

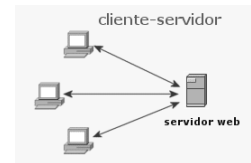
# • Capítulo 2

## • Conceitos básicos de comunicação de redes

Prof. Ms. Adinarte Correa da Silva  
falecom@adinarte.eti.br

## Cliente Servidor

- ♦ O sistema cliente/servidor é um paradigma lógico para o processamento distribuído, sendo que o cliente e o servidor podem estar ou não em uma mesma máquina física.
- ♦ Em um processamento cliente/servidor os processos cliente enviam pedidos a um processo servidor, que responde com resultados para esses pedidos.



## Cliente Servidor

- ♦ O servidor, na verdade, oferece serviços aos seus clientes, por um meio de processamento específico e a interação entre os processos cliente e servidor é uma troca cooperativa, transacional, em que o **cliente é ativo** e o **servidor é reativo**. Sendo esta, a principal distinção entre cliente/servidor.
- ♦ O usuário do sistema interage com um cliente, que por sua vez emite pedidos e recebe resultados do servidor.
- ♦ O estilo de integração entre o cliente e o servidor não precisa utilizar um interface gráfica ou ainda uma interface de caracteres em tela cheia. É possível construir um sistema cliente/servidor usando apenas uma interface de linha de caracteres.

## Serviços Orientados a conexão e sem conexão

- ♦ As camadas de uma arquitetura de rede podem oferecer diferentes classes de serviços às camadas superiores.
- ♦ Estes serviços podem ser orientados a conexão ou não orientados a conexão (também chamado sem conexão).
- ♦ **Serviços Orientado a conexão:**
  - ♦ Equivalente ao sistema telefônico
    - ♦ Estabelecimento de conexão;
    - ♦ uso da conexão;
    - ♦ encerramento da conexão;
  - ♦ O aspecto principal da conexão é o fato de que ela funciona como uma espécie de canal virtual através do qual irão transitar as mensagens envolvidas na realização do serviço.
  - ♦ O circuito virtual determina o roteamento uma única vez para a conexão
  - ♦ pode-se reservar banda para a conexão

## Serviços Orientados a conexão e sem conexão

### Serviços sem conexão:

- ♦ Equivalente ao serviço postal
  - ♦ Cada mensagem possui o seu endereço de destino;
  - ♦ As mensagens/pacotes são roteadas no sistema, independentemente uma das outras;
  - ♦ Possibilidade de perda de seqüência das mensagens.

## Tipos de Serviços

- ♦ Os serviços podem ser divididos em **confiável** e **não-confiável**.
- ♦ Um serviço dito **confiável** é aquele em que os dados não podem ser jamais perdidos, ou melhor, jamais podem deixar de realizar o serviço adequadamente e, por vezes, algum mecanismo de recuperação em caso de falha deve ser utilizado.
- ♦ Serviços **não confiáveis** são aqueles onde, eventualmente, dados podem ser perdidos e não recuperados pela camada em questão.
- ♦ Normalmente, a implementação de serviços confiáveis é feita através da definição de mensagens de reconhecimento enviadas pelo receptor, para cada mensagem recebida do emissor.
- ♦ Este processo, embora extremamente benéfico, introduz uma lentidão na transferência de dados, o que significa que nem sempre ele é desejável num sistema.

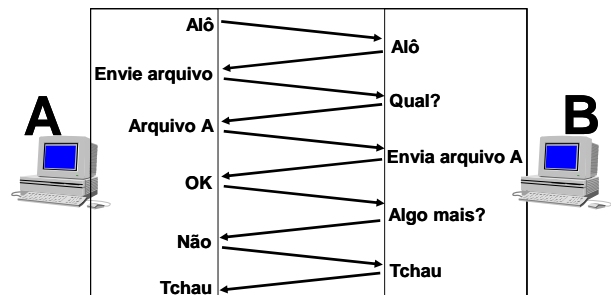
## Protocolos

- Os PROTOCOLOS definem um **conjunto de regras** e convenções que permitem especificar aspectos da realização do serviço, particularmente, o significado dos quadros, pacotes ou mensagens trocadas entre as entidades pares de uma dada camada.
- A nível de uma determinada camada, o protocolo pode ser mudado sem problemas, desde que as interfaces com a camada superior e inferior não sejam alteradas, ou seja, que elas continuem a ter a mesma visibilidade no que diz respeito aos serviços realizados pela camada que foi alterada

Na área das comunicações, um protocolo é um conjunto de regras ou convenções que governam a operação e o intercâmbio de informações entre dois sistemas computadorizados, ou seja, é um conjunto de regras que governam uma conversação

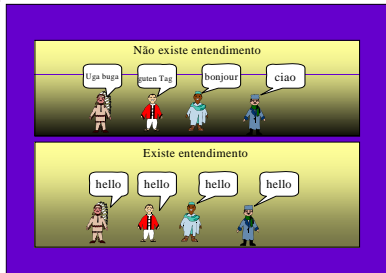
## Protocolos

- Conversação de uma pilha de protocolo;



## Protocolos

- Suponhamos que um turco queira conversar com um japonês. Se um não souber falar a língua do outro, mas se ambos souberem falar, digamos, inglês, eles podem se utilizar do "protocolo" inglês para se comunicar.



## Alguns Protocolos

- O Protocolo a qual estaremos nos baseando neste curso é a Suíte de Protocolos TCP/IP, mas, existe uma numero infundável de outros protocolos, proprietários e de estrutura aberta, que foram caindo em desuso pela próprio sucesso do TCP/IP ou pela mudança de tecnologia / necessidades.
- Alguns Protocolos não mais existem, outros são raramente utilizados.
- Estaremos citando uma breve apresentação de alguns protocolos comumente ainda utilizados.

## Protocolos

- SNA - System Network Architecture**
  - Criado pela IBM em 1974, é uma arquitetura de redes que define procedimentos e estruturas de comunicações entre programas de aplicação ou entre estes programas e um terminal.
  - Protocolo extremamente complexo;
  - Este protocolo surgiu antes que o modelo OSI fosse criado e utiliza sete camadas com as mesmas funções do RM/OSI.
  - Protocolo bastante usado em grandes corporações com processo críticos rodando em Mainframes.
  - Como o SNA foi criado para redes que utilizavam Mainframe e com o passar dos anos estes Mainframes foram sendo substituídos por minicomputadores e os terminais burros ligados a eles por PCs, a IBM lançou a segunda geração do SNA, o APPN (Advanced Peer-to-peer Networking), que dispensa o uso de mainframes.

## Protocolos

- NetBios - Network Basic Input / Output System**
  - A interface NetBIOS foi introduzida pela IBM em 1984 e usada pela Microsoft no sistema operacional de rede MS-Net.
  - O NetBIOS é uma API de programação do Protocolo NetBEUI;
  - Trabalha na camada 5 (sessão) do Modelo OSI;
  - O NetBIOS geralmente é utilizado com outros protocolos, como o TCP/IP, operando nas camadas abaixo da 5 do modelo OSI.
  - Atualmente a maior parte dos fornecedores de sistemas operacionais de redes (IBM, Novell, Microsoft, 3Com, etc.) incluem a interface NetBIOS em seus produtos, implementada em um driver de protocolo, ou implementada por um emulador (com as funções de transmissão sendo executadas por outro driver de protocolo, como fez a Novell).

## Protocolos

### ◆ NetBEUI – NetBIOS Extended User Interface

- ◆ O NetBEUI trabalha nas camadas 3 (rede) e 4 (transporte) do modelo OSI;
- ◆ Protocolo pequeno e rápido, mas inviável para redes com mais de 50 dispositivos;
- ◆ Protocolo não roteável;
- ◆ Utiliza muitas mensagens de Broadcast, congestionando a rede sem necessidade.
  - ◆ Caso uma máquina queira imprimir um documento na impressora da rede, o protocolo envia um broadcast na rede e não somente para o servidor de impressão.
- ◆ É o mais freqüente protocolo utilizado pelos produtos Microsoft (windows 95/98/Me, Windows 2000 e Windows NT). O windows XP não vem com este protocolo instalado por default, é necessário instalá-lo manualmente.

## Protocolos

### ◆ IPX/SPX

- ◆ Protocolo Proprietário criado pela Novell para seu sistema Operacional Netware;
- ◆ Utiliza os mesmos moldes do TCP/IP;
- ◆ É um protocolo Roteável;
- ◆ O protocolo IPX (Internet Packet Exchange) é o protocolo usado pela Novell para o nível de rede(camada 3 OSI).
  - ◆ Utiliza o MAC Address para diferenciar as máquinas da rede(auto-configurável);
  - ◆ O IPX fornece um serviço de datagrama não-confiável a seus usuários
- ◆ O SPX (Sequenced Packet Protocol) é o protocolo usado pela Novell para o nível de transporte (ver modelo OSI).
  - ◆ O SPX implementa um serviço de circuito virtual, ou seja, mecanismos de controle de erro, de fluxo e de sequenciação.

## Protocolos

### ◆ Apple Talk

- ◆ AppleTalk é uma arquitetura de redes proprietária da Apple.
- ◆ A primeira rede da família AppleTalk foi a LocalTalk, projetada para atender pequenos grupos de usuários. Mas a medida que os Macs foram se tornando mais populares, as redes LocalTalk foram aumentado.
- ◆ Para suprir essa necessidade foi lançado a EtherTalk, uma rede que atendia serviços AppleTalk em rede de alta velocidade, na época, a Ethernet.

### ◆ TCP/IP – Transmission Control Protocol / Internet Protocol

- ◆ É um conjunto de protocolos de comunicação utilizado para troca de dados entre computadores em ambientes de rede local (LAN) ou em rede de longas distâncias (WAN).

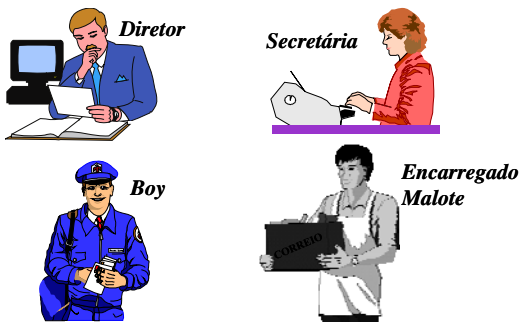
## Arquitetura Multicamadas

### ◆ Solução do problema de comunicação

- ◆ Dada a complexidade do problema, a melhor forma de encontrar a solução é decompor a tarefa de comunicação em **subproblemas**, que serão tratados individualmente...
- ◆ A solução para o problema da comunicação entre sistemas computacionais pode ser inspirada em algumas modalidades de comunicação entre seres humanos

## Arquitetura Multicamadas

### ◆ Comunicação numa empresa



## Arquitetura Multicamadas

### ◆ Comunicação numa empresa



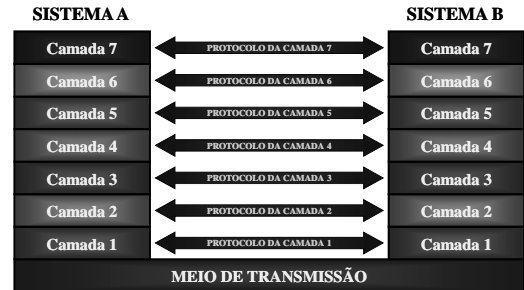
## Arquitetura Multicamadas

### ◆ Vantagens da Arquitetura Multicamadas

- ◆ **Facilidade de estudo e de implementação** da rede a partir de elementos de base existentes, o que permite a **redução dos custos de instalação**
- ◆ **Simplificação de sua operação** em função da definição de regras formais
- ◆ **Garantia de confiabilidade** de um sistema que seja aceitável, particularmente graças ao **encapsulamento das funções** o que permite limitar a propagação de erros e facilitar a manutenção
- ◆ Garantia, pela modularidade, de um grau satisfatório de **evolutividade** e de **extensibilidade** da rede
- ◆ **Otimização do desempenho.**

## Arquitetura Multicamadas

### ◆ Estruturação em Camadas



## Arquitetura Multicamadas

### ◆ Características da Arquitetura Multicamadas

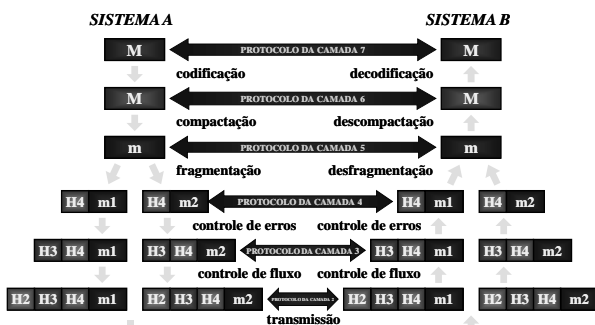
- ◆ O conhecimento que uma camada tem da camada imediatamente inferior é limitado ao serviço oferecido em sua interface
- ◆ As camadas comunicam-se entre si através de interfaces bem definidas
- ◆ Cada camada só se comunica num dado sistema com as camadas adjacentes (superior e inferior)

## Arquitetura Multicamadas

### ◆ Características da Arquitetura Multicamadas

- ◆ Cada camada executa um subconjunto das funções de comunicação atribuídas à arquitetura
- ◆ Para executar suas funções, uma dada camada faz uso dos serviços de comunicação providos pelas camadas que se localizam abaixo dela
- ◆ O serviço de comunicação provido por uma dada camada é utilizado pela camada imediatamente superior

## Arquitetura Multicamadas



## Interfaces e Serviços

### ◆ IMAGINE A REDE COMO UM CONJUNTO DE CAMADAS

### ◆ Solução do problema de comunicação

- ◆ Entidade são elementos ativos em cada camada;
- ◆ Entidade da mesma camada, mas localizadas em máquinas distintas são chamadas entidades pares;
- ◆ Uma entidade da camada n, disponibiliza um serviço à camada n+1, recorrendo, tipicamente a serviços da camada n-1;

## Interfaces e Serviços

- ◆ Conforme já mencionado, a função de cada camada é oferecer serviços para a camada acima dela.
- ◆ Os elementos ativos de uma camada, ou seja, os processos que a implementam são chamados entidades.
- ◆ **Interfaces:**
  - ◆ Conjunto de Primitivas / Funções de acesso a serviços oferecidos, responsável pela comunicação de cada entidade.
- ◆ **Serviços:**
  - ◆ O SERVIÇO corresponde a um conjunto de operações que uma camada é capaz de oferecer à camada imediatamente superior.
  - ◆ Ele define o que uma camada é capaz de executar sem se preocupar com a maneira pela qual as operações serão executadas.
  - ◆ O serviço está intimamente relacionado com as interfaces entre duas camadas, sendo a inferior a fornecedora do serviço e a superior a usuária deste.

## Primitivas de Serviços

- ◆ Implementação do **confiável** e **não-confiável**.
- ◆ Como vimos:
  - ◆ Um serviço dito **confiável** é aquele em que os dados não podem ser jamais perdidos, ou melhor, jamais podem deixar de realizar o serviço adequadamente e, por vezes, algum mecanismo de recuperação em caso de falha deve ser utilizado.
  - ◆ Serviços **não confiáveis** são aqueles onde, eventualmente, dados podem ser perdidos e não recuperados pela camada em questão.
- ◆ Normalmente, a implementação de serviços confiáveis é feita através da definição de mensagens de reconhecimento enviadas pelo receptor, para cada mensagem recebida do emissor.
- ◆ Este processo, embora extremamente benéfico, introduz uma lentidão na transferência de dados, o que significa que nem sempre ele é desejável num sistema.

## Primitivas de Serviços

- ◆ Um serviço é definido formalmente por um conjunto de **primitivas** (ou operações) disponíveis a um usuário ou a outras entidades para o acesso àquele serviço.
- ◆ Estas primitivas permitem **indicar a ação a ser executada pelo serviço** ou ainda um pedido de informação sobre uma ação executada previamente.
- ◆ As primitivas de serviço são divididas em quatro classes: **pedido** (*request*), **indicação** (*indication*), **resposta** (*response*) e **confirmação** (*confirm*).

PRIMITIVA	SIGNIFICADO
request	Pedido enviado por uma entidade que solicita um serviço
indication	Através dela, a entidade par é informada de uma solicitação de serviço
response	A entidade par responde ao pedido de serviço
confirm	A entidade solicitante é informada do resultado do serviço

## Primitivas de Serviços

- ◆ Uma analogia com o sistema telefônico pode ser de grande utilidade para que se entenda a utilização dessas primitivas.
- ◆ Nesse caso considere os procedimentos necessários para você telefonar à tia Nena e convidá-la para tomar um chá em sua casa:

AÇÃO	PRIMITIVA
01 Você disca o número do telefone da casa de sua tia.	CONNECT.request
02 O telefone dela toca.	CONNECT.indication
03 Ela tira o telefone do gancho.	CONNECT.response
04 Você ouve o "alô" dela.	CONNECT.confirm
05 Você a convida para o chá	DATA.request
06 Ela ouve o convite	DATA.indication
07 Ela diz que terá o maior prazer em ir a sua casa.	DATA.request
08 Você ouve a aceitação	DATA.indication
09 Você coloca o telefone no gancho	DISCONNECT.request
10 Ela ouve você colocar o telefone no gancho e faz o mesmo.	DISCONNECT.indication