
Sistemas Distribuídos

— Comunicação em SD —

Questões

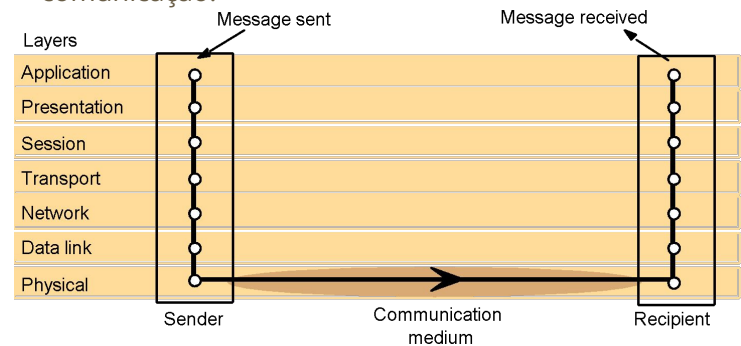
- Porque os tradicionais mecanismos de comunicação entre processos utilizados em sistemas centralizados não podem ser utilizados em sistemas distribuídos?
- Como é possível permitir a comunicação entre processos em um sistema distribuído?

Introdução: Comunicação em Sistemas Distribuídos

- A diferença mais importante entre Sistemas Distribuídos e Sistemas Centralizados é a Comunicação Interprocesso (Interprocess Communication – IPC);
- Sistemas Centralizados
 - memória compartilhada;
- Sistemas Distribuídos
 - não existe memória compartilhada;
 - Comunicação Interprocesso reformulada
 - Permitir que os processos se comuniquem para troca de dados ou acessos a recursos ou serviços em processadores remotos.

Protocolos: Camadas

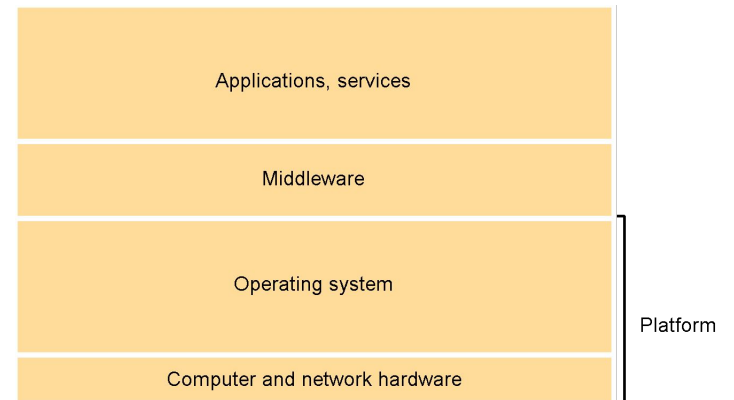
- Comunicação é dividida em sete níveis ou camadas, onde cada camada trata de um aspecto específico da comunicação.



Protocolos: Desafios

- Roteamento:
 - Prover o caminho mais eficiente para um pacote, através da aplicação de algoritmos de roteamento
- Controle de Congestionamento:
 - Evitar a degradação na vazão da rede através de atrasos no envio de pacotes;
 - Informar aos participantes da rota do pacote sobre o congestionamento.
- Internetworking:
 - Integrar diversos tipos de redes, endereçamento, protocolos, componentes de ligação (roteadores, bridges, hubs, switches).

Comunicação Interprocessos: Camadas de Serviços

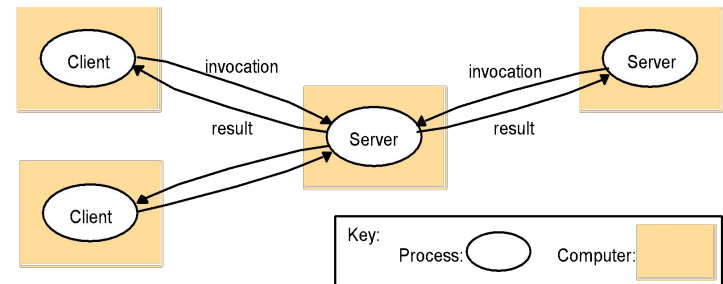


Comunicação Interprocessos: Modelo Cliente-Servidor

- A ideia é estruturar o sistema operacional como um grupo de processos cooperativos chamados:
 - Servidores: Oferecem serviços aos usuários;
 - Clientes: Usam os serviços provido pelos Servidores.
- Uma máquina pode executar:
 - Um único ou múltiplos processos clientes;
 - Um único ou múltiplos processos servidores;
 - Ou uma combinação das alternativas anteriores.

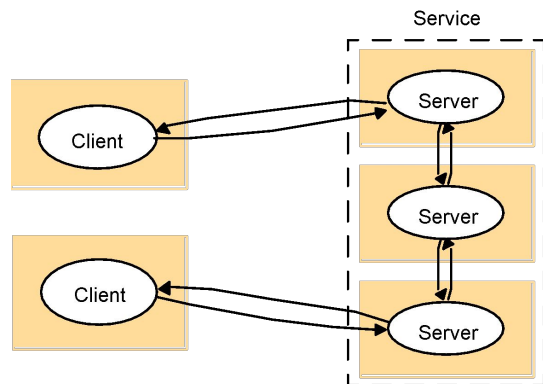
Comunicação Interprocessos: Modelo Cliente-Servidor

- Processos Clientes acionando individualmente Processos Servidores:



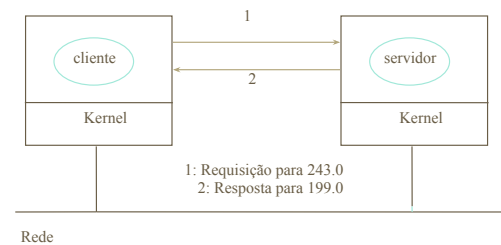
Comunicação Interprocessos: Modelo Cliente-Servidor

- Um serviço provido por Múltiplos Processos Servidores:



Troca de Mensagem: Endereçamento

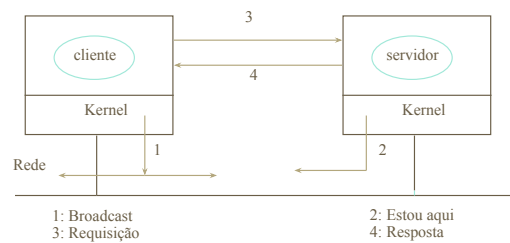
- Endereçamento Máquina Processo:



- Não é transparente
 - se um servidor não estiver disponível teremos recompilação para poder realizar o serviço em outro servidor.

Troca de Mensagem: Endereçamento

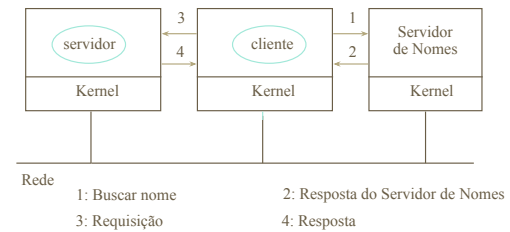
- Endereçamento Aleatório:



- Broadcast gera carga extra de comunicação no sistema.

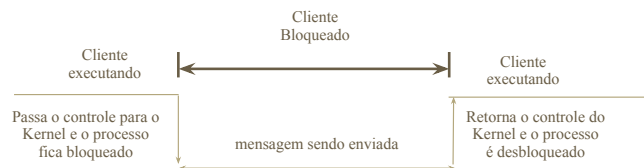
Troca de Mensagem: Endereçamento

- Endereçamento usando um Servidor de Nomes:



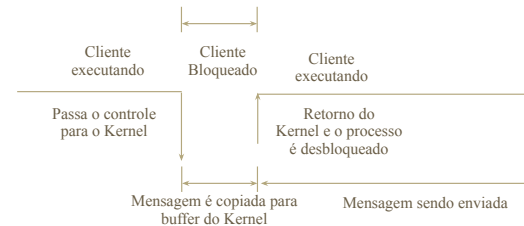
Troca de Mensagem: Primitiva Send

- Primitivas Bloqueadas (Síncronas):
 - Primitivas vistas até agora (send e receive) são chamadas primitivas bloqueadas.
 - Enquanto a mensagem está sendo enviada ou recebida, o processo permanece bloqueado (suspensão).



Troca de Mensagem: Primitiva Send

- Primitivas Não Bloqueadas (Assíncronas):
 - Quando um send é executado o controle retorna ao processo antes da mensagem ser enviada;
 - O processo que executa o send pode continuar processando enquanto a mensagem está sendo enviada.



Questões

- Como implementar no cliente mecanismos de tolerância a falhas relacionados às chamadas enviadas aos servidores?

Troca de Mensagem: Confiabilidade das Primitivas

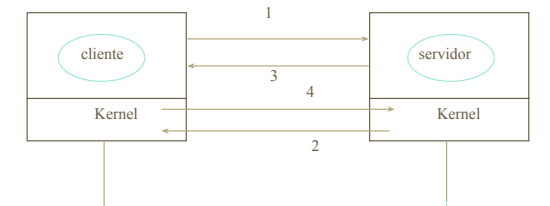
- 1 Redefinir a semântica do send para ser não confiável:
 - O sistema não dá garantias sobre uma mensagem que está sendo enviada
 - Tornar a comunicação confiável uma tarefa do usuário (complicado!!).
- 2 Requerer que o Kernel do receptor envie uma mensagem acknowledgment (ack - confirmação) para o Kernel do transmissor
 - O Kernel só libera o cliente quando o ack for recebido;
 - O ack é uma operação realizada pelos dois Kernels, sem o conhecimento do cliente e servidor

Troca de Mensagem: Confiabilidade das Primitivas

- 3 Aproveitar o fato que a comunicação cliente-servidor é estruturada como uma requisição do cliente para o servidor, seguido de uma resposta do servidor para o cliente:
 - O cliente é bloqueado depois do envio da mensagem;
 - O Kernel do servidor não envia um ack, em vez disto a resposta serve de ack;
 - Desta forma o processo de envio da mensagem permanece bloqueado até a resposta chegar;
 - Se isto demorar muito o Kernel pode reenviar a requisição, protegendo-se contra a perda de mensagens.

Troca de Mensagem: Confiabilidade das Primitivas

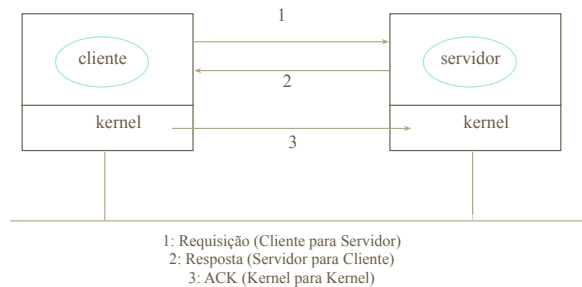
Mensagens de ACK individual



- 1: Requisição (Cliente para Servidor)
2: ACK (Kernel para Kernel)
3: Resposta (Servidor para Cliente)
4: ACK (Kernel para Kernel)

Troca de Mensagem: Confiabilidade das Primitivas

Resposta sendo usada como ACK



Troca de Mensagem: Implementação

- Detalhes de como a passagem de mensagem é implementada depende das escolhas feitas durante o projeto;
- Algumas considerações:
 - Há um tamanho máximo do pacote transmitido pela rede de comunicação;
 - Mensagens maiores precisam ser divididas em múltiplos pacotes que são enviados separadamente;
 - Alguns dos pacotes podem ser perdidos ou chegar na ordem errada;
 - Solução: Atribuir a cada mensagem o número da mensagem e um número de seqüência.

Troca de Mensagem: Implementação

- O acknowledgment pode ser para cada pacote individual ou para a mensagem como um todo;
 - No primeiro caso na perda de mensagem, somente um pacote precisa ser re-transmitido, mas na situação normal requer mais pacotes na rede de comunicação;
 - No segundo caso há a vantagem de menos pacotes na rede mas a desvantagem da recuperação no caso de perda de mensagem é mais complicada;
- Conclusão: A escolha de um dos dois métodos depende da taxa de perdas na rede.

Protocolo Cliente/Servidor

Código	Tipo	De	Para	Significado
REQ	Requisição	Cliente	Servidor	Solicitação de Serviço
REP	Resposta	Servidor	Cliente	Resposta à solicitação
ACK	Confirmação	Cliente Servidor	Servidor Cliente	A última mensagem chegou
AYA	Are You Alive?	Cliente	Servidor	Testa se o servidor está ativo
IAA	I Am Alive	Servidor	Cliente	Servidor ativo
TA	Try Again	Servidor	Cliente	Não posso atender
AU	Address Unknown	Servidor	Cliente	Não há processos com este endereço

Algumas nomenclaturas

Cliente / Servidor

- Cliente
 - Aplicação que tem por finalidade enviar requisições a um servidor (*software*) em uma máquina remota
- Servidor
 - Aplicação dedicada em receber as requisições dos clientes, processá-las e enviar as respostas

Algumas Terminologias

- Host
 - Computador ou máquina conectado à Web
- Conexão
 - Canal de comunicação entre dois hosts
- Pacote
 - Unidade básica de comunicação na rede

Algumas Terminologias

- IP
 - *Internet Protocol*, protocolo que coordena a remessa de pacotes entre os hosts
- Endereço IP
 - Endereço numérico de 32 bits (IP v4) representando um host na Internet

Endereçamento

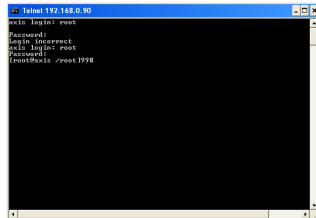


Algumas Terminologias

- DNS
 - Domain Name System – serviço responsável por traduzir nomes de hosts em endereços IP
- TCP
 - *Transmission Control Protocol* – protocolo que estabelece um canal de comunicação confiável e bidirecional

Serviços Existentes na Internet

- HTTP – *Hyper Text Transfer Protocol*
 - Protocolo de transferência de HiperTextos
 - HTTPS – HTTP sobre SSL - *Secure Socket Layer*
- FTP – *File Transfer Protocol*
 - Protocolo de transferência de arquivos
- Telnet
 - Emula um terminal remotamente
 - Conexão com um servidor via prompt de comando
- SSH
 - Emula um terminal com conexão segura
 - Utiliza o protocolo SSL



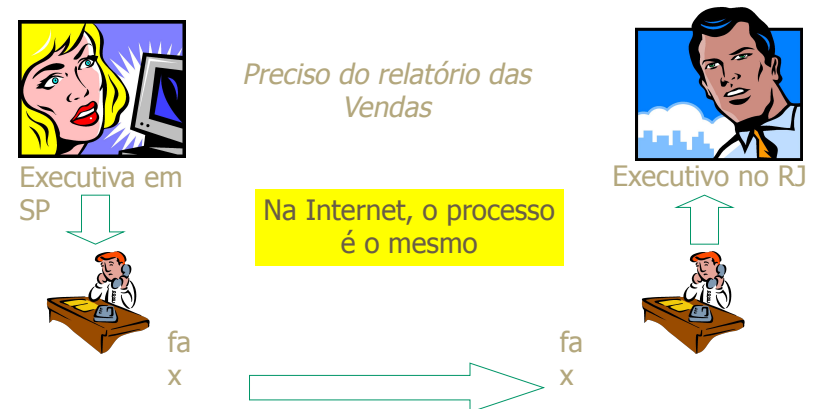
Prof. Emerson Paduan: emerson@paduan.pro.br

Protocolos

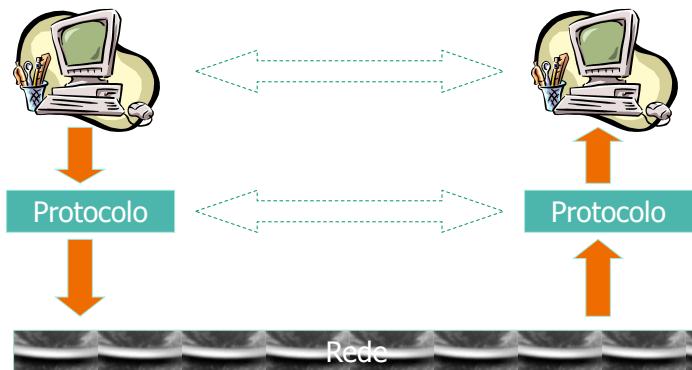
Introdução

- O que é um protocolo de comunicação?
 - Conjunto de regras e procedimentos para que duas entidades distintas possam estabelecer um canal de comunicação
 - Etapas de estabelecimento e finalização de conexão
 - Cabeçalhos indicativos de numeração e ordem dos pacotes
 - Endereço do emissor e receptor

Introdução

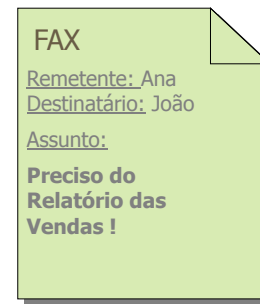


Protocolos na Internet



Prof. Emerson Paduan: emerson@paduan.pro.br

Comparação



Vida Real

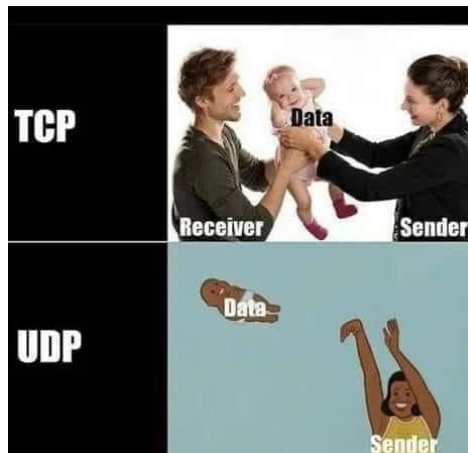


Source	200.18.92.1
Destination	187.231.97.2
Flags	1 0 1 1 0 1
Data	"Preciso do Relatório das Vendas"

Pacote de Dados

Prof. Emerson Paduan: emerson@paduan.pro.br

TCP x UDP

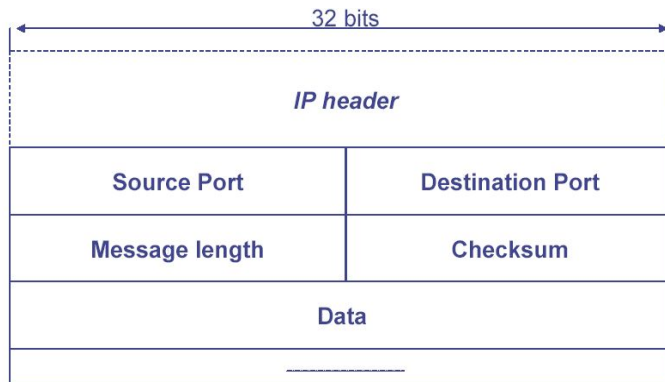


Protocolo UDP

- **User Datagram Protocol (UDP):**
- Protocolo não orientado à conexão;
- Não há garantia de entrega dos dados (não há mensagens de confirmação);
- Perdas durante as transmissões não são tratadas por este protocolo;
- Usado em redes com alta confiabilidade, onde as taxas de perda são baixas;

Protocolo UDP

- **Header UDP:**

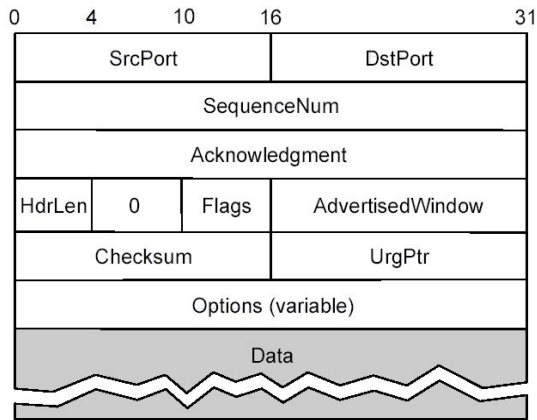


Protocolo TCP/IP

- Protocolo atualmente utilizado na Internet.
- Baseado no modelo OSI da ISO (7 camadas)
- TCP – protocolo de transporte que utiliza os serviços do protocolo IP para garantir estabelecimento de conexões e integridade de dados
- IP – protocolo de rede responsável pelo endereçamento das máquinas (endereço IP) e rotas entre dispositivos.

Protocolo TCP

- **Header TCP:**



Protocolos TCP

- **Transmission Control Protocol (TCP):**
- Protocolo orientado à conexão:
 - Exige o estabelecimento de um canal lógico para iniciar a transmissão de dados, em 3 fases:
 - Fase de conexão
 - Fase de dados
 - Fase de desconexão
- Exemplos de aplicação:
 - TELNET, Web Browser, ...

Etapas em uma Conexão TCP

