitivas do modelo cliente/servidor
 - ▶ O endereço de grupo indica que é uma comunicação em grupo
 - SEND(socket, msg, tamanho, flag);
 - **SEND(endereçoDeGrupo, msg, tamanho, flag);**
 - ▶ O Receive aceita tanto mensagens endereçadas ao processo, quanto mensagens endereçadas ao grupo.

Comunicação em Grupo

- ▶ **Primitivas de comunicação Send e Receive**
 - ▶ Usar novas primitivas
 - ▶ group-send
 - group-send(endereçoGrupo, msg, tamanho, flag)
 - ▶ group-recv
 - group-recv(endereçoGrupo, msg, tamanho, flag)
 - ▶ get-reply

Comunicação em Grupo

▶ Gerência de Grupos

▶ Servidor de grupo

- ▶ Banco de dados de grupos
- ▶ Todas as requisições são enviadas para o servidor de grupos
- ▶ Simples
- ▶ Pouco confiável
 - Ponto de falha (centralização)
 - Solução via Replicação

Comunicação em Grupo

▶ Gerência de Grupos

▶ Gerência distribuída

- ▶ Trivial em grupos abertos
- ▶ Grupos fechados
 - Permitir Requisição de Inscrição de processos fora do grupo
 - “Abertura”
- ▶ Tratamento de falha de membros
 - Saída do grupo?
 - Como detectar?
- ▶ Consistência das visões do Grupo
 - Sincronismo de membros com mensagens

Comunicação em Grupo

- ▶ **Atomicidade**
 - ▶ Propriedade do todos-ou-nenhum
 - ▶ Recebimento das mensagens por todos os membros ou nenhum
 - ▶ Facilita a programação de aplicações distribuídas
 - ▶ Implementação não é simples, especialmente em decorrência da possibilidade de faltas
 - ▶ Ex: buffer cheio em alguns destinatários

Comunicação em Grupo

- ▶ **Atomicidade**

- ▶ Exemplo: algoritmo de Joseph e Birman

- ▶ O transmissor envia a mensagem para todos os membros e inicia um timeout, para possíveis retransmissões.
 - ▶ Cada membro, ao receber uma mensagem pela primeira vez, envia a mensagem para todos os demais membros, iniciando um timeout para possíveis retransmissões.

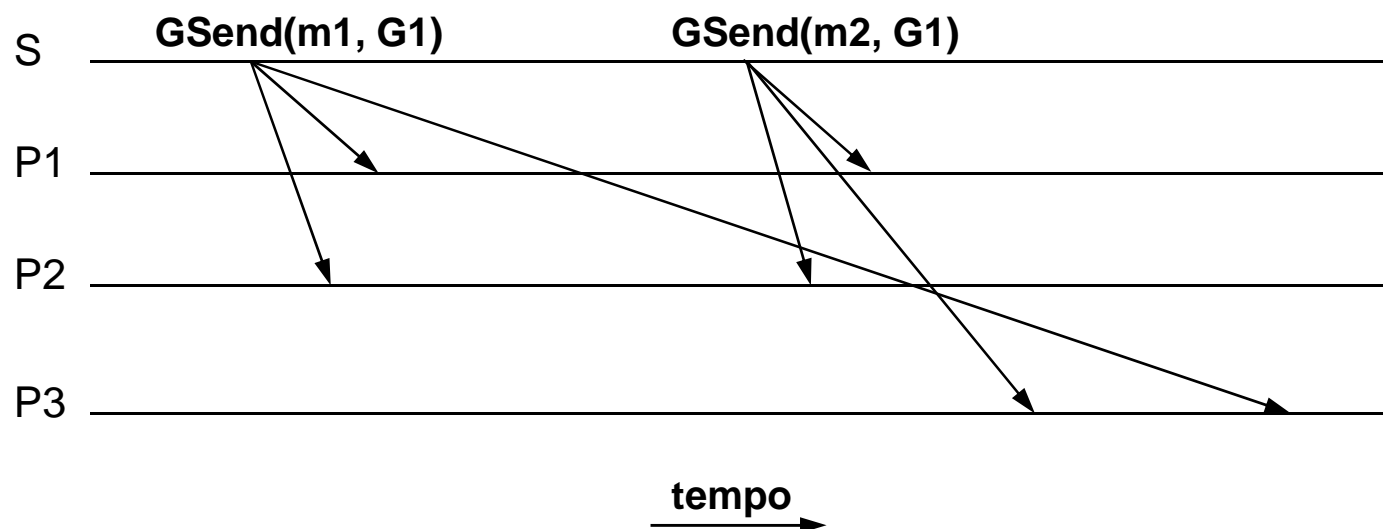
Comunicação em Grupo

- ▶ Ordenação de mensagens : **ORDENAÇÃO CAUSAL**
- ▶ Segundo Lamport, considerando-se os eventos de enviar e receber mensagens, pode-se definir a relação *happened-before*, expressa através da notação \rightarrow , como sendo:
 - ▶ Se a e b são eventos no mesmo processo e a acontece antes de b então $a \rightarrow b$
 - ▶ Se a corresponde ao evento de enviar mensagem por um processo e b ao evento de receber esta mensagem por outro processo, então $a \rightarrow b$
 - ▶ Se $a \rightarrow b$ e $b \rightarrow c$ então $a \rightarrow c$. Dois eventos distintos a e b são concorrentes se : $a \rightarrow b$ e $b \rightarrow a$

Comunicação em Grupo

- ▶ Ordenação de mensagens
 - ▶ Nenhuma ordenação e não confiável

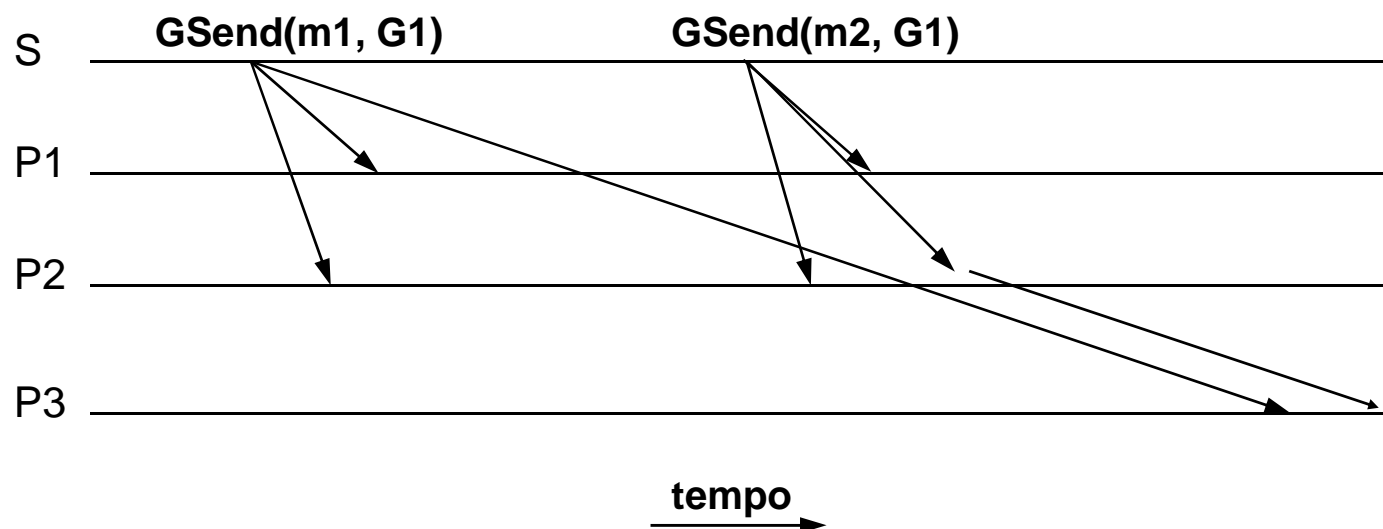
$G1 = \{P1, P2, P3\}$



Comunicação em Grupo

- ▶ Ordenação de mensagens
 - ▶ Uso de FIFO com base em confirmação

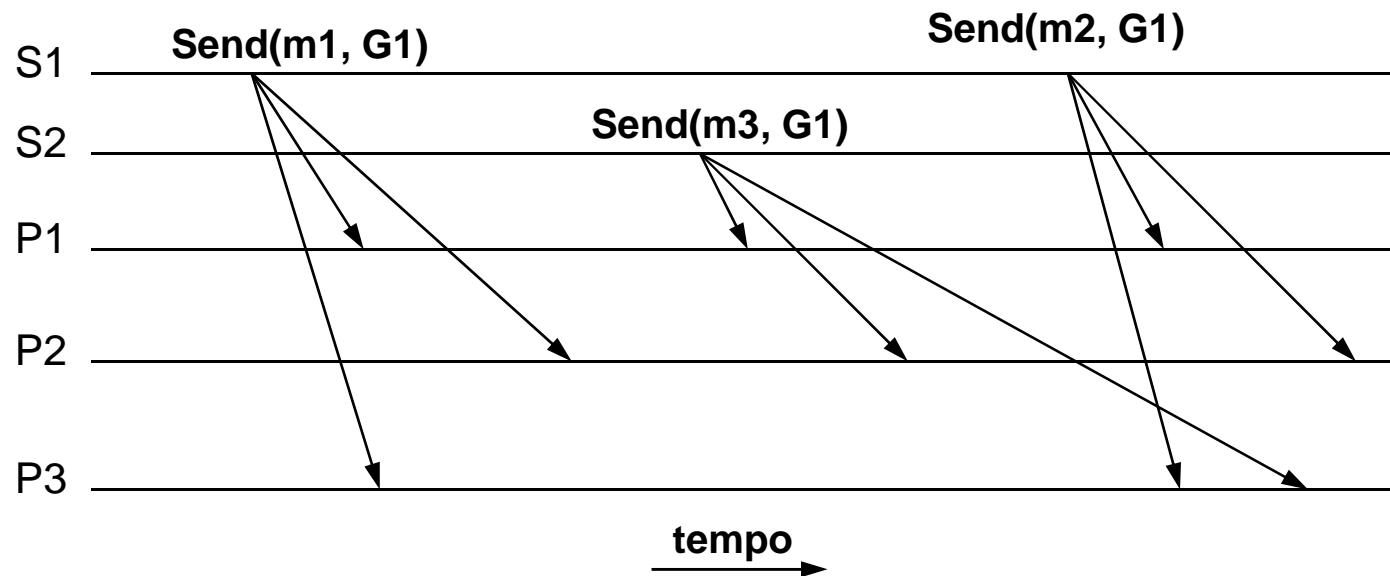
$G1 = \{P1, P2, P3\}$



Comunicação em Grupo

- Ordenação de mensagens
 - FIFO : as mensagens de cada cliente são entregues na ordem em que foram enviadas

$G1 = \{P1, P2, P3\}$

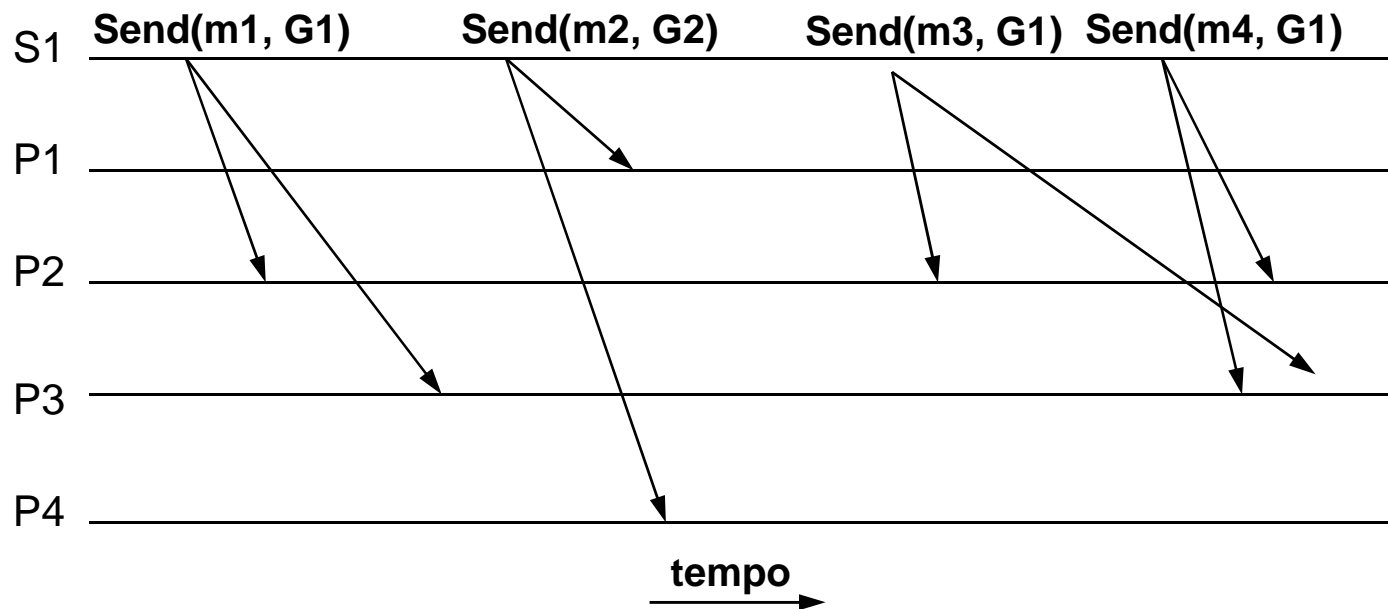


Comunicação em Grupo

- ▶ Ordenação de mensagens : ORDENAÇÃO CAUSAL

$G1 = \{P2, P3\}$

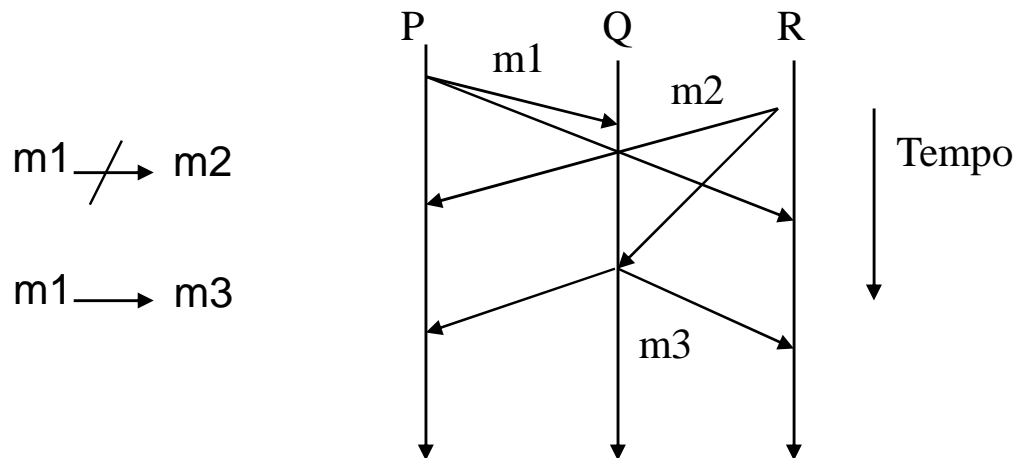
$G2 = \{P1, P4\}$



Comunicação em Grupo

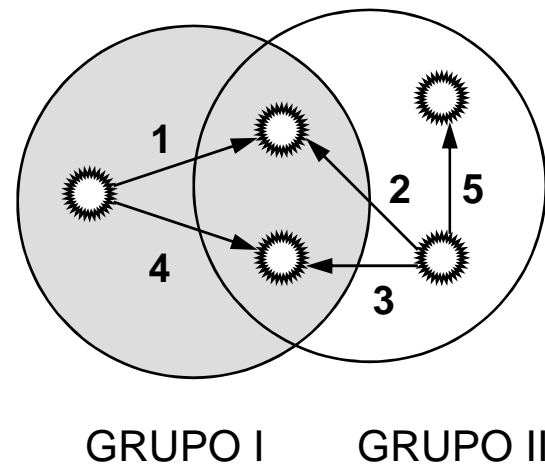
○ Ordenação de mensagens : ORDENAÇÃO TOTAL

- A ordenação total garante que todas as mensagens enviadas para um grupo serão entregues na mesma ordem, a todos os componentes do grupo, mesmo as mensagens concorrentes.



Comunicação em Grupo

- ▶ Sobreposição de grupos
 - ▶ Poucos sistemas tratam esse problema



Comunicação em Grupo

▶ Escalabilidade

- ▶ Muitos protocolos só funcionam bem quando os grupos são pequenos
- ▶ Muitos protocolos só funcionam em uma única LAN
 - ▶ Problemas em gateways
- ▶ Outros problemas
 - ▶ Complexidade computacional
 - ▶ Componentes centralizados

Comunicação em Grupo no ISIS

○ Primitivas de comunicação

- Todas as três primitivas são atômicas
- ABCAST
 - Ordenação consistente no tempo
 - Implementação complexa e cara
- CBCAST
 - Ordenação causal
 - Suporta superposição de grupos
- GBCAST
 - Ordenação consistente no tempo
 - Usado para gerenciar o membership do grupo

Comunicação em Grupo no ISIS

► Protocolo CBCAST

