

Computação Distribuída

Odorico Machado Mendizabal



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



Comunicação em Grupo

Comunicação em grupo

- Coleção de processos que atuam como parte de um grupo
- Comunicação ponto à ponto (1:1) ou multiponto (1:n)
 - *Unicast, Multicast, Broadcast*
- Gerenciamento do grupo (entrada e saída de membros)
 - *Group membership*
- Outras garantias podem ser atribuídas ao grupo de processos comunicantes
 - Ex. segurança e confiabilidade

Uso de grupos de processos em Computação Distribuída

- Vídeoconferência
- Difusão de conteúdo
 - Áudio e vídeo
- Memória compartilhada distribuída
- Replicação de serviços
 - Ex. replicação ativa
- Bancos de dados distribuídos e sistemas transacionais
- Computação em nuvem

Propriedades da comunicação em grupo

- Tipos de comunicação (*unicast*, *multicast* e *broadcast*)
- Tipos de Grupos (abertos e fechados)
- Hierarquia de Grupos (hierárquicos e não hierárquicos)
- Dinâmica de Grupos (criação, destruição de grupos)
- Confiabilidade e Ordenação

Tipos de comunicação

- *Unicast*: mensagens são enviadas ponto a ponto entre dois processos (um para um)
- *Multicast (Difusão seletiva)*: mensagens são enviadas a processos que fazem parte de um determinado grupo (um para vários)
- *Broadcast (Difusão total)*: mensagens são enviadas a todos os processos que fazem parte de um determinado grupo (um para todos)

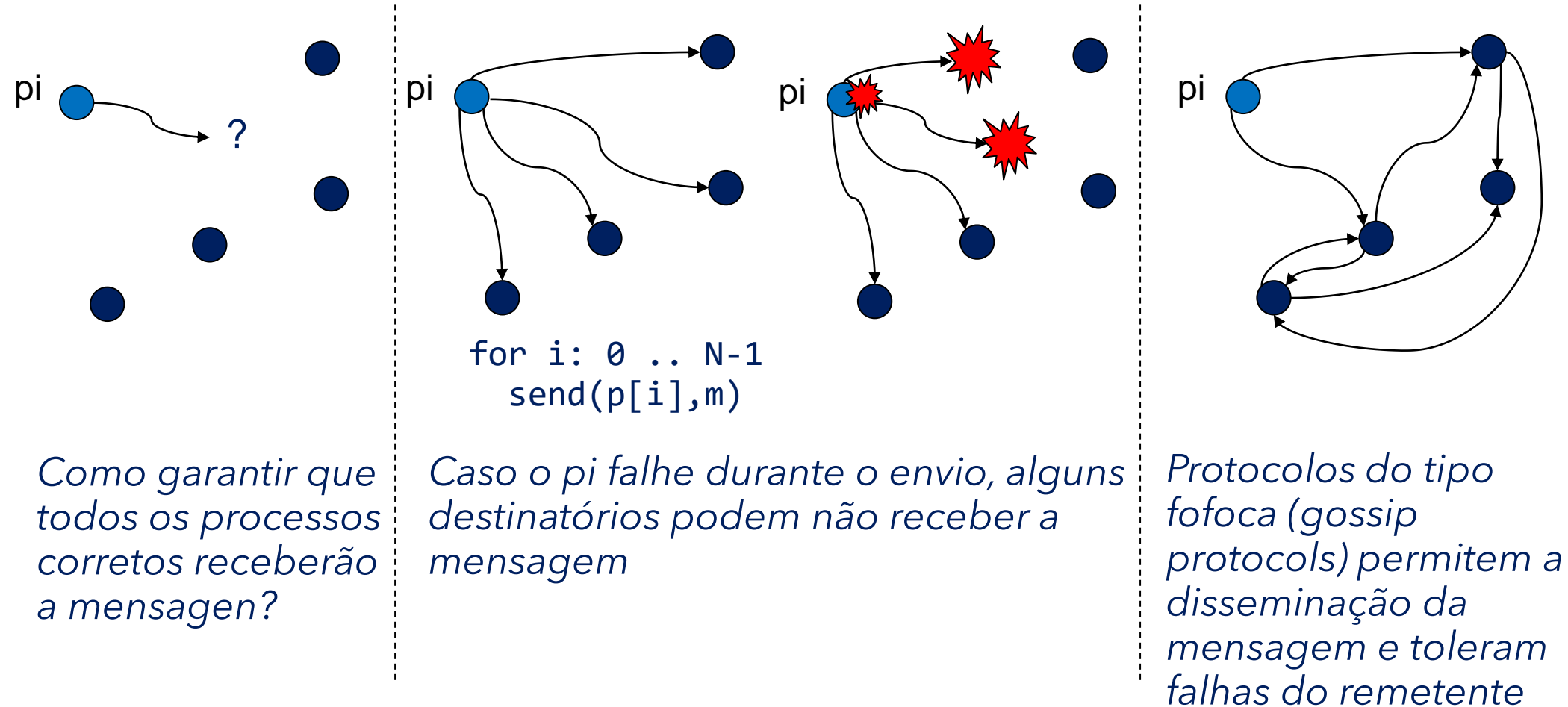
Diferentes propriedades podem ser oferecidas na troca de mensagens

Confiabilidade: garantias de entrega

Ordem: tipo de ordenação de mensagens

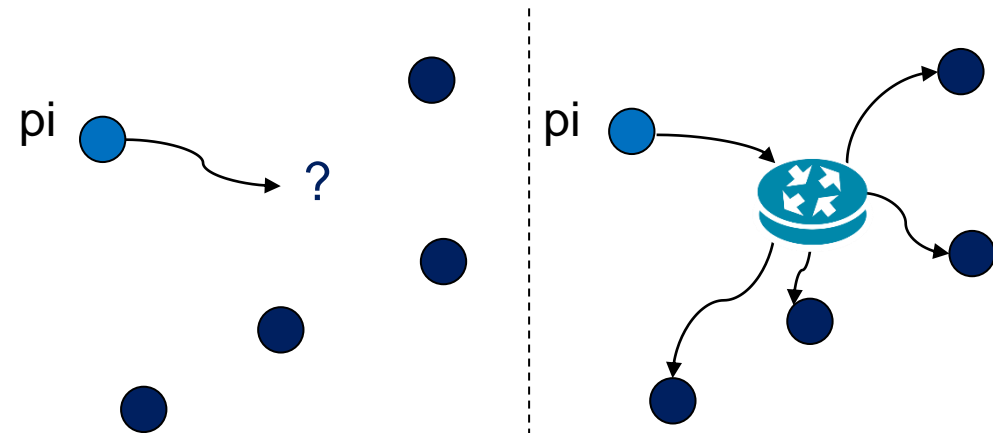
Tipos de comunicação – Implementação de difusão

pi quer enviar uma mensagem aos demais processos distribuídos



Tipos de comunicação – Implementação de difusão

pi quer enviar uma mensagem aos demais processos distribuídos



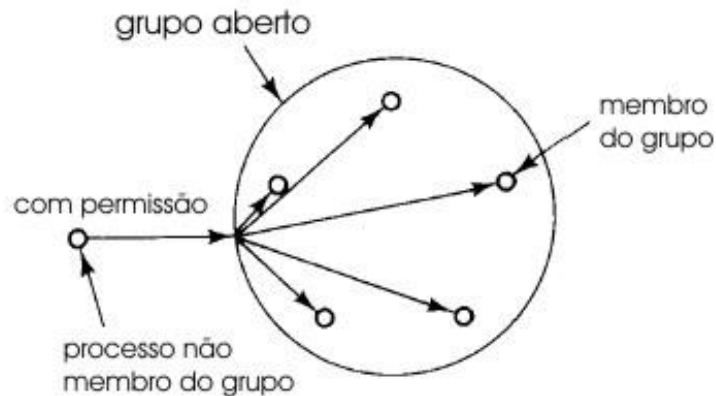
Como garantir que todos os processos corretos receberão a mensagem?

Suporte de rede

- *Uso de IP multicast (RFC 1112, 4604)*
- *Endereço de rede especial destina mensagem a todos os nodos da rede*
- *Nem sempre esta configuração é implementada pelos roteadores*

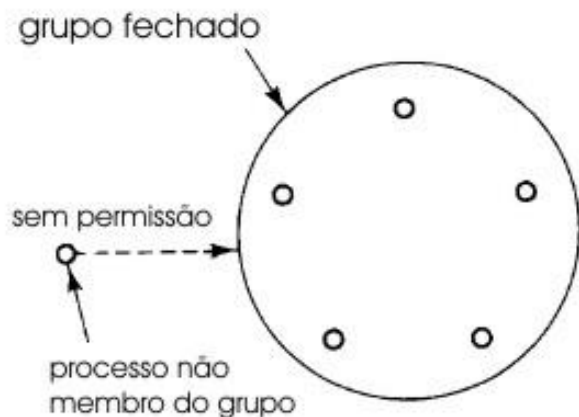
Tipos de grupos

- **Grupos Abertos:** processos externos ao grupo podem enviar mensagens aos grupo



Ex: servidores replicados permitindo que clientes externos ao grupo comuniquem com os servidores

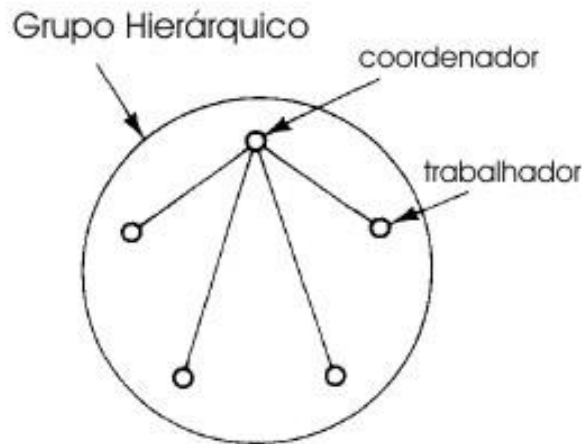
- **Grupos Fechados:** somente processos membros do grupo podem se comunicar internamente



Coleção de processos colaborativos. Não interagem com processos de fora do grupo.
Ex. uma sala de bate papo

Hierarquia de grupos

• Grupos Hierárquicos:

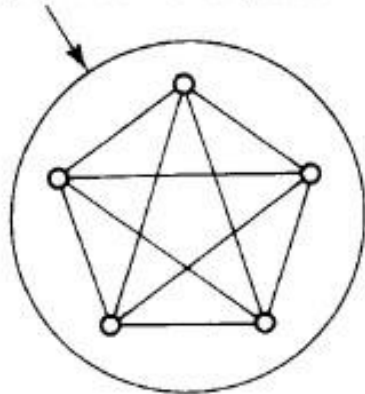


- um membro é o coordenador (mestre) do grupo
- membros restantes são os trabalhadores (escravos)
- controle das mensagens fica com o coordenador

vant. – rápidas decisões tomadas pelo coordenador
desv. – falha no coordenador exige eleição de novo líder

• Grupos Não Hierárquicos:

Grupo não hierárquico



- todos os membros são iguais
- não existe o conceito de coordenador
- as decisões são tomadas em conjunto (votação)

vant. – na falha em um processo, o sistema não para
desv. – maior complexidade e latência nas tomadas de decisões

Dinâmica de grupos

- Novos grupos podem ser criados
- Grupos podem ser destruídos
- Processos podem juntar-se ou deixar um grupo
 - Estaticamente:
 - Com base em configuração de rede ou de middleware para comunicação em grupo
 - Dinamicamente: em tempo de execução
 - Usando uma API de comunicação

- **Confiabilidade:**

- garantir que uma mensagem enviada a um grupo seja entregue a todos os membros, caso contrário ela é desconsiderada
- propriedade para **todos** ou para **nenhum**

- **Difusão Confiável (*Reliable broadcast*)**

- *Validade*: Se um processo correto envia m , então todos os processos corretos entregam m em algum momento
- *Acordo*: Se um processo correto entrega m , então todos os processos corretos entregam m em algum momento
- *Integridade*: Para qualquer mensagem m , cada processo correto entrega m no máximo uma vez, e apenas se m foi enviado previamente por algum processo

Confiabilidade e ordenação

- **Ordenação:**

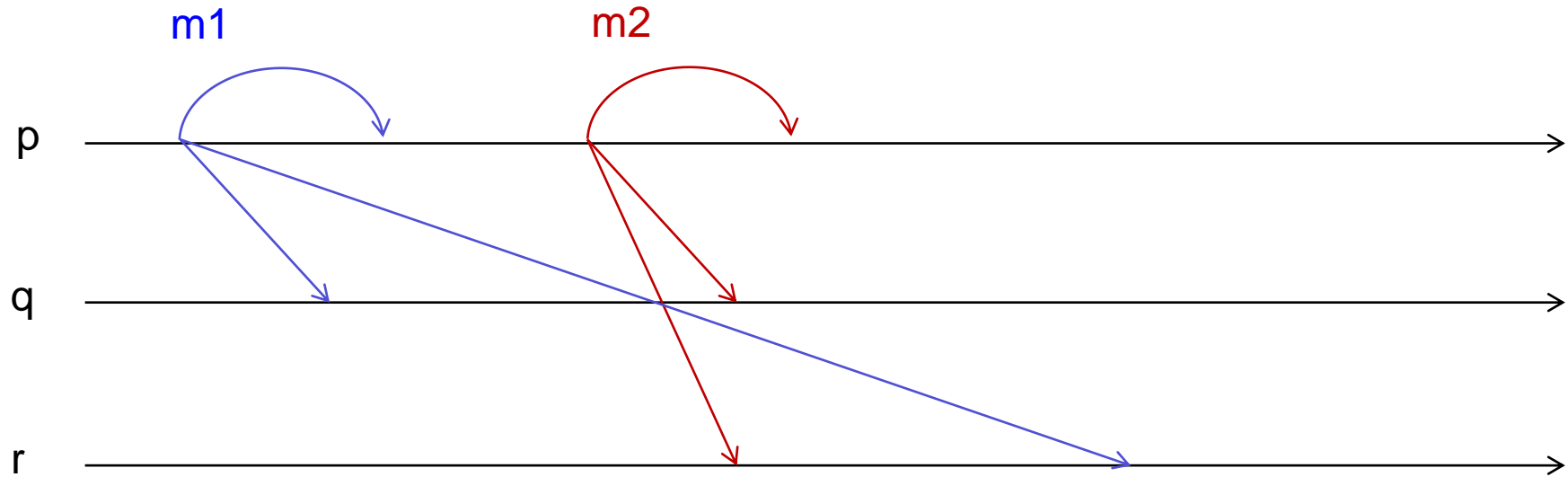
- Garantir que as mensagens sejam entregues aos processos respeitando algum critério

Ordem FIFO: Se um processo correto envia m antes de m' , então nenhum processo correto entrega m' a menos que m já tenha sido entregue

Ordem Causal: Se o envio de uma mensagem m precede causalmente o envio de uma mensagem m' , então nenhum processo correto entrega m' a menos que m já tenha sido entregue

Ordem Total: Se quaisquer processos corretos p e q entregam as mensagens m e m' , então p entrega m antes de m' , se e somente se q entrega m antes de m'

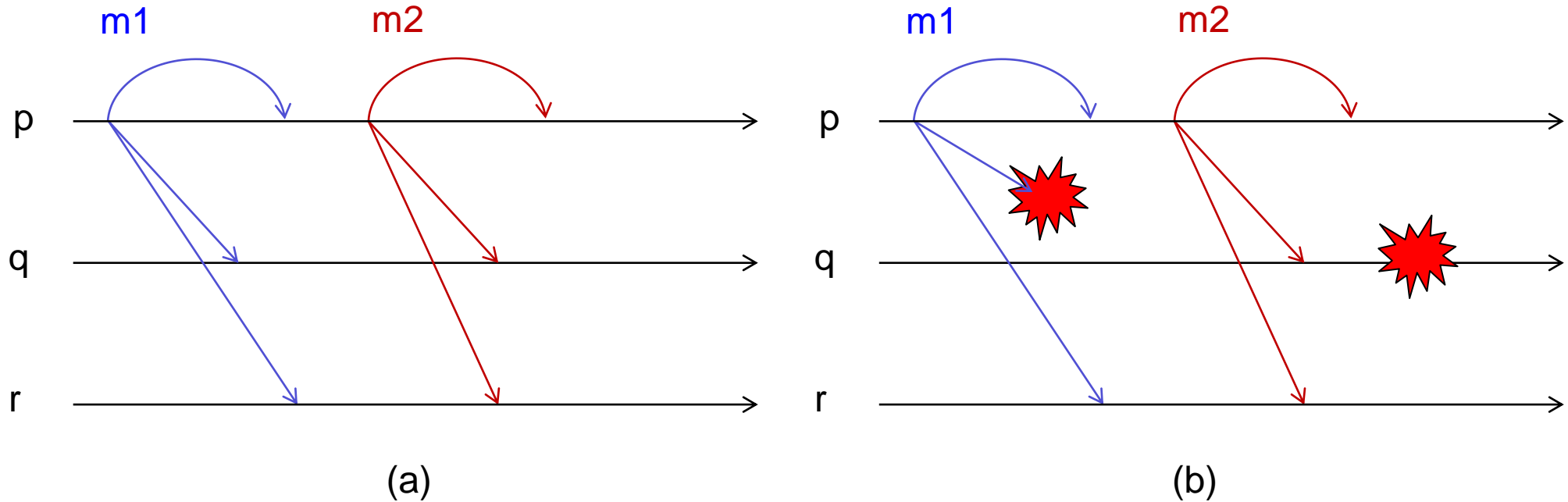
Confiabilidade e ordenação



Esta representação de difusão representa *difusão confiável*?

Sim, porém as ordens de entrega FIFO, Causal e Total são violadas

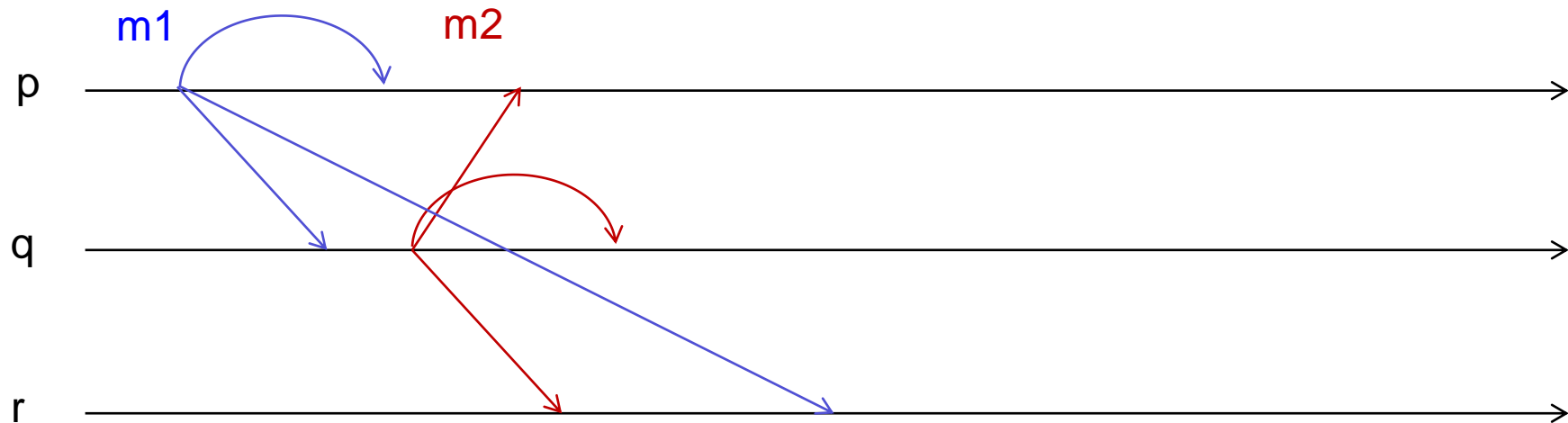
Confiabilidade e ordenação



As situações acima preservam ordem **FIFO**?

Sim, pois todos os processos corretos entregam as mensagens em uma mesma ordem

Confiabilidade e ordenação



Esta situação preserva as propriedades de entrega em ordem **FIFO**?

Sim

Imagine que as mensagens m1 e m2 sejam versões de um artigo revisado por autores (representados pelos processos p, q e r).

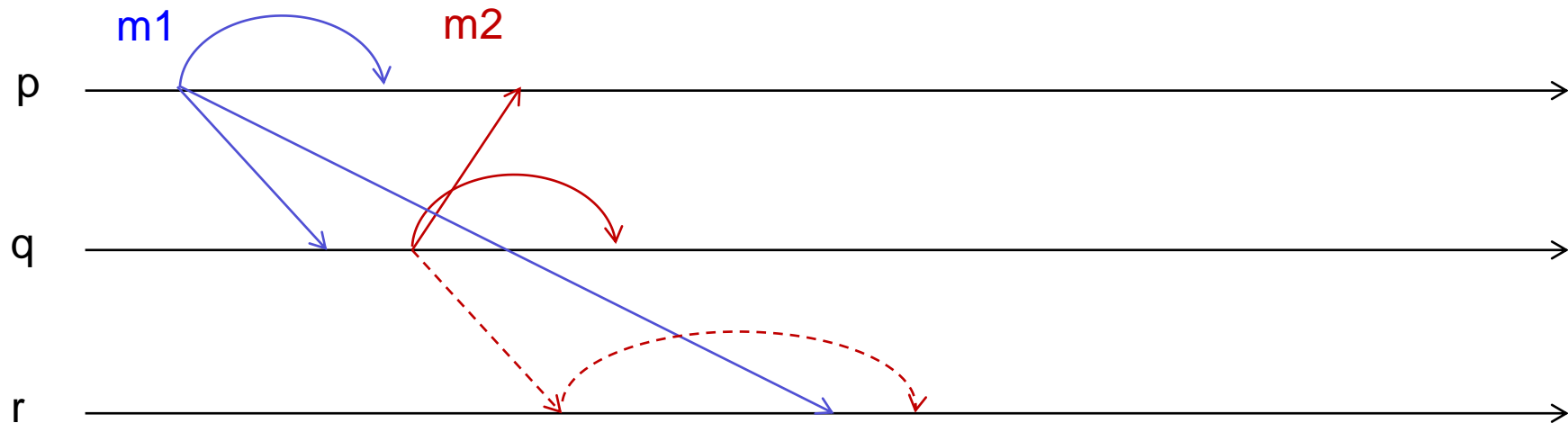
*A última versão recebida pelo processo r seria uma versão desatualizada. Modificações deste autor na última versão recebida não estariam contemplando as sugestões feitas por q. **Como resolver esta situação?***

Garantindo entrega de mensagens segundo a ordem causal

Confiabilidade e ordenação

- **Ordem Causal:** Se o envio de uma mensagem m precede causalmente o envio de uma mensagem m' , então nenhum processo correto entrega m' a menos que m já tenha sido entregue.
- Respeita a relação “**acontece antes**”, definida por Lamport (Time, Clocks, and the Ordering of Events in a Distributed System, 1978)
- Um evento a precede causalmente b , $a \rightarrow b$, se:
 - i) Um mesmo processo executa a e depois executa b (localmente)
 - ii) Evento a é o envio de uma mensagem m e b é o recebimento de m
 - iii) Causalidade é uma propriedade transitiva, se $a \rightarrow b$, $b \rightarrow c$, então $a \rightarrow c$

Confiabilidade e ordenação



Esta situação preserva as propriedades de entrega em ordem **Causal**?

Sim

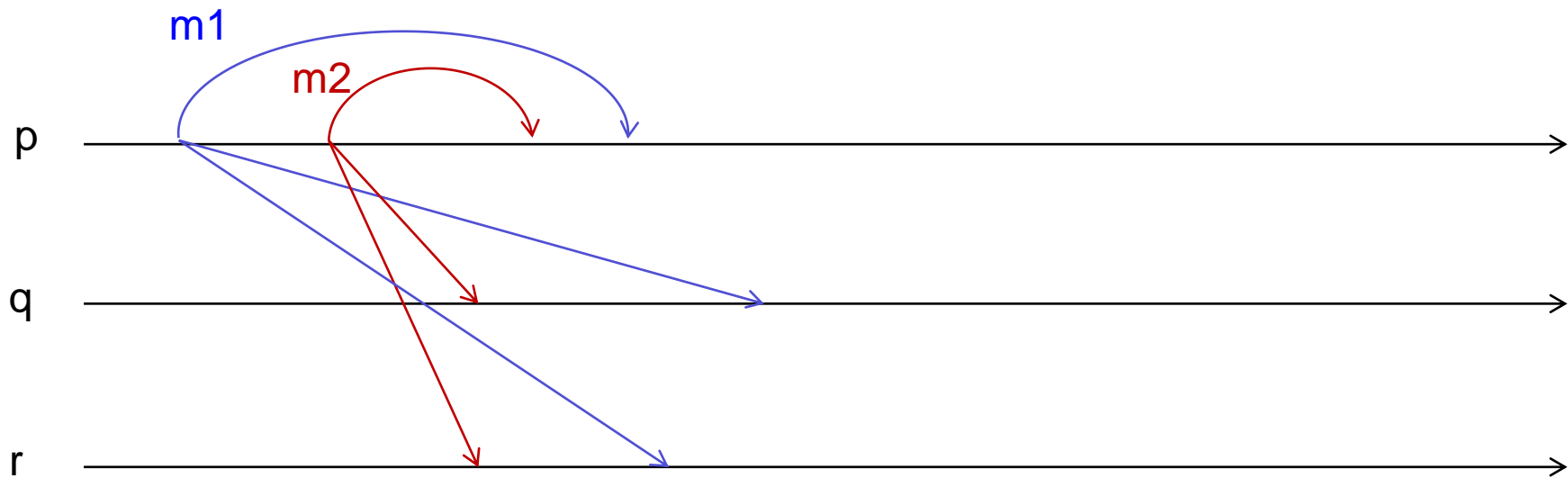
Note que:

Ordem causal implica em **Ordem FIFO**. Se duas mensagens m e m' são enviadas pelo mesmo processo p , e m foi enviada antes de m' , então m **acontece antes** de m'

Multicast que implementa ordem causal respeitará FIFO, mas o contrário não é verdade

Confiabilidade e ordenação

- **Ordem Total:** Se quaisquer processos corretos p e q entregam as mensagens m e m' , então p entrega m antes de m' , se e somente se q entrega m antes de m'



Esta situação preserva as propriedades da **Difusão Atômica**?

Sim

Esta situação preserva as propriedades de ordem **FIFO**?

Não

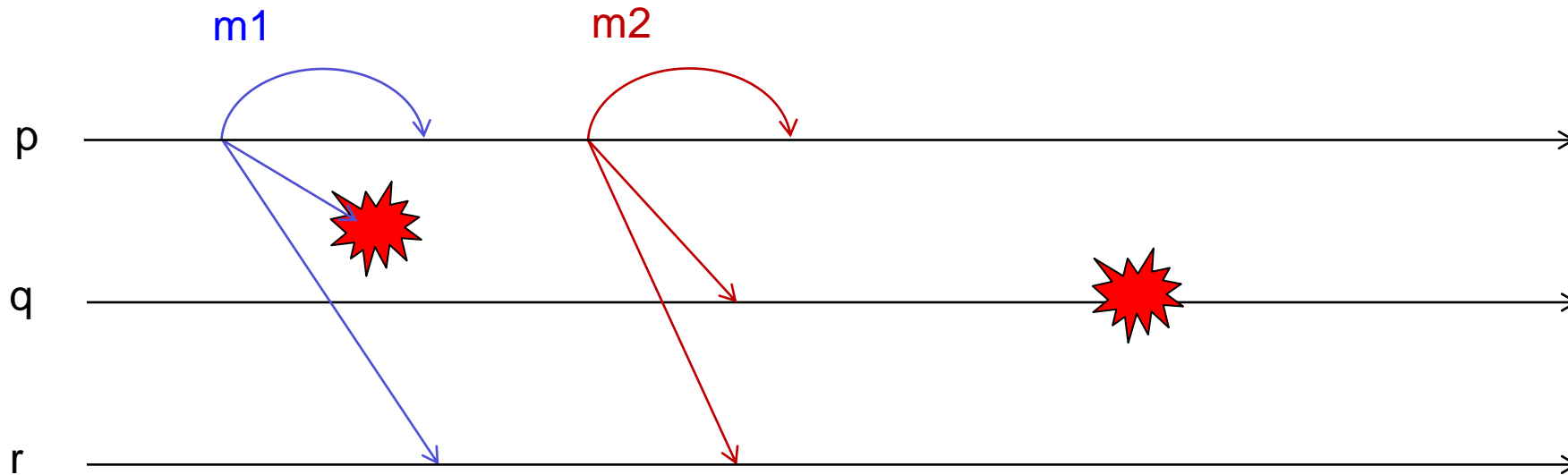
Difusão Confiável Uniforme (*Uniform Reliable Broadcast*)

- *Validade*: Se um processo correto envia m , então todos os processos corretos entregam m em algum momento
- *Acordo Uniforme*: Se um processo ~~correto~~ entrega m , então todos os processos corretos entregam m em algum momento
- *Integridade Uniforme*: Para qualquer mensagem m , cada processo ~~correto~~ entrega m no máximo uma vez, e apenas se m foi enviado previamente por algum processo

Difusão Confiável Uniforme (*Uniform Reliable Broadcast*)

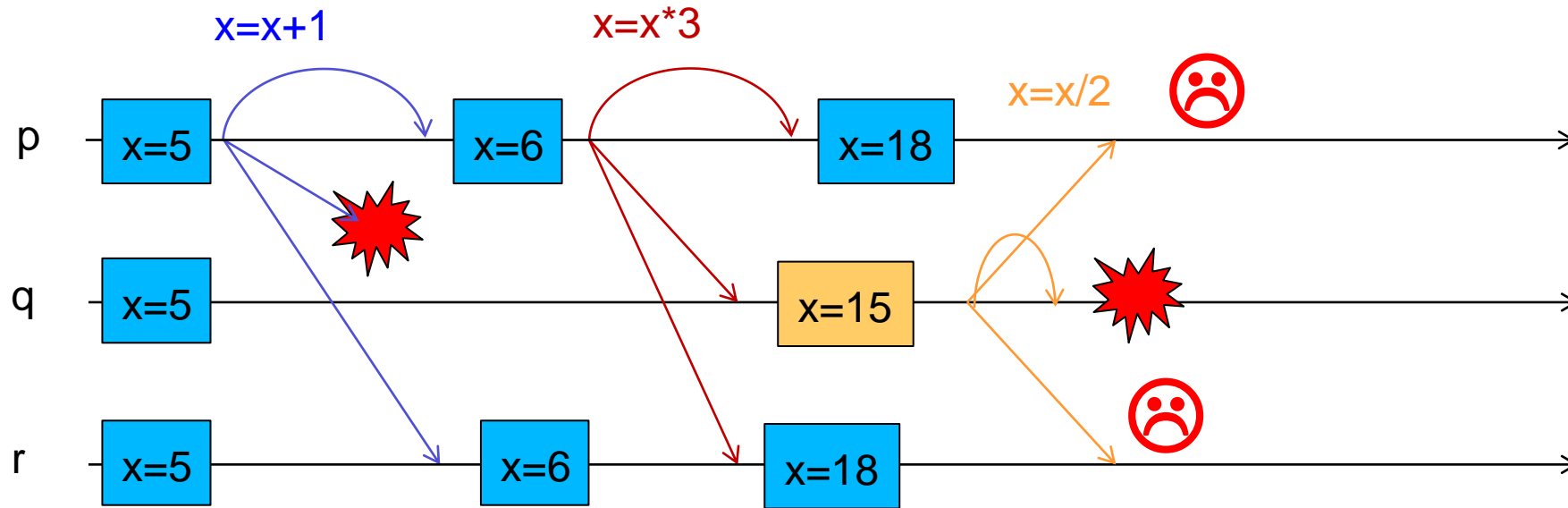
- **Difusão FIFO Uniforme:** Se um processo envia m antes de m' , então nenhum processo entrega m' a menos que m já tenha sido entregue
- **Difusão Causal Uniforme:** Se o envio de uma mensagem m precede causalmente o envio de uma mensagem m' , então nenhum processo entrega m' a menos que m já tenha sido entregue

Difusão Confiável Uniforme (*Uniform Reliable Broadcast*)



Este caso preserva as propriedades de ordem **FIFO**, mas não preserva as propriedades de ordem **FIFO Uniforme**

Difusão Confiável Uniforme (*Uniform Reliable Broadcast*)



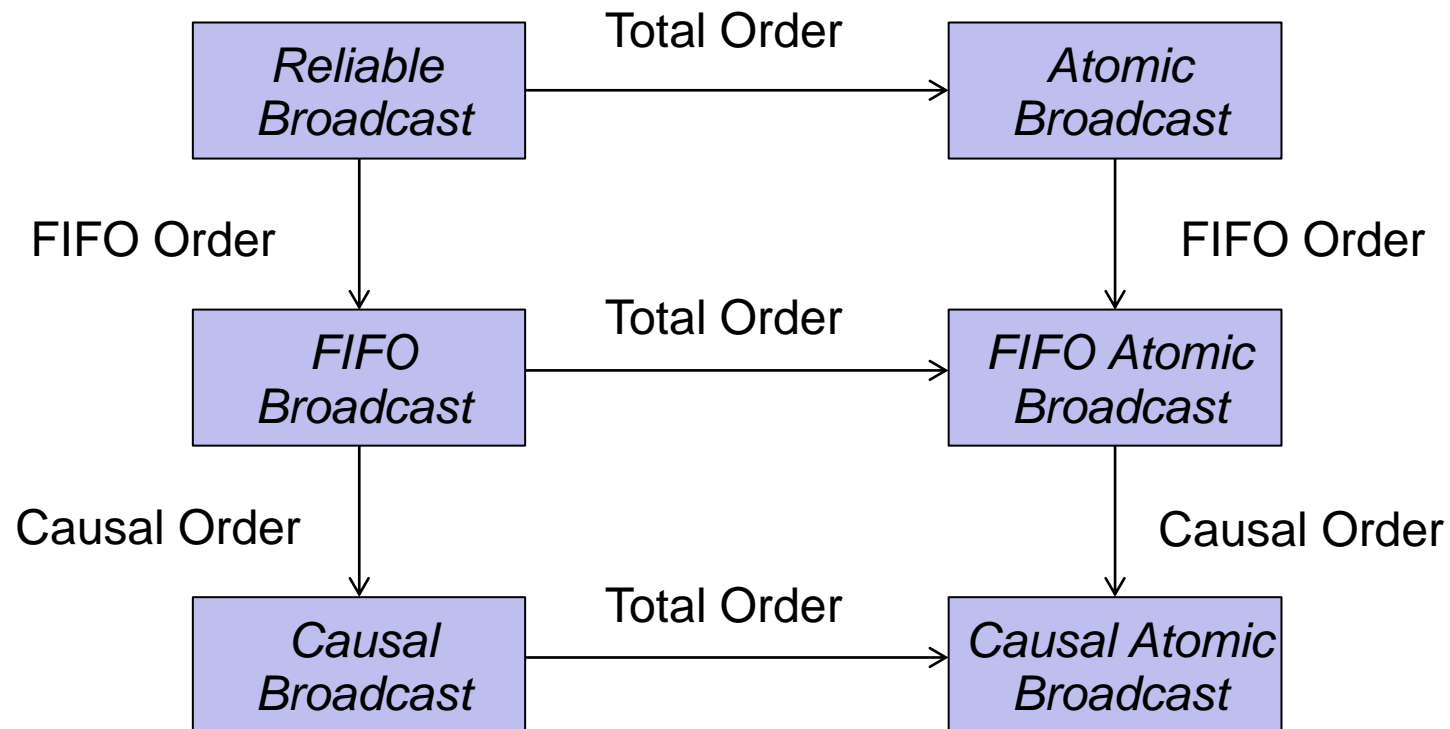
Este caso preserva as propriedades de ordem **FIFO**, mas não preserva as propriedades de ordem **FIFO Uniforme**

Ordem de entrega apenas FIFO não garante que processos corretos não sejam contaminados

Neste exemplo o processo q envia uma mensagem decorrente de um estado inconsistente em que o processo se encontrava

Primitivas de Comunicação Confiável

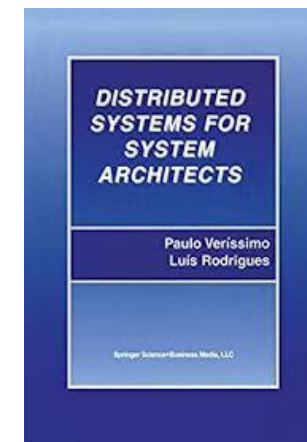
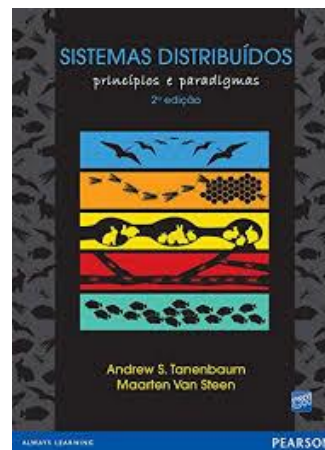
Relação entre as primitivas apresentadas



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.
- Veríssimo, P.; Rodrigues, L. *Distributed Systems for System Architects*. Springer; 1ª edição. 2001.



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