

Introdução a camada de aplicação

Aula 8

Prof. Me. Márcio Moraes Lopes

marcio.comp@gmail.com
Universidade Federal de Jataí



Jataí, 27 abril de 2018

- Data da última atualização: **25 de abril de 2018**
- Estes *slides* são incompletos e podem conter erros de digitação, inconsistências, imperfeições e outros tipos defeitos
- Ao constatarem algum erro, por gentileza, reportar ao autor
- Alguns termos ou slides completos podem estar em inglês
- Estes *slides* **NÃO** devem ser utilizados como **ÚNICA** fonte de estudos

Referências

- FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. 4. ed. São Paulo: McGraw-Hill, 2008.
- KUROSE, J.; ROSS, K. **Redes de computadores e a internet – uma abordagem top-down**. 5. ed. São Paulo: Pearson Addison-Wesley, 2013.
- TANENBAUM, A. S.; WETHERALL, D. **Redes de computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.
- FARREL, Adrian. **A Internet e seus protocolos: Uma análise comparativa**. Rio de Janeiro: Elsevier, 2005.
- COMER, D. E. **Redes de computadores e internet**. 6ª ed. Porto Alegre: Bookman, 2016.

Agenda

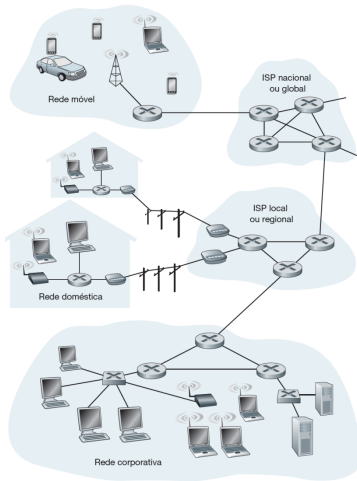
1 Alguns aspectos interessantes

- Estrutura da Internet
- Interação entre sistemas finais
- Redes de acesso
- Núcleo da rede

2 Camada de Aplicação

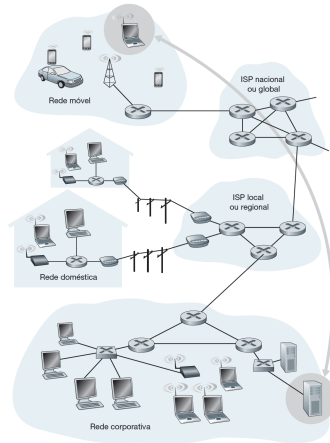
- O que é uma aplicação?
 - Desenvolvimentos de aplicações
- Arquitetura de uma comunicação
 - Arquitetura cliente/servidor
 - Arquitetura peer-to-peer - P2P
 - Portas
 - Escolha de um transporte
- Comunicação entre processos
 - Sockets

Estrutura da Internet



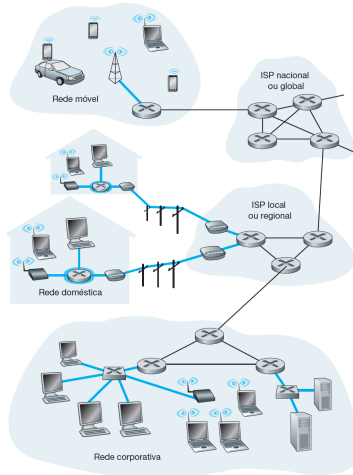
Fonte: *Kurose, 2013.*

Interação entre sistemas finais



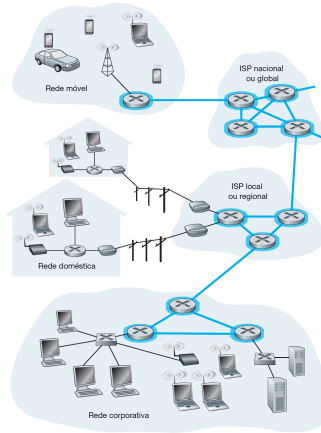
Fonte: Kurose, 2013.

Redes de acesso



Fonte: Kurose, 2013.

Núcleo da rede



Fonte: *Kurose, 2013.*

O que é uma aplicação?

- “As aplicações são programas executados para usuários ou em nome deles a fim de tratar de requisições de serviços específicos feitas por estes usuários.” **Farrel, 2005.**
- “Aplicações de rede são a razão de ser de uma rede de computadores.” **Kurose, 2013.**
 - “Se não fosse possível inventar aplicações úteis, **não haveria necessidade** de projetar protocolos de rede para suportá-las.”
- “A *camada* de aplicação permite ao usuário, **seja ele humano ou software**, acessar a rede.” **Forouzam, 2008.**
 - “Fornece interfaces com o usuário e suporte para serviços (...).”
 - “A camada de aplicação é responsável por fornecer serviços ao usuário final.”
- “A *camada* de aplicação contém uma série de protocolos comumente necessários para os usuários.” **Tanenbaum, 2011.**

Núcleo do desenvolvimento de uma aplicação

- Programas que rodem em sistemas finais **diferentes** e se **comuniquem entre si**
- Normalmente são compostas por duas partes
 - Um *cliente* que recebe instruções do usuário
 - Um *servidor* remoto que recebe as requisições enviadas pelo cliente. Uso de um *daemon* (processo em segundo plano)
 - Exemplo
 - O *browser* (Firefox) que está em execução no *host* do usuário
 - O servidor Web (Apache) que está em execução no próprio *host* servidor
- Podem ter outros modelos, como por exemplo
 - Peer-to-peer- P2P
- Não a necessidade de escrever programas para o núcleo da rede

Desenvolvimento de aplicações

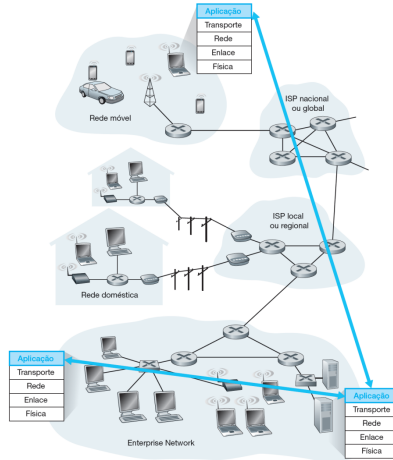
Protocolos

- Sintaxe e semântica da mensagem que pode ser trocada
- O indicador de interação (quem é o cliente ou o servidor)
- As ações a serem realizadas em caso de erro
- Como os dois lados sabem quando a comunicação termina

Tipos de serviço

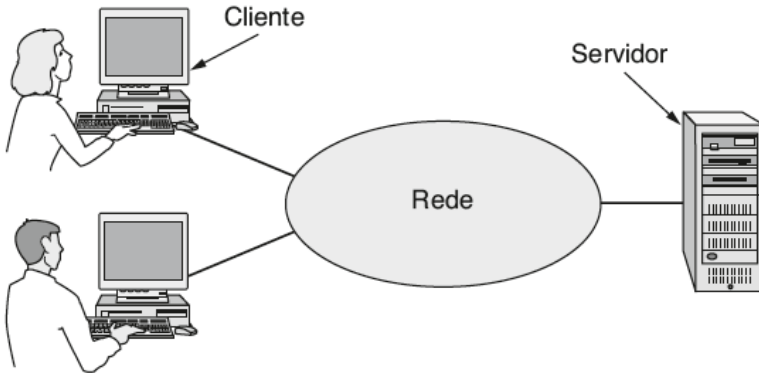
- Privado
- Público

Arquitetura da comunicação entre sistemas finais



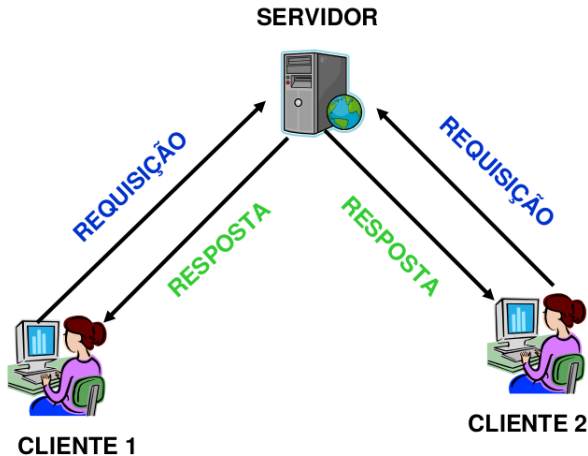
Fonte: Kurose, 2013.

Arquitetura cliente/servidor

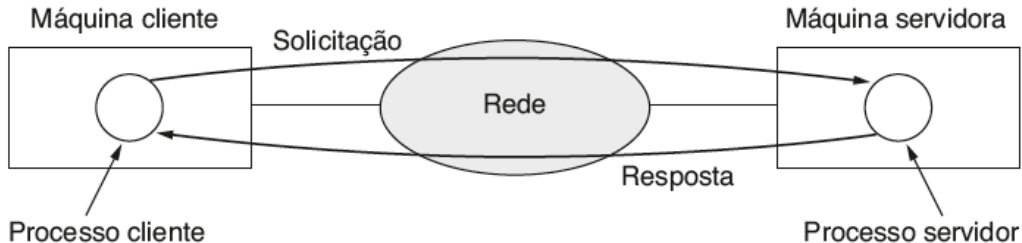


Fonte: *Tanenbaum, 2013.*

Arquitetura cliente/servidor



Arquitetura cliente/servidor



Fonte: *Tanenbaum, 2013.*

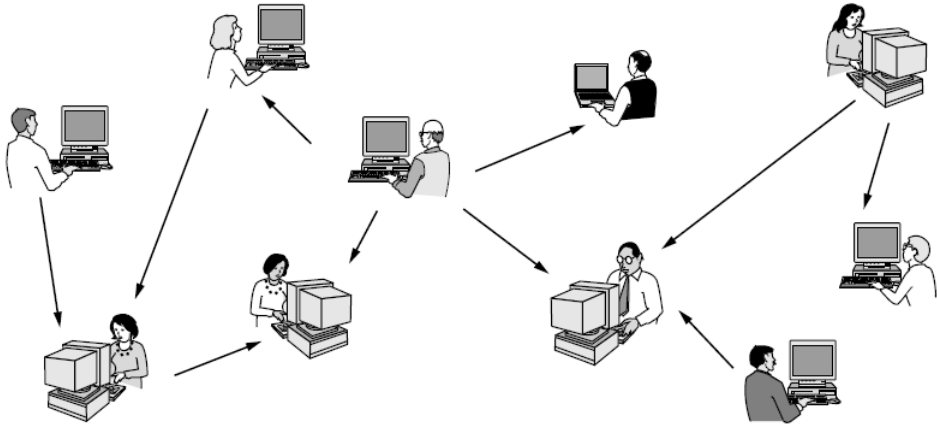
Interativos

- Passos de uma negociação
 - 1 Espera uma requisição de um cliente
 - 2 Processa a requisição do cliente
 - 3 Envia a resposta para o cliente que enviou a requisição
- Vantagem: Mais simples
- Desvantagem: Processa somente uma requisição por vez

Concorrentes

- Passos de uma negociação
 - 1 Espera uma requisição de um cliente
 - 2 É iniciado um novo processo ou *thread* para gerenciar a requisição do cliente
 - 2.1 No final do processamento de uma requisição o servidor destrói os processos/threads criados
- Vantagens: Podem gerenciar várias conexões
- Desvantagem: Maior complexidade

Arquitetura peer-to-peer - P2P



Fonte: Tanenbaum, 2013.

Função

- Prover um meio prático de identificar os fluxos de comunicação entre um cliente e um servidor
- Vários clientes e vários servidores podem estar em execução em um único host, o endereço IP não é suficiente para distinguir as conversações
 - O tráfego é rotulado com a porta de origem e de destino

Geração de portas

- **No servidor**
 - Uma aplicação servidora escuta as requisições em uma porta bem-conhecida (*well-know port*)
 - Todos os clientes saibam como se conectar a esta aplicação
- **No cliente**
 - Seleciona sua própria porta de origem

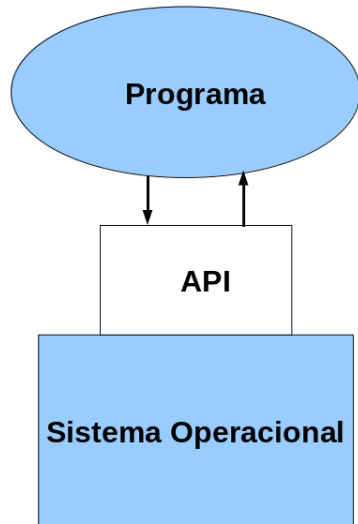
Escolha de um transporte

Troca de dados

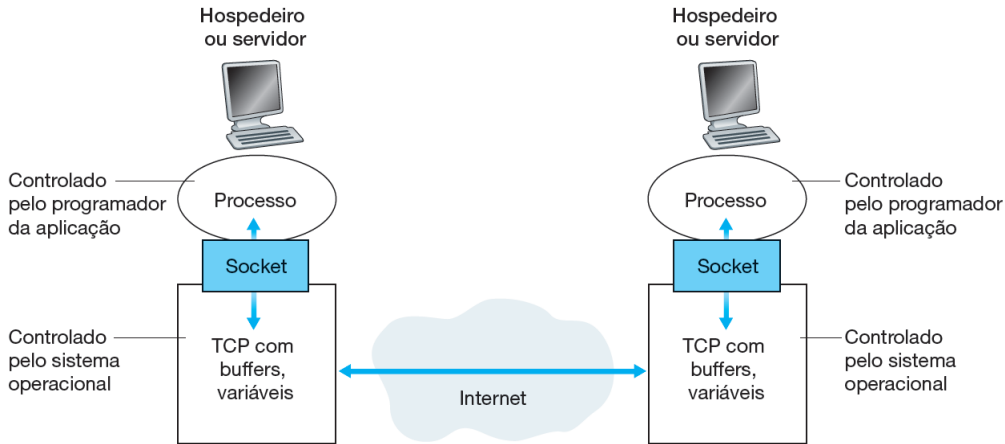
- As aplicações precisam escolher um meio de comunicarem os dados entre o cliente e servidor
- Na Internet isso envolve escolher um protocolo de transporte IP ou simplesmente enviar os dados em IP “bruto”
- Projetista/implementadores de aplicações desejam estar protegidos alguns problemas como
 - Entrega fora de ordem
 - Corrupção dos dados
 - Perda de mensagens
- Sem necessidade de “reinventar a roda”, uma vez que pode ser utilizado as vantagens do serviço de transporte fornecidos pelos protocolos já existentes
 - É raro um protocolo de aplicação operar usando pacotes IP diretamente

API - Application Programming Interface

- Interface de Programação da Aplicação
- Uma camada de software (middleware) entre a camada de aplicação e a camada de transporte
- Permitem que aplicações troquem dados



Sockets



Fonte: Kurose, 2013.