Sistemas Distribuídos

Aula 4 – Comunicação entre processos (cont.)

DCC/IM/UFRRJ
Marcel William Rocha da Silva

Objetivos da aula

- **Aula anterior**
  - Processos e threads
  - Comunicação entre processos
  - *Middleware* de comunicação
    - Chamada de procedimento remoto (RPC)
- **Aula de hoje**
  - *Middleware* de comunicação
    - Comunicação orientada a mensagem
    - Comunicação orientada a fluxo
  - Multicast em SDs

Conteúdo Programático

- Introdução e visão geral
- Princípios de sistemas distribuídos
  - Arquiteturas
  - Processos
  - Comunicação
  - Nomeação
  - Sincronização
  - Consistência e replicação
  - Tolerância à falha
  - Segurança

RPC/RMI

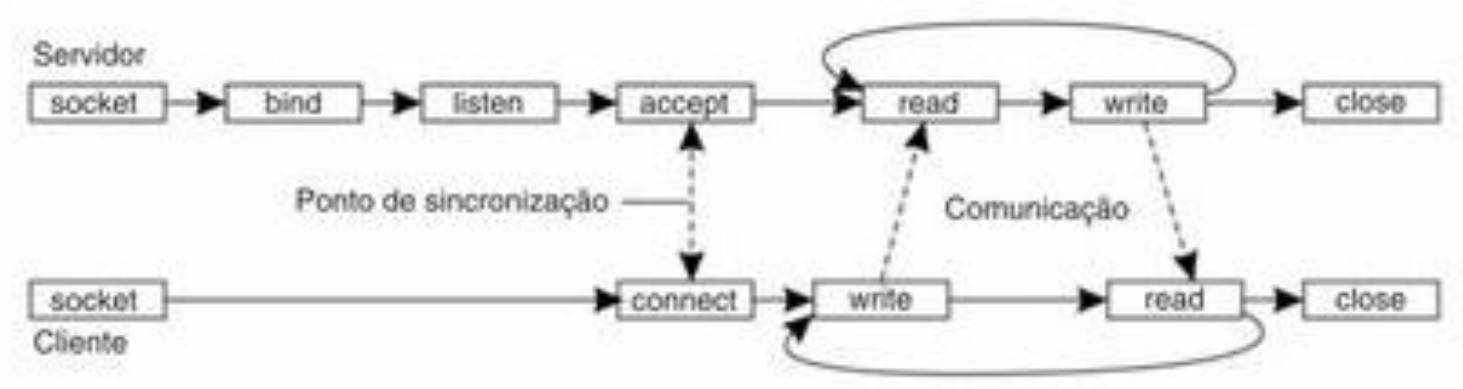
- Ocultam a complexidade da comunicação entre componentes de SDs
  - Transparência na comunicação!

- Mas nem sempre são eficientes
  - Quando receptor não está executando no momento da requisição
  - Natureza síncrona da comunicação RPC

Comunicação orientada a mensagem

- Os serviços fornecidos pela camada de transporte são um modelo simples orientado a mensagem

- Sockets TCP → possuem uma interface de uso bem definida
  - **`socket, bind, listen, accept, connect, send, receive, close`**
  - Suficiente em muitos cenários



Comunicação orientada a mensagem

- Outros tipos de interface de troca de mensagem também existem
- Ex.: MPI (*Message Passing Interface*)
  - Desenvolvida para rede de comunicação proprietárias, que não utilizam a pilha de protocolos TCP/IP
    - Que não possuem necessidade de comunicação via Internet
  - Fornece maior controle sobre as mensagens
  - Exemplos de primitivas para comunicação transiente
    - **`MPI_bsend`** → apenas anexa a mensagem em um buffer local
    - **`MPI_send`** → envia msg e espera ser copiada para buffer remoto
    - **`MPI_ssend`** → envia msg e espera o recebimento começar
    - **`MPI_sendrecv`** → envia msg e espera a resposta chegar

Comunicação orientada a mensagem

- Sistemas de Enfileiramento de Mensagens
  - Middleware orientado a Mensagem (MOM)
  - Proporcionam suporte extensivo para comunicação assíncrona persistente
  - Oferecem capacidade de armazenamento de médio prazo para as mensagens
  - Não exigem que o remetente ou o receptor estejam ativos durante a transmissão da mensagem

Comunicação orientada a mensagem

- Sistemas de Enfileiramento de Mensagens
  - Visam dar suporte para a transferências de mensagens que podem durar minutos, ao invés de apenas segundos ou milissegundos (como nos casos de sockets ou MPI)

Modelo de enfileiramento de mensagens

- Aplicações se comunicam inserindo mensagens em filas específicas
- Mensagens podem passar por vários servidores até serem entregues ao destinatário
- Cada aplicação tem sua fila particular para a qual outras aplicações podem enviar mensagens

Modelo de enfileiramento de mensagens

- A única garantia dada ao remetente é de que sua mensagem será inserida na fila de recepção do destinatário
- Não há garantia sobre quando e nem se a mensagem será realmente lida pelo destinatário
  - Ações totalmente determinadas pelo comportamento do receptor!
- Remetente e o receptor podem executar em completa independência um em relação ao outro
- Tão logo uma mensagem tenha sido depositada em uma fila, permanecerá até ser removida, independentemente de remetente e/ou receptor estarem em execução

Modelo de enfileiramento de mensagens

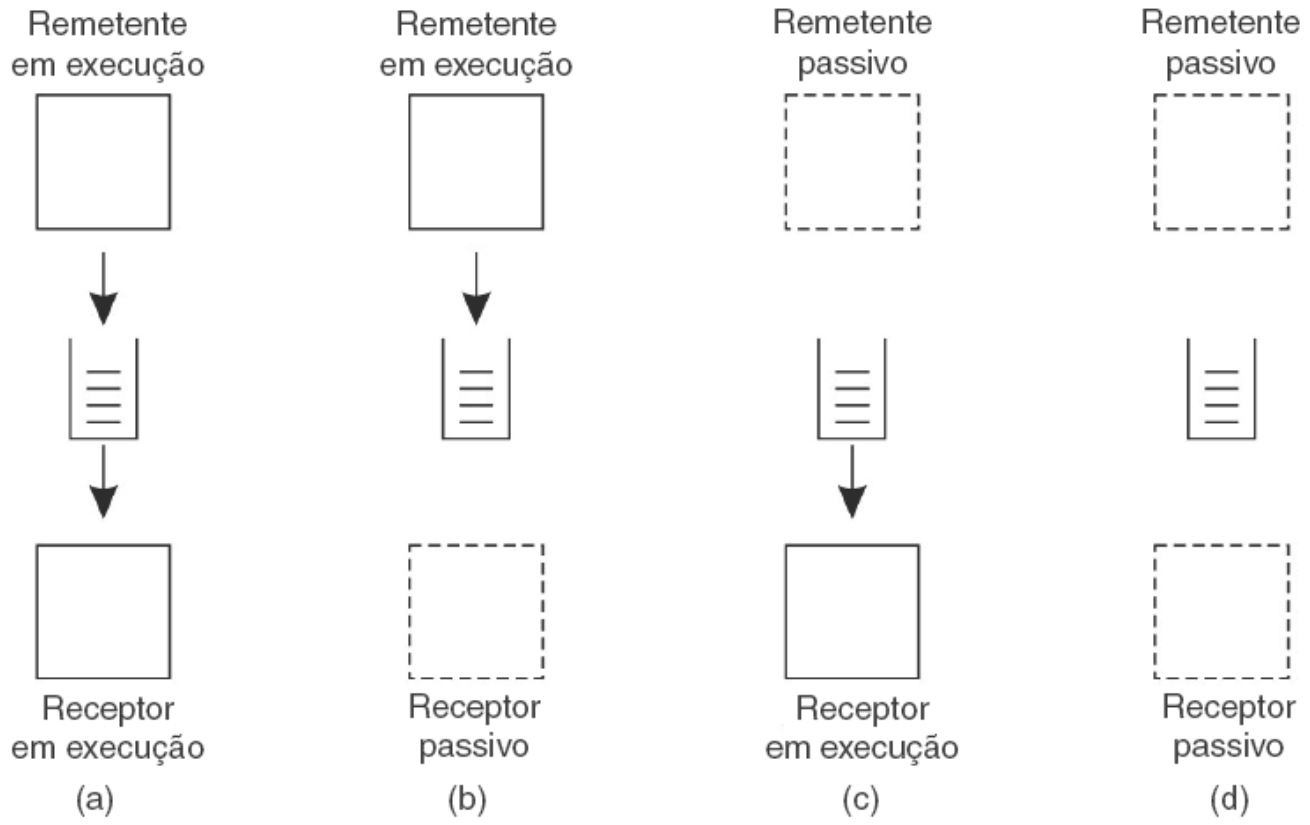


Figura 4.15 Quatro combinações para comunicações fracamente acopladas que utilizam filas.

Modelo de enfileiramento de mensagens

- Mensagens podem conter qualquer tipo de dado
  - Em alguns casos são de tamanho limitado
  - Middleware é responsável por fragmentar e remontar as mensagens grandes quando necessário

- Mensagens devem ser adequadamente endereçadas
  - O endereçamento é feito com o fornecimento de um nome exclusivo da fila de destino no âmbito do sistema

Modelo de enfileiramento de mensagens

- Interface básica

Primitiva	Significado
Put	Anexe uma mensagem a uma fila especificada
Get	Bloqueie até que a fila especificada esteja não vazia e retire a primeira mensagem
Poll	Verifique uma fila especificada em busca de mensagens e retire a primeira. Nunca bloqueie
Notify	Instale um manipulador a ser chamado quando uma mensagem for colocada em uma fila específica

Arquitetura da comunicação por enfileiramento de mensagens

- Filas
  - Mensagens somente podem ser colocadas em filas locais do remetente, na mesma máquina ou em uma máquina na mesma LAN → **filas de fonte**
  - Mensagem colocada em uma fila contém a especificação de uma **fila de destino**
  - Sistema de enfileiramento é responsável por fornecer filas para remetentes e receptores e providenciar para que as msg sejam transferidas de sua fila de fonte para a fila de destino

Arquitetura da comunicação por enfileiramento de mensagens

- Filas
  - Sistema de enfileiramento deve manter mapeamento de filas para localizações de rede (similar ao serviço de DNS)

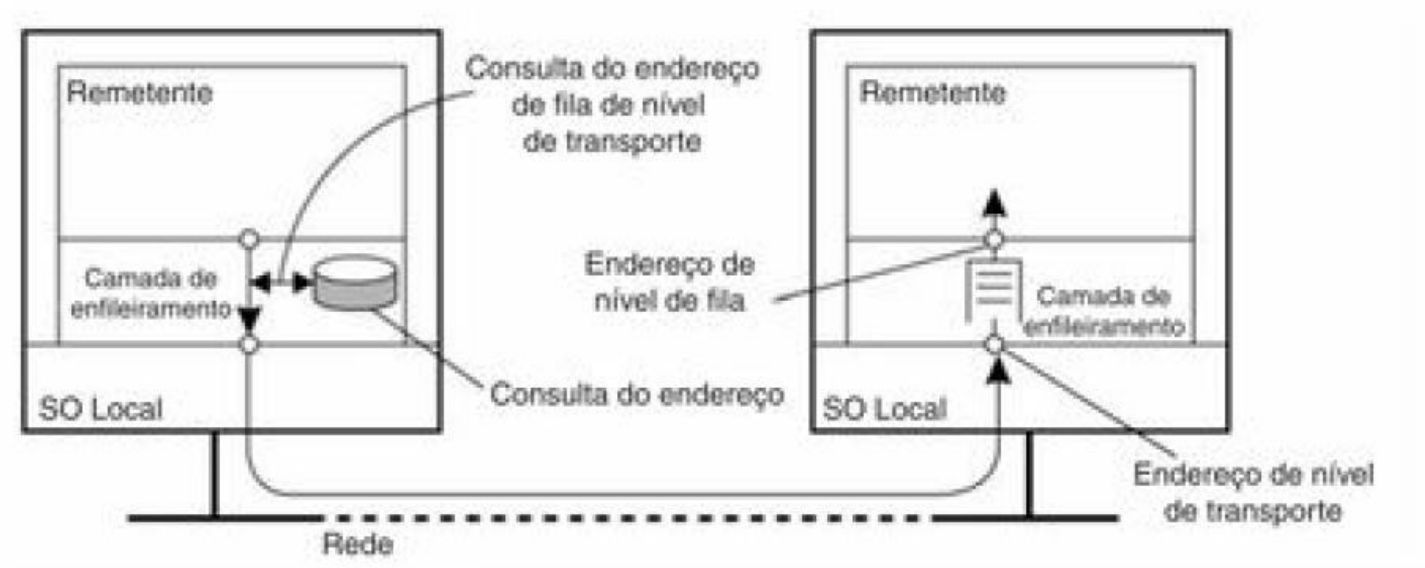


Figura 4.16 *Relação entre endereçamento de nível de fila e endereçamento de nível de rede.*

Arquitetura da comunicação por enfileiramento de mensagens

- Gerenciador de Filas
  - Interage diretamente com a aplicação que está enviando ou recebendo uma mensagem

- Repassadores
  - Gerenciadores especiais, que funcionam como roteadores
  - Sistema de enfileiramento pode crescer gradativamente até uma **rede de sobreposição** de nível de aplicação
  - Podem ser usados também para ***multicasting*** de msg

Arquitetura da comunicação por enfileiramento de mensagens

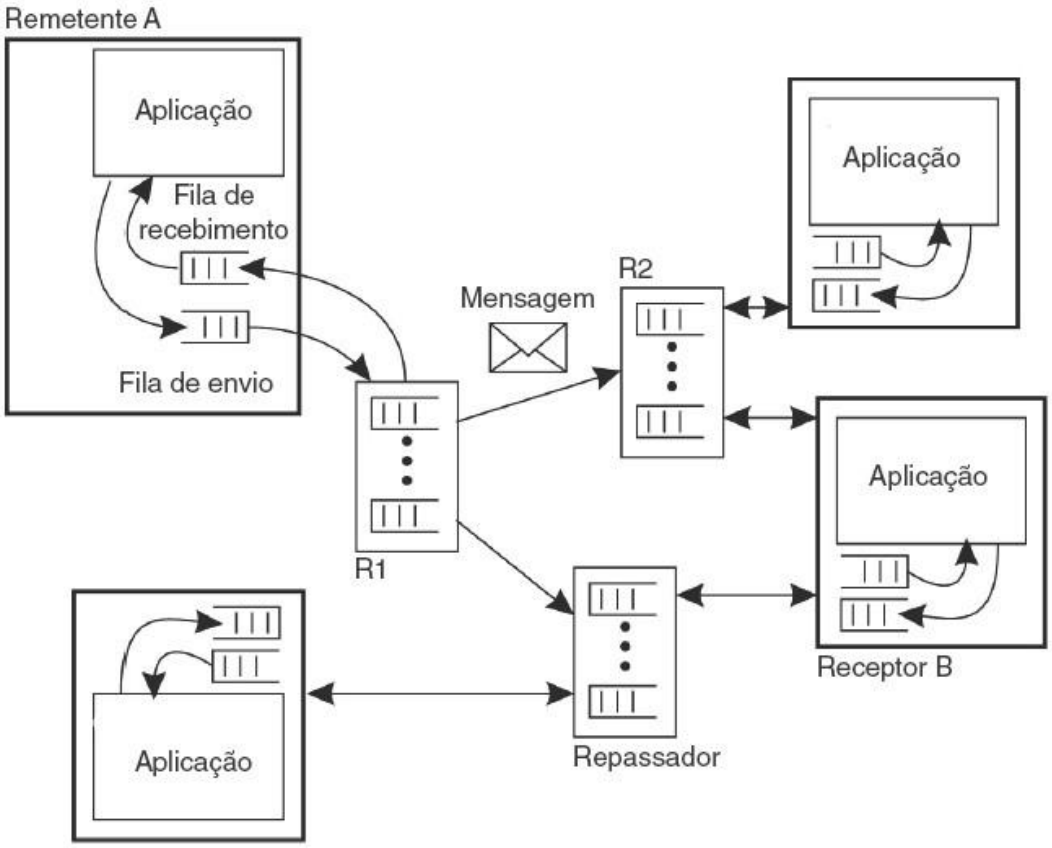


Figura 4.17 Organização geral de um sistema de enfileiramento de mensagens com repassadores.

Enfileiramento de msg vs. Email

- **Email**
  - Visa dar suporte a comunicação entre **usuários**
  - Requisitos específicos → filtragem automática de msg, armazenamento permanente e avançado de mensagens, etc

- **Enfileiramento de mensagens**
  - É mais genérico → dá suporte a comunicação entre **aplicações**
  - Requisitos diferentes do email → garantia de entrega (na fila do dest.), prioridade de msg, tolerância à falha, multicasting eficiente, etc

Até o momento...

- Troca de unidades de informação completas e independentes
  - Com um tamanho bem definido
  - **Mídia discreta**
  - Ex.: Requisição para invocar um procedimento

- Mas existem também dados dependentes do tempo
  - **Mídia contínua**
  - Fluxos de áudio e vídeo

Comunicação Orientada a Fluxo

- Tipos de fluxo
  - **Fluxos Simples**
    - Sequência simples de dados.
    - Ex: Voz
  - **Fluxos "Complexos"**
    - Consiste em vários fluxos simples relacionados denominados subfluxos
    - Existe uma relação temporal entre os subfluxos
    - Ex: Transmissão de um filme: vídeo, som, legenda

Qualidade de Serviço (QoS)

- Requisitos que descrevem o que é necessário para garantir que as relações temporais em um fluxo de dados possam ser preservadas

- Está relacionada com: Pontualidade, Volume e Confiabilidade

- Sistemas operacionais e redes TCP/IP não suportam QoS!
  - Sem garantias temporais no SO
  - Serviço IP é de “melhor esforço” (*best-effort*)

Qualidade de Serviço (QoS)

- Fatores que influenciam a QoS e suas causas
  - **Taxa de bits** → Codificação
  - **Atraso inicial** → Buffer de recepção
  - **Atraso fim-a-fim** → Caminho dos dados na rede
  - **Variação do atraso (jitter)** → Instabilidades da rede
  - **Taxa de perda de dados** → Idem ao anterior

Técnicas para prover QoS

- Buffer para reduzir jitter no receptor

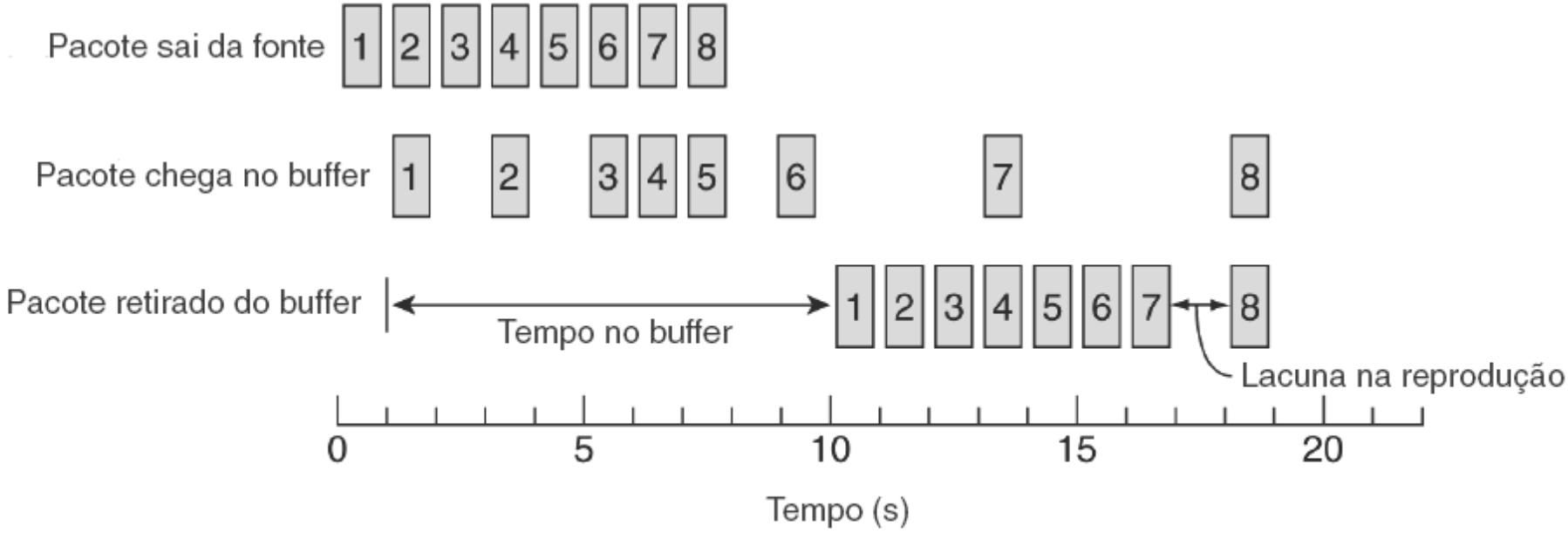


Figura 4.22 Utilização de um buffer para reduzir variância de atraso.

Técnicas para prover QoS

- Correção de erros antecipada (**FEC** – *Forward Error Correction*)

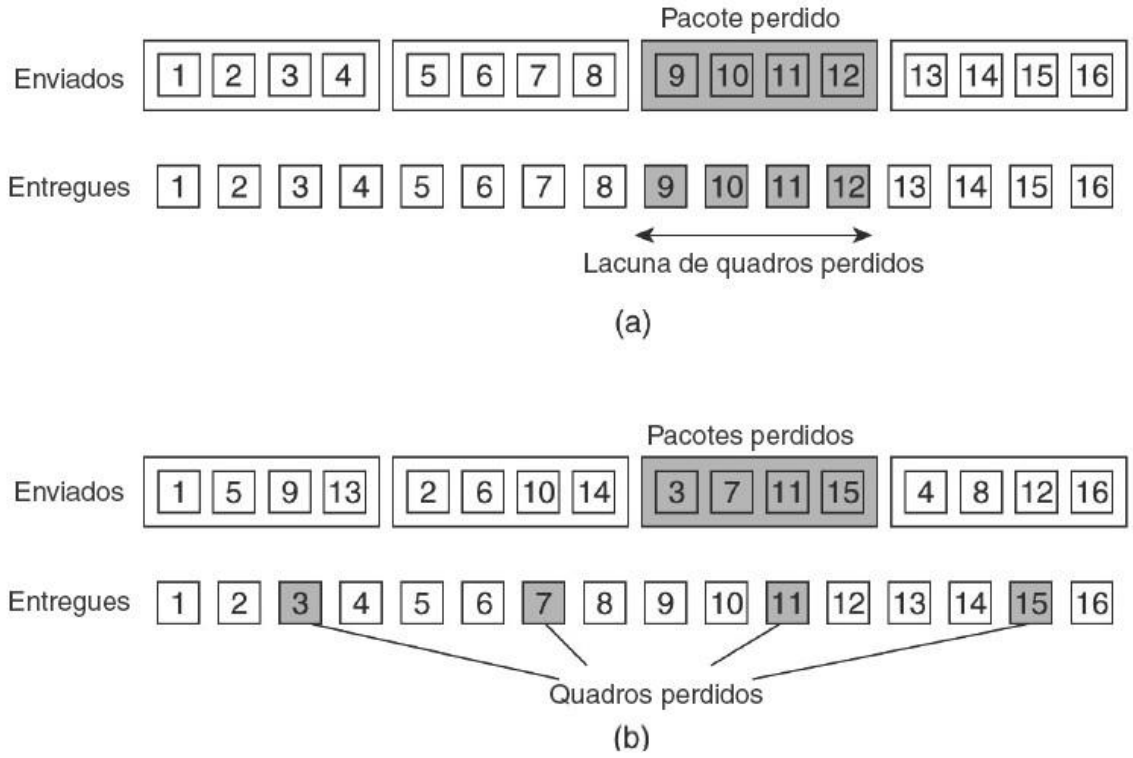


Figura 4.23 Efeito da perda de pacotes em (a) transmissão não intercalada e em (b) transmissão intercalada.

Comunicação multicast

- Diferentes tipos de comunicação:
  - **Unicast** – transmissão de um para um
  - **Broadcast** – transmissão de um para todos
  - **Multicast** – transmissão de um para vários

- Multicast na Internet...
  - Endereços IP de multicast → único endereço IP que referencia um grupo de destinatários
  - Mas não é tão simples...

Multicast na Internet

- IGMP → protocolo de gerenciamento de grupos
  - Apenas entre o sistema final e seu roteador (rede local)
  - Permite ao sistema final informar aos outros membros da rede local que uma aplicação quer se “juntar” a um grupo multicast
    - Mensagens para aquele endereço de multicast devem ser enviadas para o referido sistema final
  - Para outras redes saberem deste novo membro a informação deve ser propagada no núcleo
    - Protocolos de roteamento especializados (PIM, DVMRP, MOSPF)
  - Necessário investimento e esforço de gerenciamento → fracasso do multicast na Internet ☹

Multicast em SDs

- Multicast na Internet é custoso → Inviável

- Solução → **Multicast em camada de aplicação**
  - Utiliza uma **rede de sobreposição** → natural em vários tipos de sistemas distribuídos (Ex.: P2P)
  - Informação em multicast é disseminada na rede de sobreposição
  - Roteadores da rede física não precisam ter ciência da comunicação em multicast
    - Conexão entre nós da rede de sobreposição podem cruzar vários roteadores

Multicast em SDs

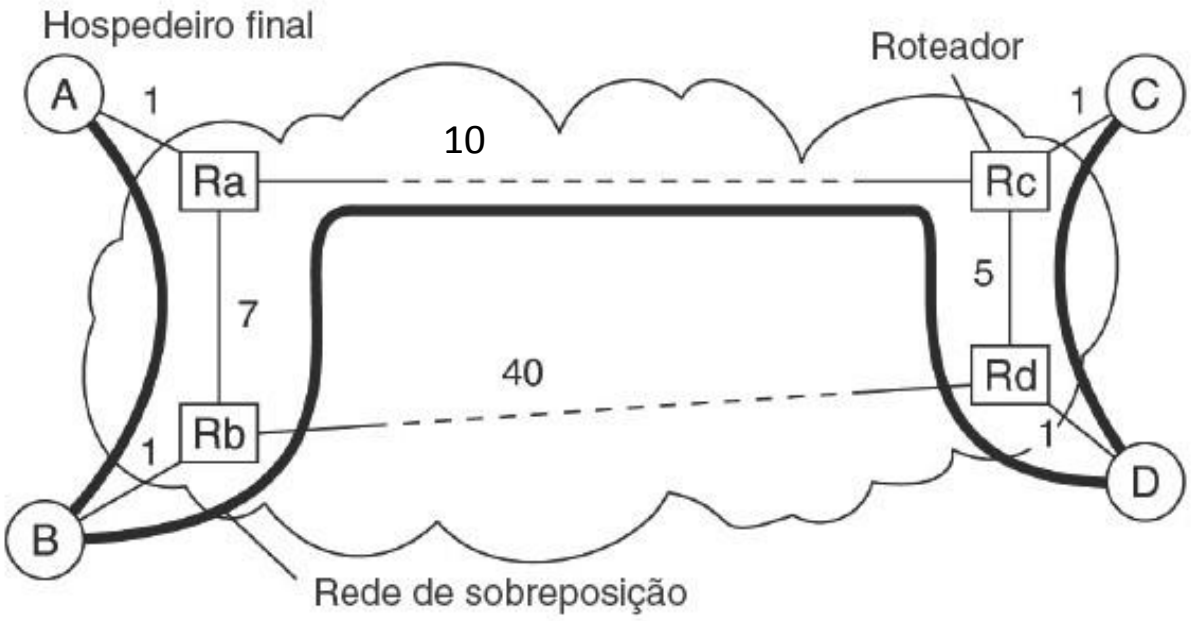
- Questões principais:
  - Como construir o overlay para obter robustez, diminuir o atraso de difusão ? (explorando, por exemplo características dos peers ou de localização geográfica)
  - Como difundir o conteúdo de maneira eficiente?

Multicast em SDs

- Estruturas possíveis
  - **Árvore**: Um único caminho entre cada par de nós.
    - Reorganização da estrutura a cada entrada/saída de nós
    - **Baixa resiliência**: a saída de um nó pode desconectar outros
  - **Mesh**: Neste tipo de overlay, os nós se organizam em uma **malha**
    - Existência (com grande probabilidade) de vários caminhos entre pares de nós
    - **Alta resiliência**

Multicast em SDs

- **Árvore**
  - Construir uma árvore é relativamente simples
  - Problema é ter uma árvore eficiente!
  - Desafio: **roteamento lógico vs. roteamento físico**



Multicast em SDs

- Qualidade da Árvore:
  - **Estresse de enlace**: Quantas vezes uma mensagem atravessa o mesmo enlace? Exemplo**:** mensagem de A a D atravessa Ra,Rb duas vezes
  - **Penalidade de atraso relativo**: Razão entre o atraso entre dois nós na sobreposição e o atraso que esses dois nós sofreriam na rede subjacente. Exemplo: mensagens de B a C
  - **Custo da árvore**: parâmetro de medição global, relacionado com a minimização dos custos agregados de enlaces. Exemplo: atraso entre dois nós finais

Multicast em SDs

- **Mesh**
  - Diferentes grafos podem ser usados
  - **Grafos aleatórios**: Nenhum tipo de informação é usado para construir topologias
    - Simples, mas pode ser ineficiente (lógico vs. físico)
  - **Grafos "com inteligência"**: Neste caso, podemos considerar informações como banda dos nós ou localização geográfica (através do RTT, por exemplo) para construção da topologia