

Redes de computadores

Camada de aplicação

Gabriel V C Candido
gabriel.candido@ifpr.edu.br

Instituto Federal do Paraná - Pinhais

Sumário

Camada de aplicação

Arquiteturas de aplicação

Comunicação de processos

Protocolos da camada de aplicação

Sumário

Camada de aplicação

Arquiteturas de aplicação

Comunicação de processos

Protocolos da camada de aplicação

Camada de aplicação

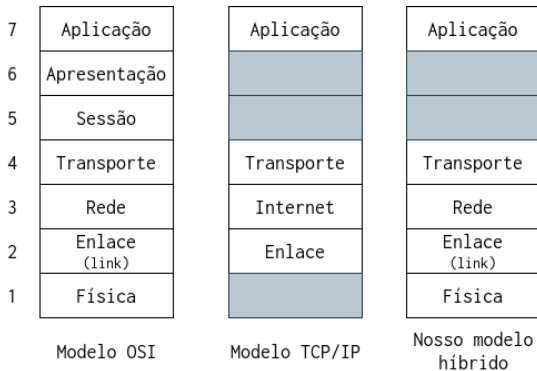
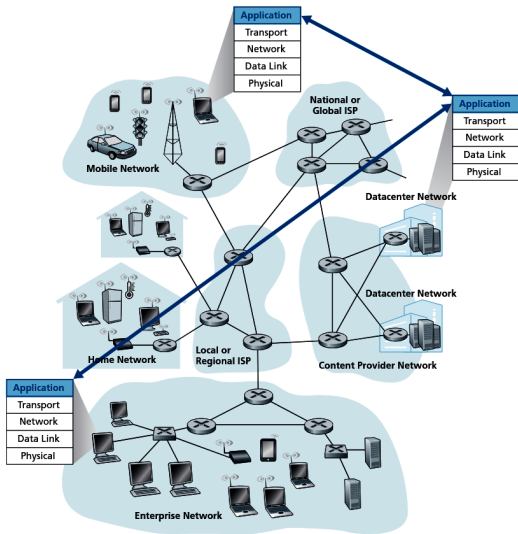


Figura: Os modelos de camadas que estudamos.

Camada de aplicação

- ▶ É a *razão de ser* das redes de computadores;
- ▶ Existem *diversas* aplicações;
 - ▶ Aplicações de texto: e-mail, acesso remoto, notícias, transferência de arquivos, ...
 - ▶ *World Wide Web*: páginas, pesquisa, comércio, ...
 - ▶ Vídeoconferência: Skype, Meet, Teams, ...
 - ▶ YouTube, Netflix, Spotify, ...
 - ▶ Redes sociais;
 - ▶ Jogos;
 - ▶ Mapas e rotas, pagamentos, mensagens, ...
- ▶ Vamos estudar algumas poucas aplicações.

Camada de aplicação



Camada de aplicação

- ▶ Escrever programas que executam em sistemas (*hosts*) diferentes e se comunicam pela rede;
- ▶ Independente de linguagem de programação;
- ▶ Não precisamos escrever software para os dispositivos do meio da rede;

Sumário

Camada de aplicação

Arquiteturas de aplicação

Comunicação de processos

Protocolos da camada de aplicação

Arquiteturas de aplicação

- ▶ Cliente-servidor:
 - ▶ Servidor possui o conteúdo ou vai mediar as trocas de informações;
 - ▶ Clientes não se comunicam diretamente!;
 - ▶ Servidor possui um endereço IP fixo e conhecido;
 - ▶ Podemos combinar servidores para atender à demanda;
 - ▶ Exemplos: Web, telnet, e-mail, FTP.

Arquiteturas de aplicação

- ▶ *Peer-to-peer* (P2P):
 - ▶ *Hosts* se comunicam diretamente;
 - ▶ Arquitetura descentralizada;
 - ▶ Não depende de um servidor dedicado, ou depende *muito* pouco;
 - ▶ Escalabilidade;
 - ▶ Desafios de segurança, desempenho e disponibilidade;
 - ▶ Exemplo: torrents.

Sumário

Camada de aplicação

Arquiteturas de aplicação

Comunicação de processos

Protocolos da camada de aplicação

Processos que se comunicam

- ▶ Programas executando (processos) em *hosts* diferentes trocam mensagens;
- ▶ Teremos sempre um par de processos comunicando;
- ▶ Cliente inicia o contato com o servidor;



Como enviar mensagens?

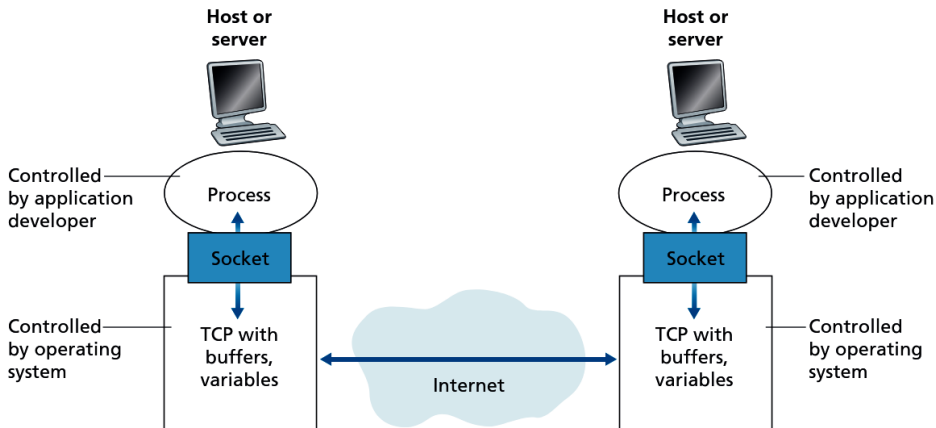


Figura: A interface de *sockets*. Fonte: KR 2.

Como enviar mensagens?

- ▶ *Sockets* são a interface entre a aplicação e a camada de transporte;
- ▶ O programador da aplicação não controla as camadas de baixo;
- ▶ Através dos *sockets* ele consegue apenas escolher qual protocolo de transporte utilizar e alguns parâmetros.

Endereços e portas

- ▶ Todo *host* que está se comunicando precisa ter um *endereço IP*;
- ▶ Além disso, precisamos das *portas* para saber qual processo está se comunicando.
 - ▶ Se um computador está executando 3 aplicações, quem está se comunicando?
 - ▶ Jogo;
 - ▶ Spotify;
 - ▶ Discord;



Sumário

Camada de aplicação

Arquiteturas de aplicação

Comunicação de processos

Protocolos da camada de aplicação



Protocolos da camada de aplicação

Um protocolo de aplicação define:

- ▶ Os tipos de mensagens trocadas (por exemplo, requisições e respostas);
- ▶ O formato das mensagens: campos e tamanhos;
- ▶ O que cada campo significa;
- ▶ Regras de quando e como um processo pode enviar cada mensagem, e como um processo pode responder às mensagens.

Existem protocolos públicos (RFCs) e protocolos proprietários.

Protocolo de aplicação vs Aplicação

- ▶ Uma aplicação Web é uma aplicação cliente-servidor para um usuário obter documentos da Web;
- ▶ Uma aplicação Web possui vários componentes:
 - ▶ Um formato para os documentos (p. ex.: HTML);
 - ▶ Navegadores (p. ex.: Chrome, Firefox);
 - ▶ Servidores Web (p. ex.: Apache, nginx);
 - ▶ Protocolo de aplicação (HTTP).



Protocolo de aplicação vs Aplicação

- ▶ Netflix possui vários componentes:
 - ▶ Clientes (p. ex.: aplicativo Android);
 - ▶ Servidores que armazenam e transmitem os vídeos;
 - ▶ Servidores que gerenciam contas e pagamentos;
 - ▶ Protocolo de aplicação (DASH).



Quais protocolos de aplicação estudaremos?

- ▶ HTTP, HTTPS;
- ▶ E-mail (vários protocolos);
- ▶ DNS.

