

Arquiteturas Paralelas e Distribuídas

# Programação distribuída: passagem de mensagens

Eduardo Furlan Miranda

Baseado em: MAZIERO, C. A. Sistemas Operacionais:  
Conceitos e Mecanismos. 2019.

# Comunicação entre tarefas

- Tarefas cooperantes precisam trocar informações entre si
- Se as tarefas estão no mesmo processo, elas compartilham a mesma área de memória e a comunicação pode então ser implementada facilmente, usando variáveis globais comuns
- Caso as tarefas pertençam a processos distintos, não existem variáveis compartilhadas
  - A comunicação tem de ser feita, p.ex., usando bibliotecas
- Tarefas em computadores diferentes
  - Se comunicam usando a rede
    - Velocidade de transmissão varia conforme a tecnologia de rede

# IPC (Inter-Process Communication)

3/16

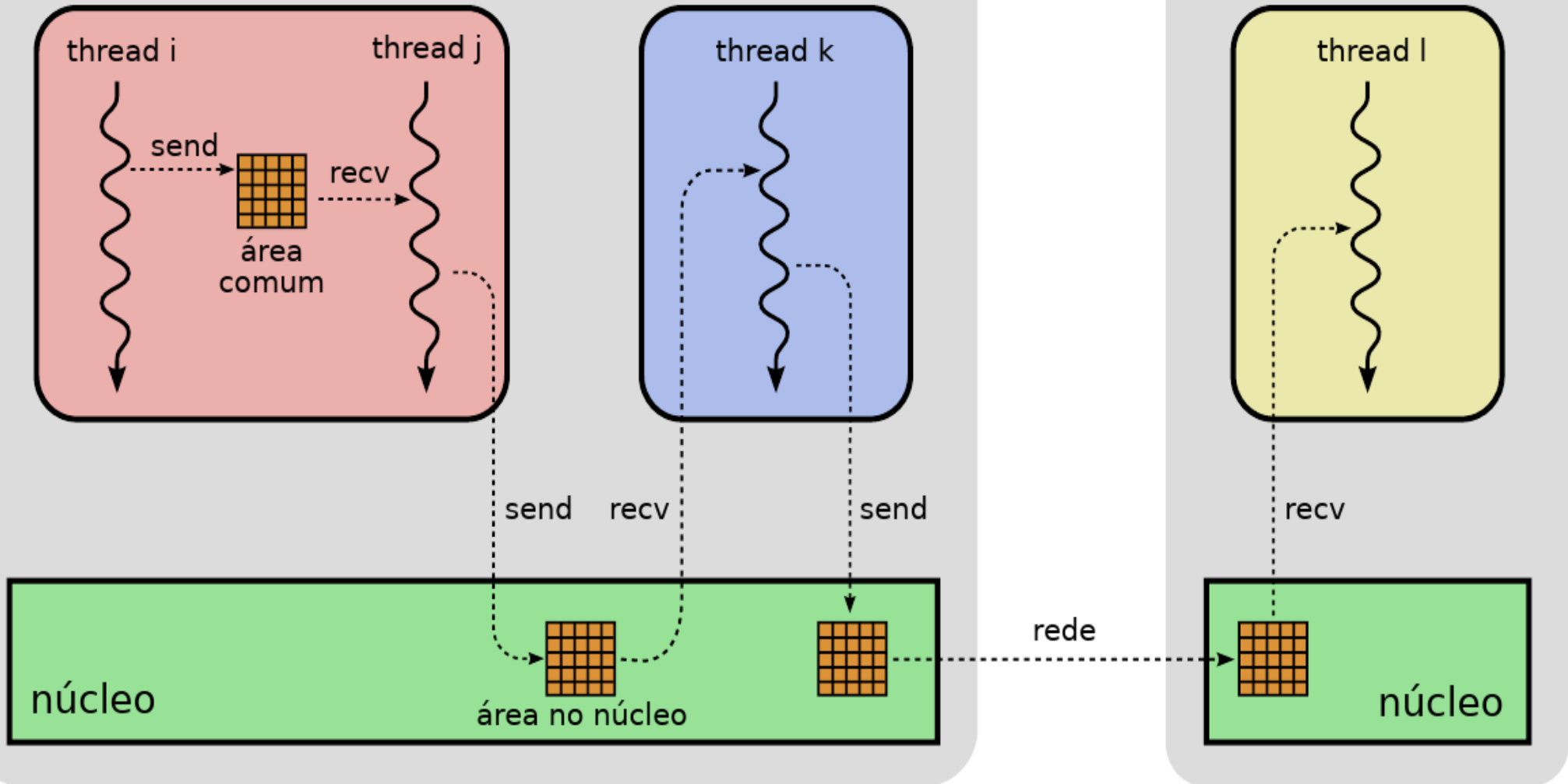
Computador 1

Computador 2

Processo  $p_a$

Processo  $p_b$

Processo  $p_c$



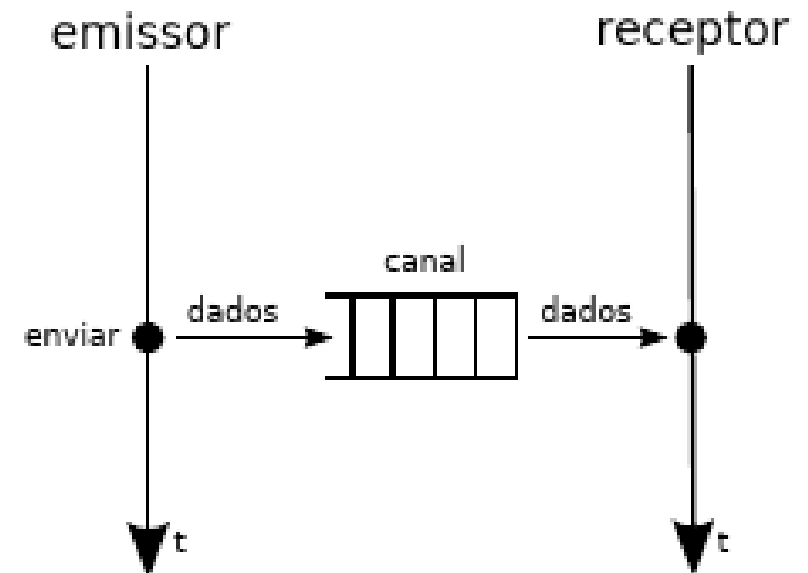
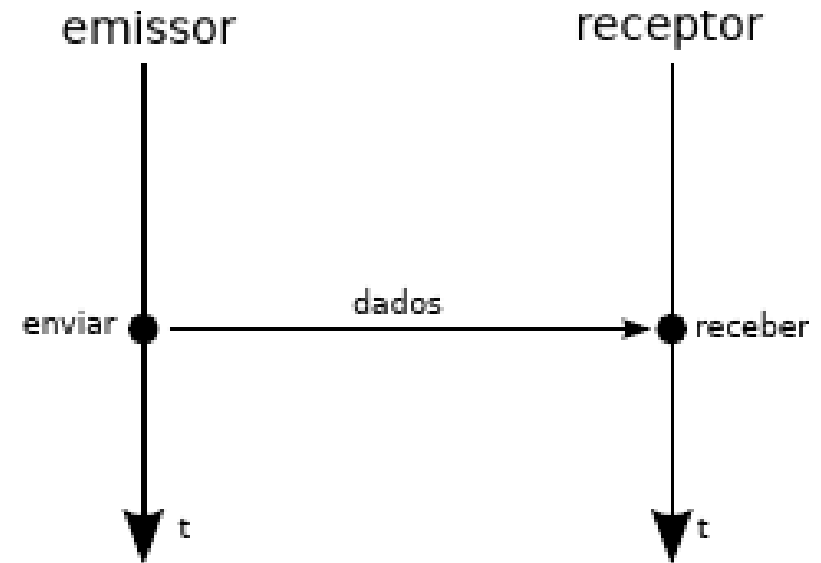
Comunicação intraprocesso ( $t_i \rightarrow t_j$ ), interprocessos ( $t_j \rightarrow t_k$ ) e intersistemas ( $t_k \rightarrow t_l$ )

# Aspectos da comunicação

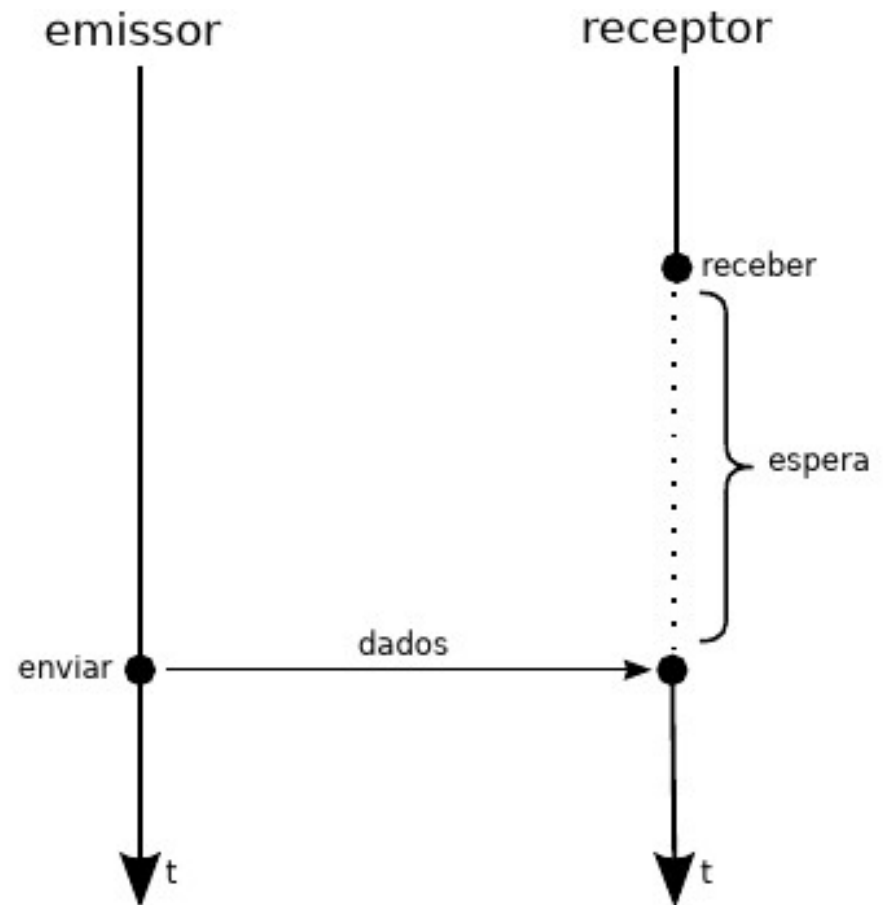
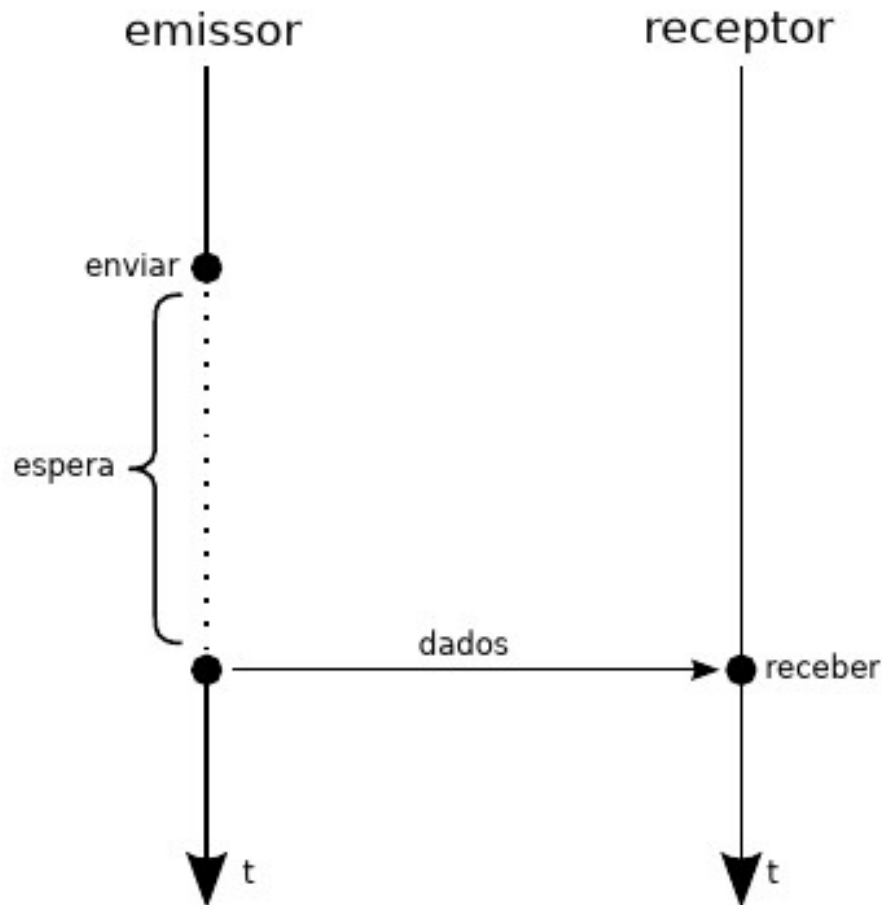
- Formato dos dados a transferir
- Sincronismo exigido nas comunicações
- Necessidade de buffers
- Nº de emissores/receptores envolvidos em cada ação de comunicação
- Velocidade
- etc.

# Comunicação direta ou indireta

- Direta
  - enviar (dados, destino)
  - receber (dados, origem)
- Indireta
  - emissor e receptor não precisam se conhecer
  - se relacionam através de um canal de comunicação
  - enviar (dados, canal)
  - receber (dados, canal)

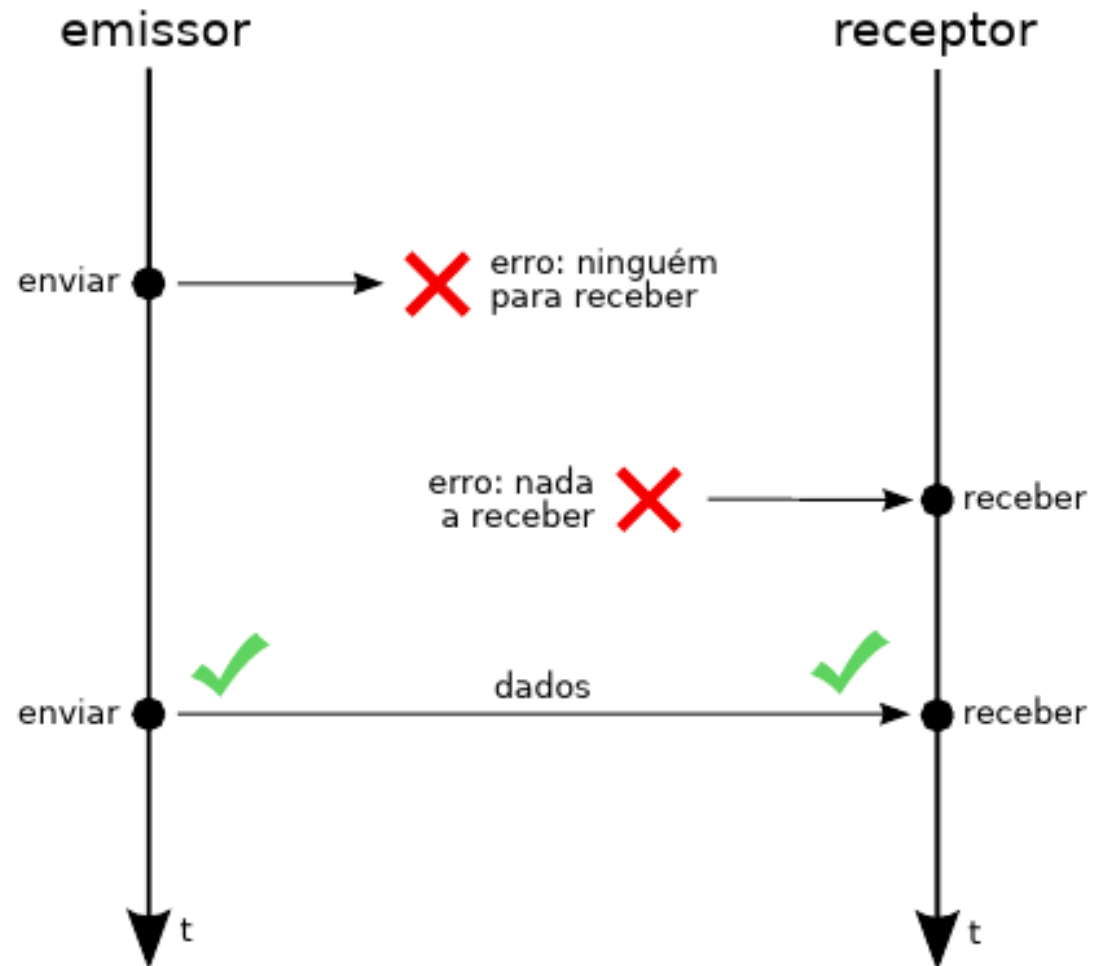


- Síncrona (ou bloqueante)
  - operações de envio e recepção de dados bloqueiam (suspendem) as tarefas envolvidas até a conclusão da comunicação



# Assíncrona (ou não-bloqueante)

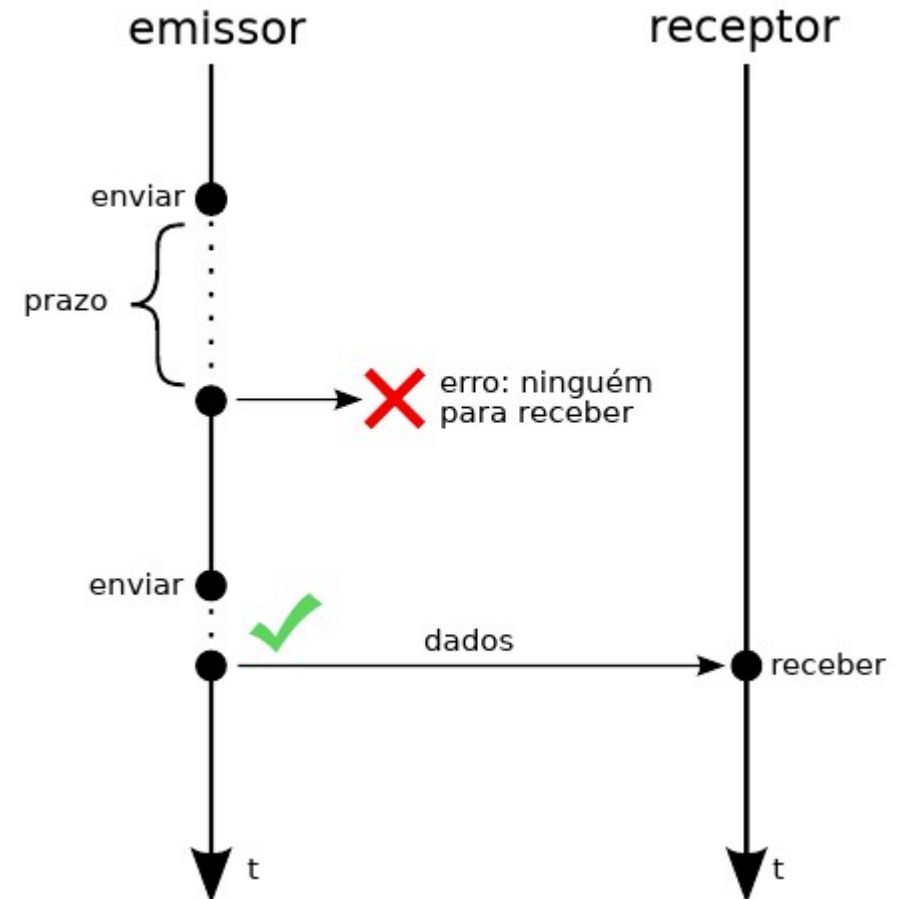
- caso o emissor e o receptor operem ambos de forma assíncrona, torna-se necessário criar um canal ou buffer para armazenar os dados da comunicação entre eles



# Semissíncrona (ou semibloqueante)

8/16

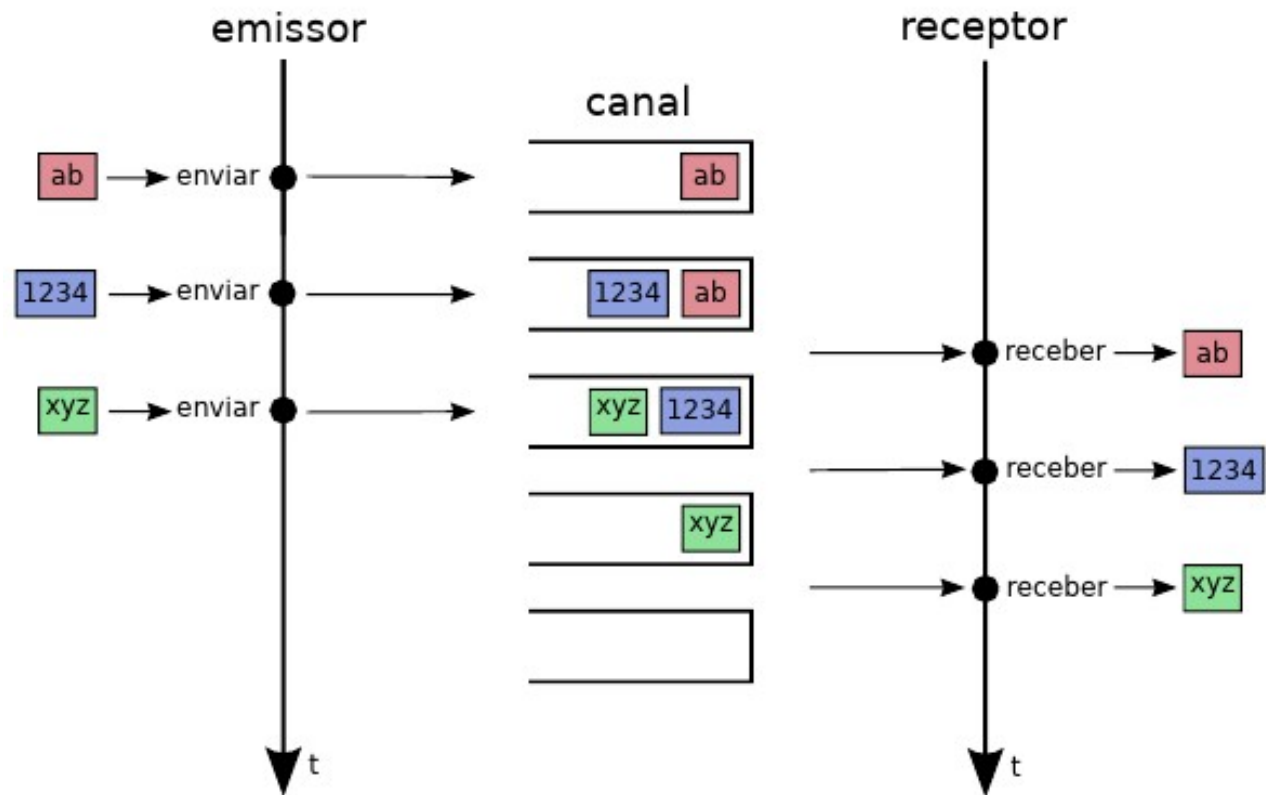
- Comportamento síncrono (bloqueante) durante um prazo predefinido
- Caso esse prazo se esgote sem que a comunicação tenha ocorrido, a primitiva se encerra com uma indicação de erro





# Formato de envio

- A informação enviada pelo emissor ao receptor pode ser vista basicamente de duas formas
  - como uma sequência de mensagens independentes, cada uma com seu próprio conteúdo, ou
  - como um fluxo sequencial e contínuo de dados, imitando o comportamento de um arquivo com acesso sequencial



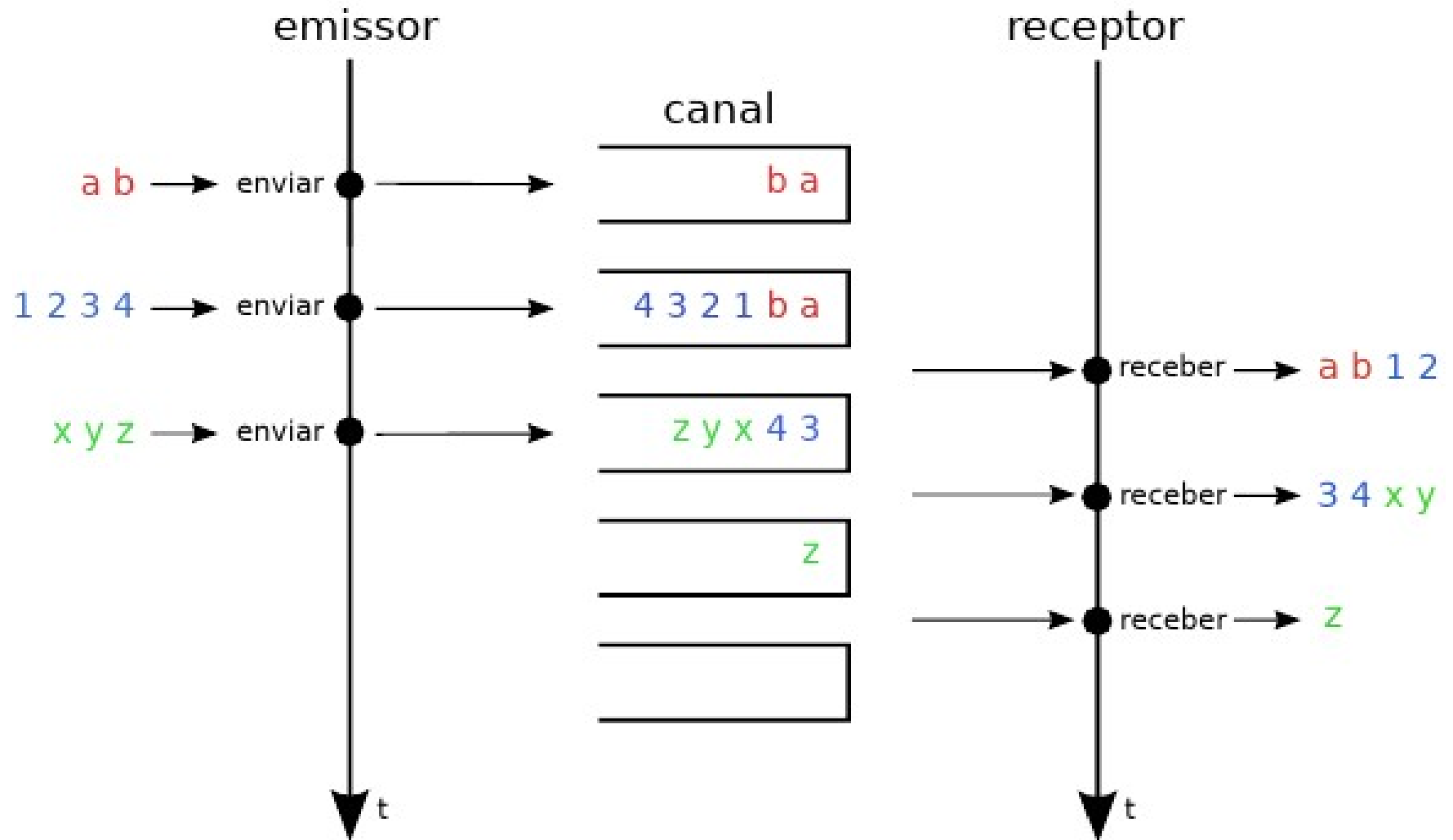
# Abordagem baseada em mensagens

- Cada mensagem consiste de um pacote de dados que pode ser tipado ou não
- O pacote é recebido ou descartado pelo receptor em sua íntegra
  - não existe a possibilidade de receber “meia mensagem”
- Ex.: message queues do UNIX e os protocolos de rede IP e UDP

# Comunicação como fluxo contínuo de dados

- O canal de comunicação é visto como o equivalente a um arquivo
  - o emissor “escreve” dados nesse canal, que serão “lidos” pelo receptor respeitando a ordem de envio dos dados
- Não há separação lógica entre os dados enviados em operações separadas
  - o receptor não vê os dados como conjuntos separados ou "pacotes" distintos; para ele, é como se fosse um único fluxo contínuo
  - eles podem ser lidos byte a byte ou em grandes blocos a cada operação de recepção, a critério do receptor
- Ex.: pipes do UNIX e o protocolo de rede TCP/IP

# Comunicação baseada em fluxo de dados <sup>12/16</sup>

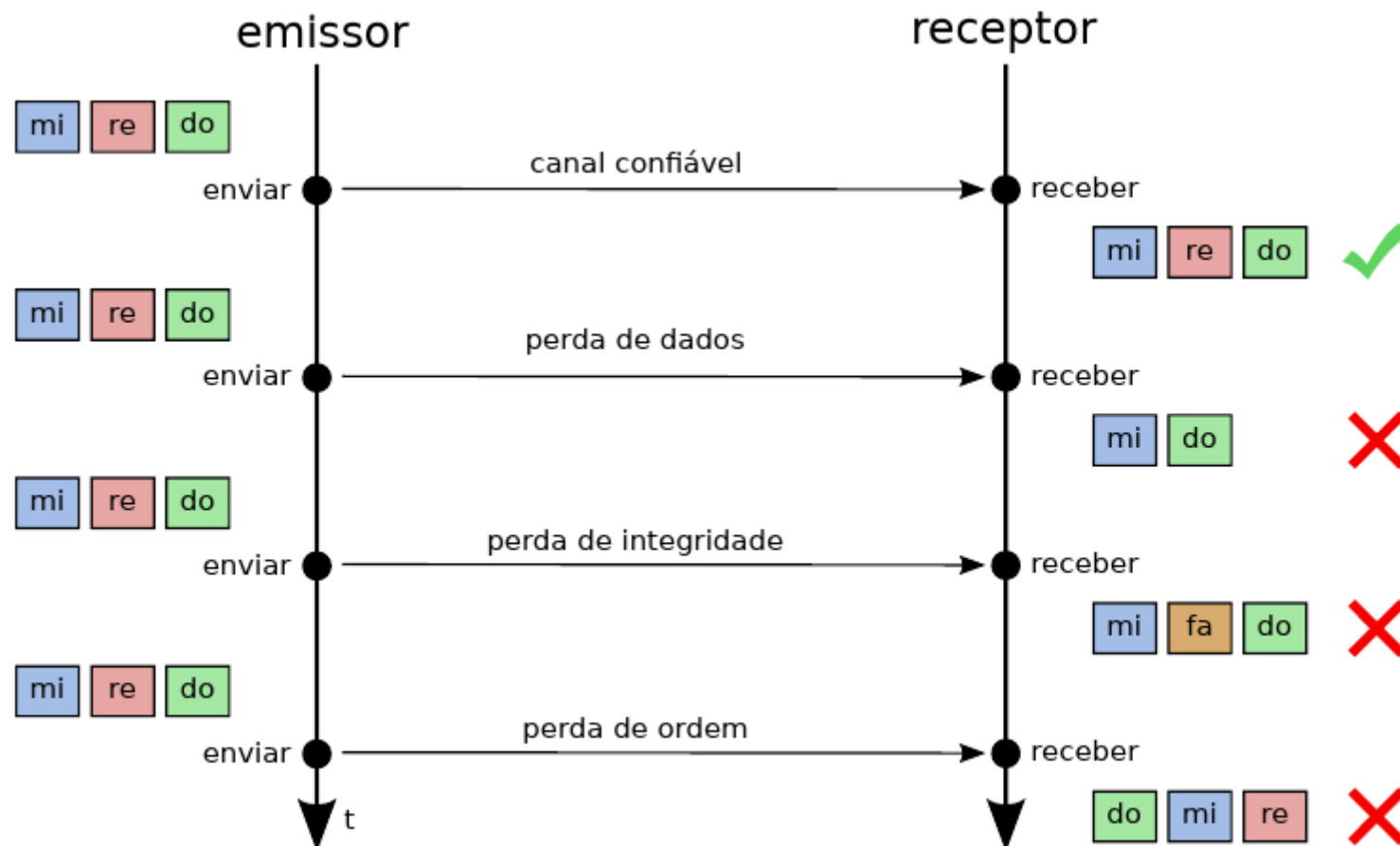


(mesma figura vista no slide 9)

# Capacidade dos canais

- O comportamento síncrono ou assíncrono de um canal de comunicação pode ser afetado pela presença de buffers
  - Capacidade nula ( $n = 0$ ): transferência direta sem cópias intermediárias
  - Capacidade infinita ( $n = \infty$ ): emissor sempre pode enviar dados, armazenados em buffer do canal enquanto o receptor não consome
  - Capacidade finita ( $0 < n < \infty$ ): ao tentar enviar dados em um canal já saturado, o emissor poderá ficar bloqueado até surgir espaço no buffer do canal e conseguir enviar (comportamento síncrono) ou receber um retorno indicando o erro (comportamento assíncrono)

- Canal confiável: transporta todos os dados enviados através dele para seus receptores, respeitando seus valores e a ordem em que foram enviados



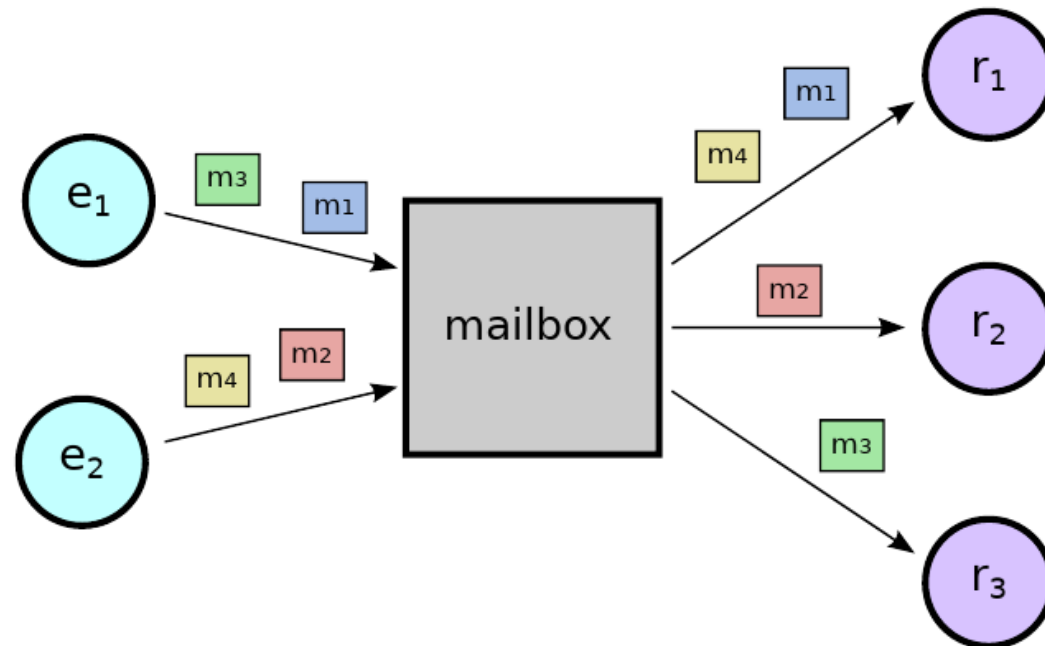
# Número de participantes

15/16

M : emissores

N : receptores

- M:N, através de um mailbox
  - Cada mensagem é recebida por apenas um receptor (em geral aquele que pedir primeiro)
  - A comunicação continua sendo ponto-a-ponto, através de um canal compartilhado
  - Essa abordagem é conhecida como mailbox
    - Funciona como um buffer de dados, no qual os emissores depositam mensagens e os receptores as consomem



- M:N, através de um barramento de mensagens
  - Cada mensagem é recebida por vários receptores
  - Conhecida como barramento de mensagens (message bus), canal de eventos ou ainda canal publish-subscribe

