

05 – Comunicação

- [illegible]



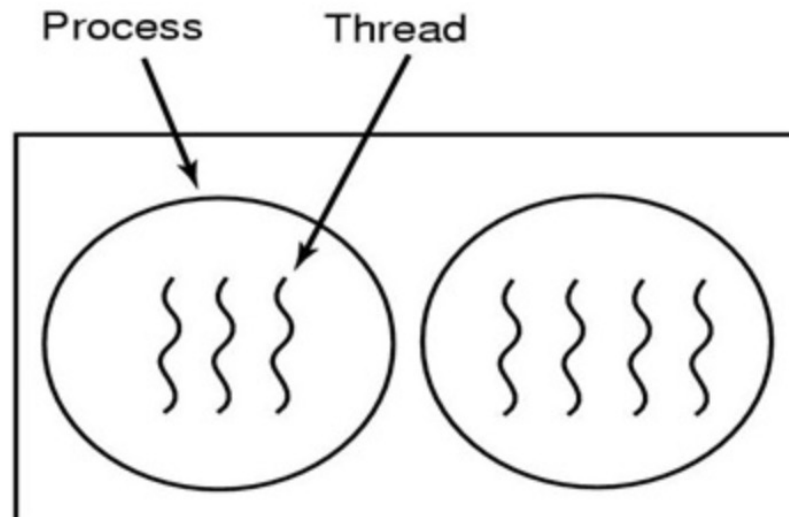
05 – Comunicação

- A comunicação é o coração de todo sistema distribuído
- Como os processos em diferentes máquinas trocam informações?



05 – Comunicação

- Como os processos em diferentes máquinas trocam informações? Cada processo possui um espaço de endereçamento, diferente de threads, que compartilham do mesmo espaço de endereçamento.



05 – Comunicação

- O Objetivo é prover transparência desta comunicação também ao desenvolvedor. A mesma deve ser transparente e de forma simples.

Exemplo:

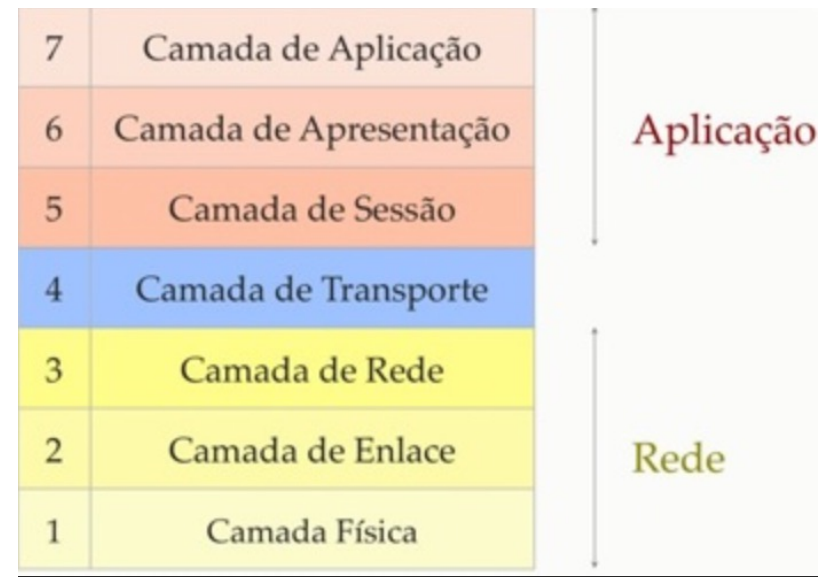
- Migração de Código,



05 – Comunicação

Protocolos em Camadas:

- Formam a base para qualquer Sistema Distribuído;
- Sem memória compartilhada: Comunicação por Mensagens, portanto através de pacotes de uma camada para a outra.

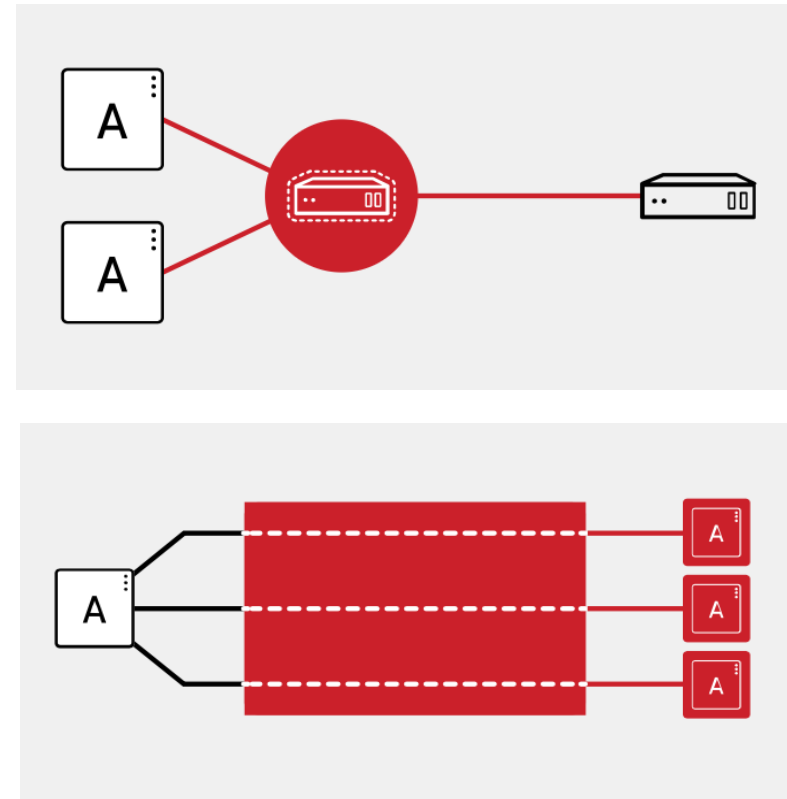


05 – Comunicação

Camada de Middleware

Dentro do cenário da camada de protocolos, podemos incluir a camada de Middleware: Faz a interface entre a camada de aplicação e a camada de transporte.

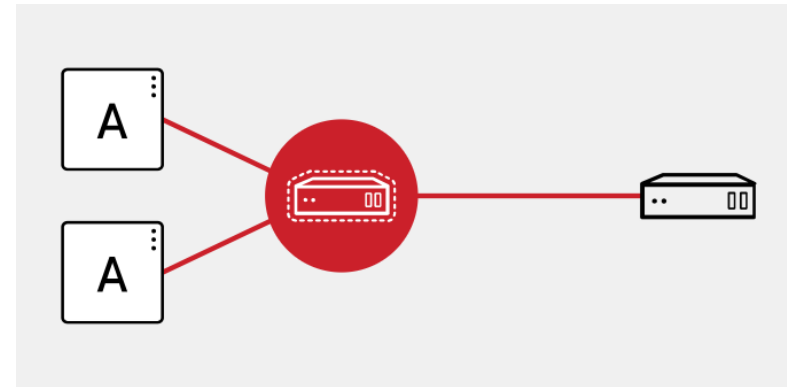
- Camada de Software que é situada logicamente entre a camada de aplicação e a camada de transporte;



05 – Comunicação

Camada de Middleware

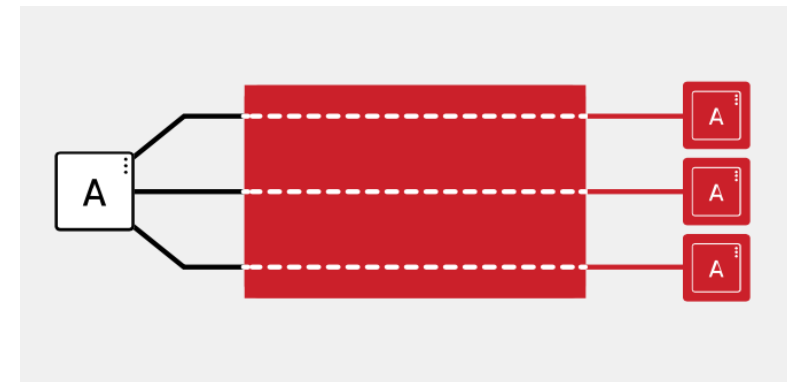
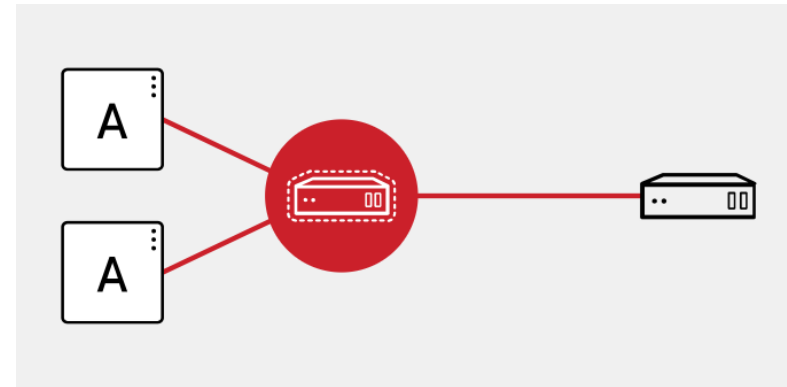
- Possui interface única (o desenvolvedor interage com essa interface única, essa interface prove acesso a aplicação móvel, site, desktop, etc.
- Por que interface única?



05 – Comunicação

Camada de Middleware

- Possui interface única (o desenvolvedor interage com essa interface única, essa interface prove acesso a aplicação móvel, site, desktop, etc.
- Por que interface única? Porque sendo única, o desenvolvedor aprende apenas uma vez e utiliza a mesma pra varias aplicações diferentes.
- Exemplo de Interface única: Posix (no Unix)



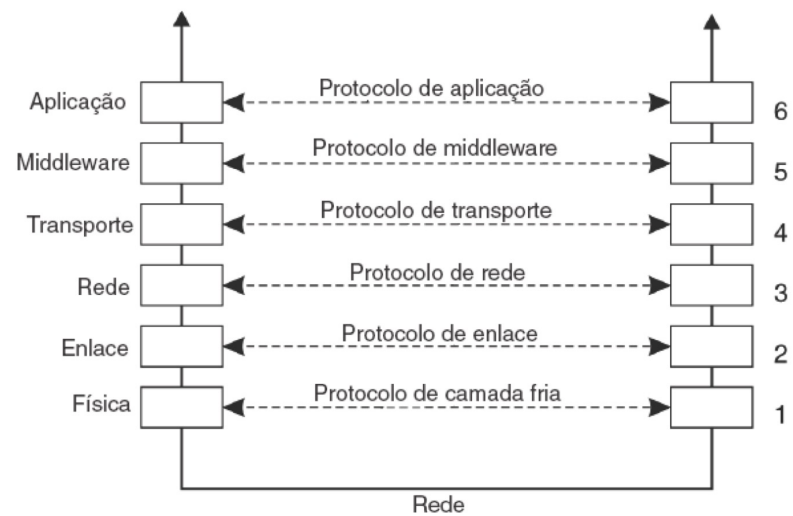
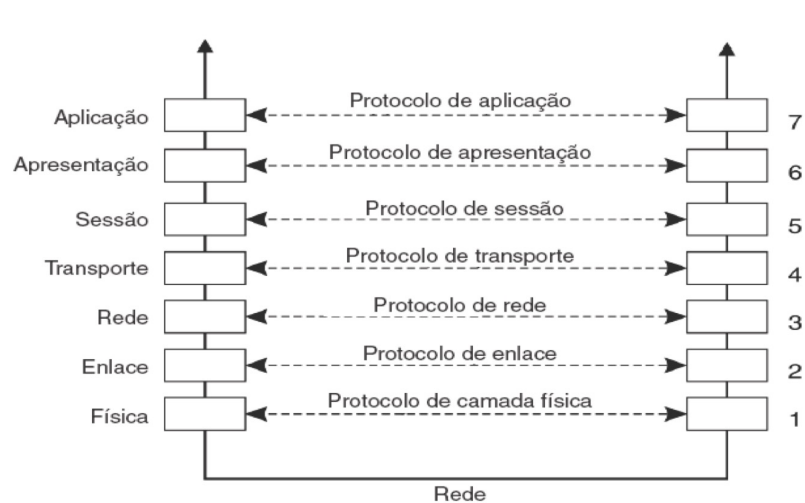
05 – Comunicação

Existem inúmeros protocolos para suportar serviços de middleware, abaixo alguns deles:

- **Protocolos para Autenticação:**
Não ligada a uma aplicação;
- **Protocolos de comprometimento;**
- **Protocolos Comunicação:** Evitar acessos simultâneos a um recurso



05 – Comunicação



05 – Comunicação

Tipos de Comunicação (quanto à persistência)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à persistência).

Quais são?



05 – Comunicação

Tipos de Comunicação (quanto à persistência)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à persistência).

Quais são?

São os persistentes e os transientes.



05 – Comunicação

Tipos de Comunicação (quanto à persistência)

Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à persistência).

Quais são?

São os persistentes e os transientes.



05 – Comunicação

Tipos de Comunicação (quanto à persistência)

- **Persistente:** Mensagem armazenada durante o tempo que for necessário para entregá-la ao receptor.



05 – Comunicação

Tipos de Comunicação (quanto à persistência)

- **Transiente:** Mensagem armazenada somente durante execução do remetente e receptor.

Obs.: Middleware pode ou não armazenar as mensagens.



05 – Comunicação

Tipos de Comunicação (quanto à sincronização)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à sincronização).

Quais são?



05 – Comunicação

Tipos de Comunicação (quanto à sincronização)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à sincronização).

Quais são?

Assíncrona e Síncrona.



05 – Comunicação

Tipos de Comunicação (sincronização)

- **Assíncrona:** Remetente continua sua execução imediatamente após enviar mensagem.



05 – Comunicação

Tipos de Comunicação (sincronização)

- **Síncrona:** Remetente é bloqueado até saber que sua requisição foi aceita.

Dependendo do ponto de sincronização:

- *Remetente bloqueado até que o middleware avise.*
- *Remetente bloqueado até a requisição chegar no receptor.*
- *Remetente bloqueado até o receptor responder.*



05 – Comunicação

Tipos de Comunicação (quanto à granuralidade)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à granuralidade).

Quais são?



05 – Comunicação

Tipos de Comunicação (quanto à granuralidade)

- Existem dois tipos de comunicação em Sistemas Distribuídos (quanto à granuralidade).

Quais são?

Discreta e Fluxo.



05 – Comunicação

Tipos de Comunicação (granuralidade)

- **Discreta:** Comunicam por mensagens, onde cada uma delas é uma unidade de informação completa (mensagem enviada cliente/servidor e servidor/cliente).



05 – Comunicação

Tipos de Comunicação (granuralidade)

- **Fluxo:** comunicam por várias mensagens que estão relacionadas uma com as outras (Ex.: vídeo, voz).

Obs.: Relacionada pela ordem ou pela relação temporal.



05 – Comunicação

Protocolos de Comunicação (Middlewares de Comunicação)

Serviços de Comunicação que podem ser adicionados a camada de middleware e seus tipos:

- Chamadas de procedimento remoto;
- Comunicação orientada a mensagens;
- Comunicação orientada a fluxo;

05 – Comunicação

Protocolos de Comunicação (Middlewares de Comunicação)

Serviços de Comunicação que podem ser adicionados a camada de middleware e seus tipos:

- Chamadas de procedimento remoto;
- Comunicação orientada a mensagens;
- Comunicação orientada a fluxo;

**Remote
Procedure
Call**

05 – Comunicação

Protocolos de Comunicação (Middlewares de Comunicação)

É um middleware de comunicação que permite a processos chamar procedimentos localizados em outras máquinas (Birrel and Nelson, 1984);

A ideia é fazer com que a chamada de procedimento remoto pareça com uma chamada local;

**Remote
Procedure
Call**

05 – Comunicação

Protocolos de Comunicação (Middlewares de Comunicação)

Problemas: Passar parâmetros em espaços de endereçamento diferentes (é complexo, pois trabalhamos com arquiteturas que podem ser diferentes, queda da máquina durante transferência de parâmetros, etc.

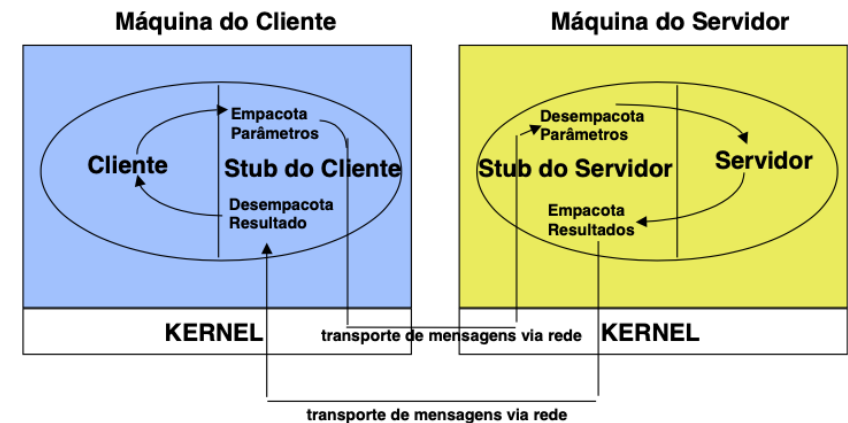
**Remote
Procedure
Call**

05 – Comunicação

RPC e os Stubs (permite que uma chamada remota pareça local)

A transparência é alcançada através dos stubs (apêndices):

- Stub do cliente: Empacota os parâmetros em uma mensagem e a envia para a máquina do servidor;
 - Desempacota e compatibiliza a arquitetura;
- Stub do servidor: Desempacota os parâmetros e invoca o procedimento correto passando os parâmetros;
 - Empacota os parâmetros na mensagem de resposta;
 - Ambos são compilados antes;



05 – Comunicação

Passos para uma RPC

Empacotar parâmetros em uma mensagem é conhecido como **marshalling** de parâmetro.

RPC suporta passagem de parâmetros das seguintes formas:

- Passagem por **valor**.
- Passagem por **referência** (endereço do dado);



05 – Comunicação

RPC–Linguagem de Programação de Interface (IDL)

Interface: É um conjunto de procedimento que pode ser chamado por um cliente e implementado por um servidor. *Pode ser usada em Java, C#, C++, etc*

IDL: é a linguagem voltada para especificar a interface e permite definir procedimentos como idempotentes.

Há aplicações, como stubgen, que geram o stub a partir da IDL (descrição própria que independe da linguagem).



05 – Comunicação

Protocolos de Comunicação (Middlewares de Comunicação)

Serviços de Comunicação que podem ser adicionados a camada de middleware e seus tipos:

- Chamadas de procedimento remoto;
- Comunicação orientada a mensagens;
- Comunicação orientada a fluxo;



05 – Comunicação

Comunicação Orientada a Mensagem:

- Middlewares RPC e RMI podem ser inadequados (p1 e p2 tem que estar sempre ativos).
- Receptor sempre “acordado”;
- O comportamento de sincronismo e bloqueio pode ser inadequado em ambientes com dispositivos voláteis.
 - RSSF;
 - IoT;
 - Computação Oblíqua e Pervasiva;
- Como contornar as limitações? Orientando a mensagens.

05 – Comunicação

Comunicação Orientada a Mensagem:

Interface de Troca de Mensagens (MPI)

- Aumento das arquiteturas paralelas e data centers;
- Objetivo: Escrever com facilidade aplicações para essas arquiteturas paralelas;
- Ideal seria: Independência de hardware e de arquitetura que suportassem comunicação orientada a mensagem.

05 – Comunicação

Comunicação Orientada a Mensagem:

Interface de Troca de Mensagens (MPI)

- Por que não usar sockets?
- Ferramenta muito genérica para modelar comunicação remota para TCP/IP (sincronismo e buffer limitados);
- MPI: Considerado um padrão de troca de mensagens para clusters;
- Modelo voltado para aplicações paralelas (e.g.funções)
- Comunicação transiente: Mensagem é armazenada no sistema enquanto remetente e receptor estiverem ativos;

05 – Comunicação

Comunicação Orientada a Mensagem:

Algumas primitivas da MPI

Possuí mais de 100 funções diferentes para troca de mensagens:

- MPI_send: envia mensagem e espera até que seja copiado para um buffer;
- MPI_send recv: Envia mensagem e espera por uma resposta;
- MPI_ALLtoall: Envia uma mensagem de todos para todos os nós que participam da comunicação.
- Provê funções de passagem de mensagem para as linguagens como C, C++ e Fortran.

05 – Comunicação

Comunicação Orientada a Mensagem:

MPI (características)

- A comunicação ocorre dentro de um grupo conhecido de processos;
- Cada grupo tem um identificador próprio;
- Cada processo dentro de um grupo recebe um identificador (local);
- Par (groupId, processID) identifica fonte ou destinatário único;
- Vários grupos de processos podem fazer parte de uma computação;

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM)

- Suporte para comunicação assíncrona e persistente;
- Capacidade de armazenamento de médio prazo para mensagens trocadas
- Ideia básica: Aplicações se comunicam retirando e inserindo mensagens em filas específicas;
- Mensagem será eventualmente entregue ao receptor;
- Comunicação fracamente acoplada;

05 – Comunicação

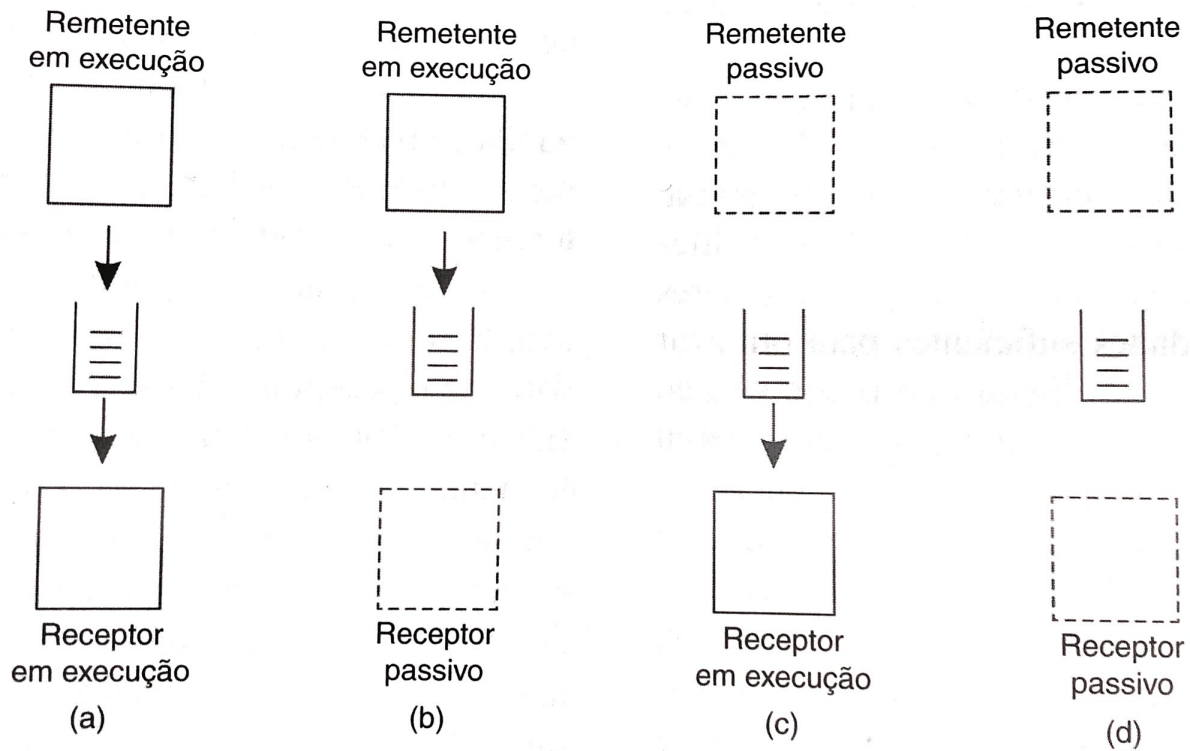
Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – ideia básica:

- Aplicações se comunicam inserindo mensagens em filas específicas;
- As mensagens são repassadas por uma série de servidores de comunicação;
- Essas são entregues ao destinatário mesmo que ele não esteja em funcionamento;

Exemplo: Email (remetente e receptor podem executar em completa independência);

05 – Comunicação



05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – arquitetura;

- Fila de Fonte;
- Fila de Destino;
- Gerenciadores de Fila;
- Repassadores;

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – arquitetura;

- Fila de Fonte: Fila no qual o remetente envia a mensagem, filas locais do remetente ou próximas a ele.

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – arquitetura;

- Fila de Destino: Identificador da fila de destino para onde ela deve ser transferida;

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – arquitetura;

- Gerenciadores de Fila: Interage com a aplicação que está enviando ou recebendo a mensagem (e.g. fila FIFO?).

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – arquitetura;

- Repassador: Gerenciadores de filas especiais que repassam mensagens para outros gerenciadores.

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – Componentes da Arquitetura.

- A organização da fila é feita pelo gerenciador de fila (FIFO?).
- Repassadores também são chamados de roteadores;

05 – Comunicação

Comunicação Orientada a Mensagem:

Middleware orientando a mensagem (MOM) – Mais informações sobre os repassadores:

- Repassadores ajudam a construir sistemas escaláveis de gerenciamento de fila;
- Atualizações de remoção e adição de filas devem ser informadas aos repassadores;
- Gerenciadores de fila devem saber onde está o repassador mais próximo (escalabilidade);

05 – Comunicação

Comunicação Orientada a fluxo

- Lidamos até agora com mensagens mais ou menos completas e independentes;
- Comunicação orientada a fluxo:
 - Mensagem de fluxo de áudio e vídeo;
 - Dependentes de tempo;
 - E o tempo é crucial;

05 – Comunicação

Comunicação Orientada a fluxo

- Quando falamos de fluxo estamos lidando com Qualidade de Serviço (Qos);
- Está relacionada com:
 - Atraso;
 - Variação de atraso;
 - Largura de banda;
 - Redes IPv6: mais suporte para o QoS do que o IPv4;
 - QoE (Quality of Experience).

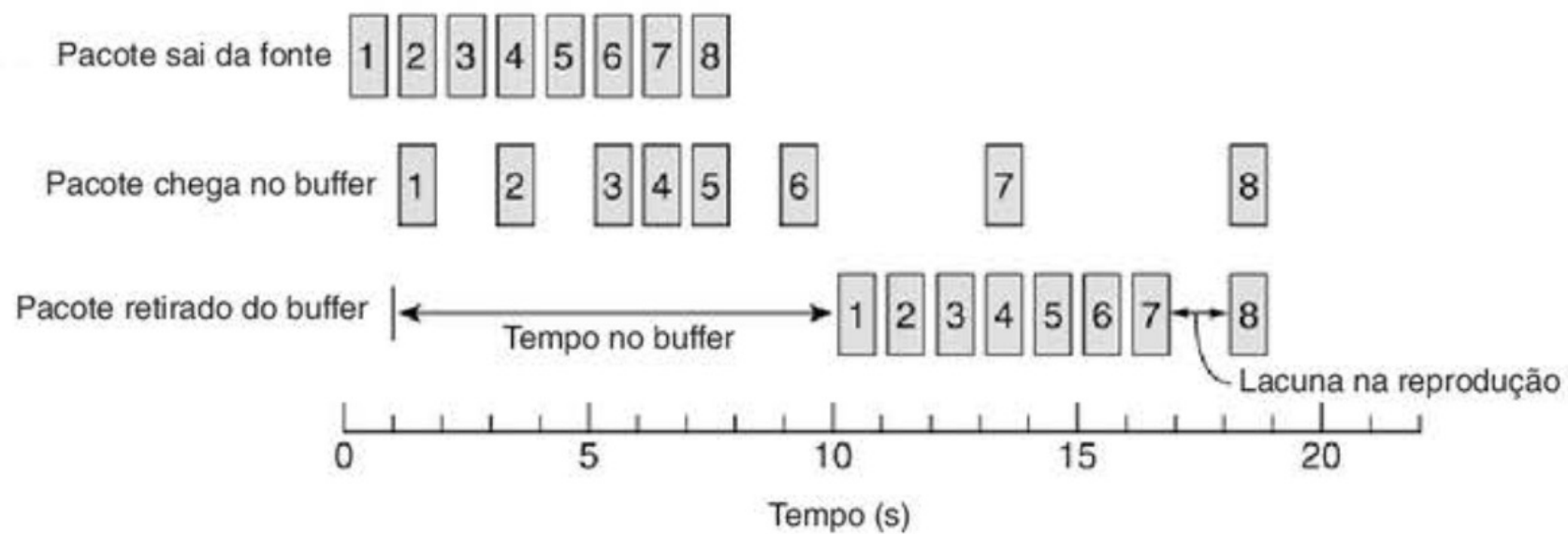
05 – Comunicação

Comunicação Orientada a fluxo

Técnicas para QoS

- Serviço diferenciado (divide em classes de serviços)
- Bufferização para reduzir a variância de atraso no receptor;

05 – Comunicação



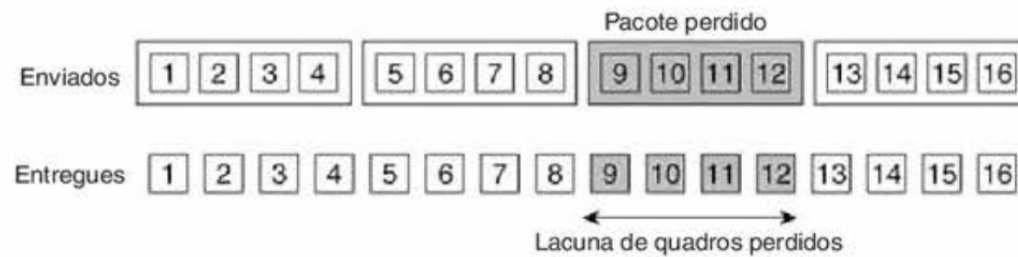
05 – Comunicação

Comunicação Orientada a fluxo

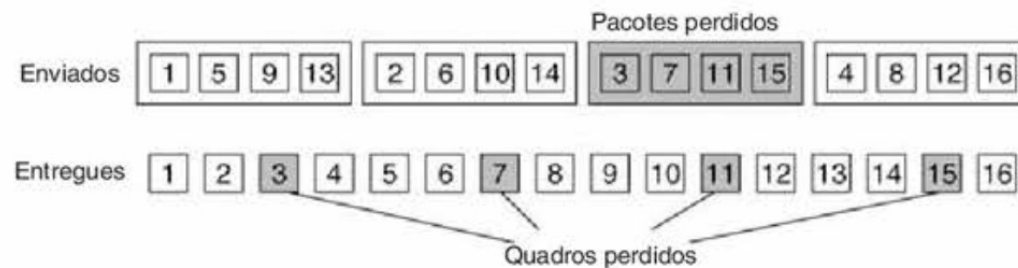
Técnicas para QoS

- Correção de Erro de Envio (Forward Error Correction – FEC).

05 – Comunicação



(a)



(b)