
MAC0352 - Redes de Computadores e Sistemas Distribuídos

Daniel Macêdo Batista

IME - USP, 27 de Abril de 2021

- [Camada de aplicação](#)
- [Visão geral de sockets](#)
- [Softwares](#)

Camada de aplicação

Visão geral de sockets

Softwares

▷ Camada de aplicação

Visão geral de sockets

Softwares

Camada de aplicação

2

Parte 2: Camada de aplicação

Nossos objetivos:

- Conceitual, aspectos de implementação de protocolos de aplicação de redes
 - Modelos de serviço da camada de transporte
 - Paradigma cliente-servidor
 - Paradigma **peer-to-peer**
 - Aprender sobre protocolos examinando protocolos da camada de aplicação populares:
 - HTTP
 - FTP
 - SMTP/ POP3/ IMAP
 - DNS
- Programação de aplicações de rede
 - Socket API

2 Algumas aplicações de rede

- E-mail
- Web
- Mensagem instantânea
- Login remoto
- P2P file sharing
- Jogos de rede multi-usuário
- Streaming stored videoclipes
- Telefonia via Internet
- Videoconferência em tempo real
- Computação paralela massiva

2

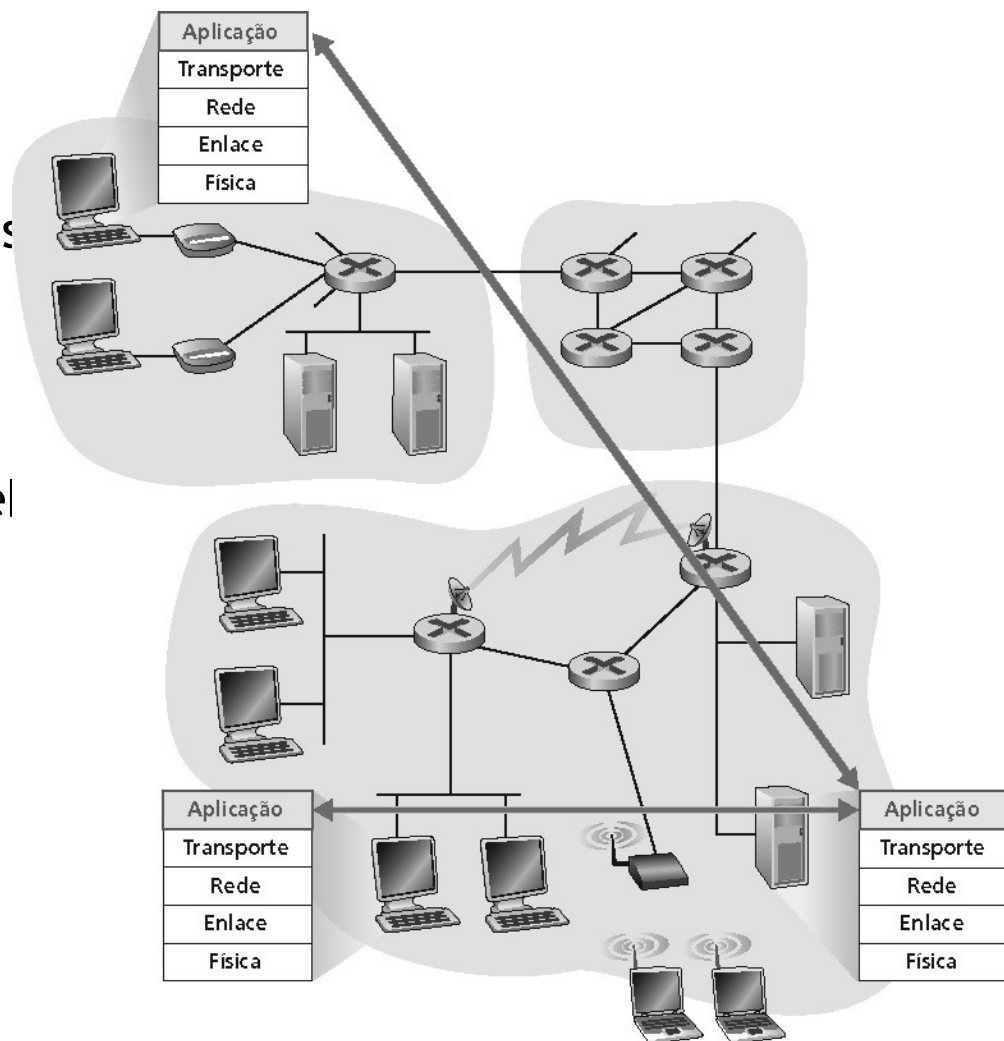
Criando uma nova aplicação de rede

Escrever programas que

- Executem sobre diferentes sistemas finais e
- Se comuniquem através de uma rede.
- Ex.: Web - software de servidor Web se comunicando com software do browser.

Nenhum software é escrito para dispositivos no núcleo da rede

- Dispositivos do núcleo da rede não trabalham na camada de aplicação
- Esta estrutura permite um rápido desenvolvimento de aplicação



2 Arquiteturas de aplicação

- Cliente-servidor
- **Peer-to-peer** (P2P)
- Híbrida de cliente-servidor e P2P

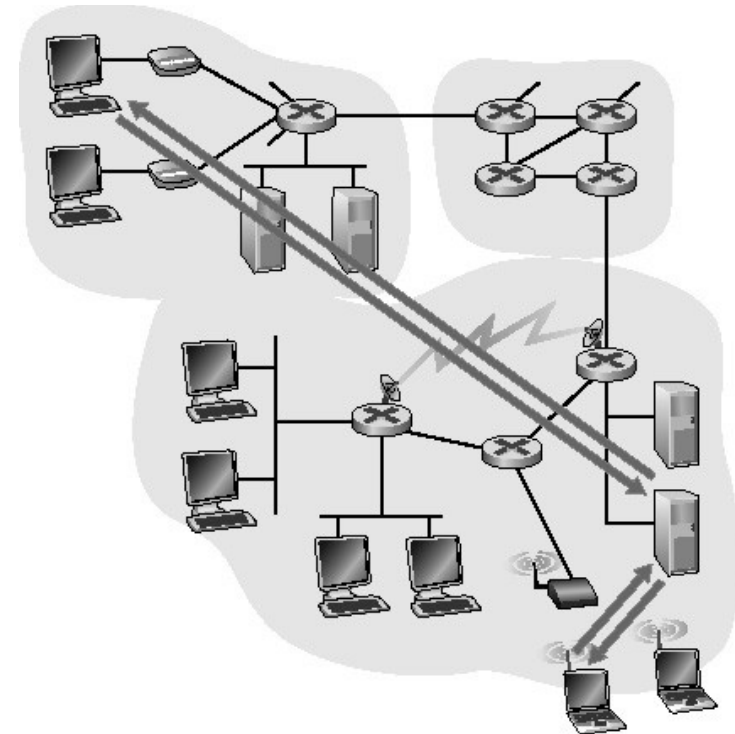
2 Arquitetura cliente-servidor

Servidor:

- Hospedeiro sempre ativo
- Endereço IP permanente
- Fornece serviços solicitados pelo cliente

Clientes:

- Comunicam-se com o servidor
- Pode ser conectado intermitentemente
- Pode ter endereço IP dinâmico
- Não se comunicam diretamente uns com os outros



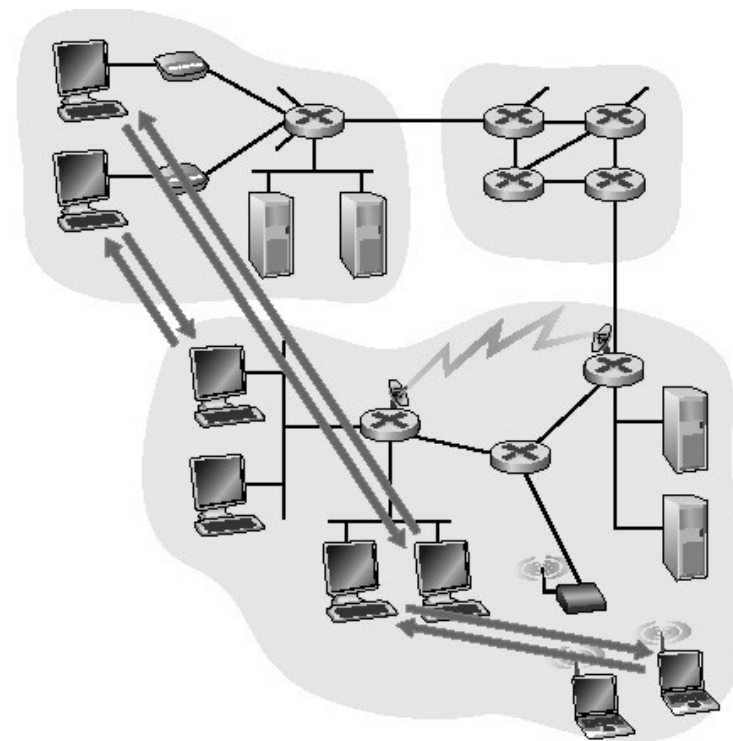
a. Aplicação cliente-servidor

2

Arquitetura P2P pura

- Nem sempre no servidor
- Sistemas finais arbitrários comunicam-se diretamente
- Pares são intermitentemente conectados e trocam endereços IP
- Ex.: Gnutella

Altamente escaláveis mas difíceis de gerenciar



b. Aplicação P2P



2

Híbrida de cliente-servidor e P2P

Napster

- Transferência de arquivo P2P
- Busca centralizada de arquivos:
 - Conteúdo de registro dos pares no servidor central
 - Consulta de pares no mesmo servidor central para localizar o conteúdo

Instant messaging

- Bate-papo entre dois usuários é P2P
- Detecção/localização centralizada de presença:
 - Usuário registra seu endereço IP com o servidor central quando fica on-line
 - Usuário contata o servidor central para encontrar endereços IP dos vizinhos

2 Comunicação de processos

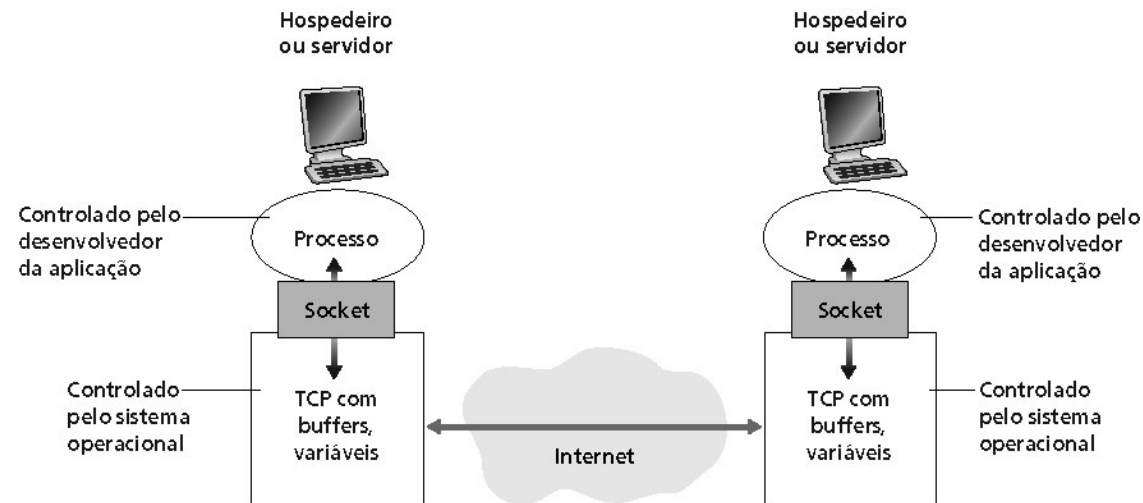
Processo: programa executando num hospedeiro

- Dentro do mesmo hospedeiro: dois processos se comunicam usando **comunicação interprocesso** (definido pelo OS)
- Processos em diferentes hospedeiros se comunicam por meio de troca de **mensagens**
- **Processo cliente:** processo que inicia a comunicação
- **Processo servidor:** processo que espera para ser contatado

Nota: aplicações com arquiteturas P2P possuem processos cliente e processos servidor

2 Sockets

- Um processo envia/recebe mensagens para/de seu **socket**
- O socket é análogo a uma porta
 - O processo de envio empurra a mensagem para fora da porta
 - O processo de envio confia na infra-estrutura de transporte no outro lado da porta que leva a mensagem para o socket no processo de recepção.



- API: (1) escolha do protocolo de transporte; (2) habilidade para fixar poucos parâmetros (**será explicado mais tarde**)

2 Processos de endereçamento

- Para um processo receber mensagens, ele deve ter um identificador
- Um hospedeiro possui um único endereço IP de 32 bits
- **P.:** O endereço IP do hospedeiro onde o processo está executando é suficiente para identificar o processo?
- **R.:** Não, muitos processos podem estar em execução no mesmo hospedeiro.
- O identificador inclui o endereço IP e o **número da porta** associada ao processo no hospedeiro
- Exemplos de números de porta:
 - Servidor HTTP: 80
 - Servidor de Correio: 25
- (mais detalhes serão mostrados mais tarde)

2

O protocolo da camada de aplicação define

- Tipo das mensagens trocadas, mensagens de requisição e resposta
- Sintaxe dos tipos de mensagem: os campos nas mensagens e como são delineados
- Semântica dos campos, ou seja, significado da informação nos campos
- Regras para quando e como os processos enviam e respondem às mensagens

Protocolos de domínio público:

- Definidos nas RFCs
- Recomendados para interoperabilidade
- Ex.: HTTP, SMTP

Protocolos proprietários:

- Ex.: KaZaA

Camada de aplicação

Camada de aplicação

Visão geral de sockets

Softwares

- ☐ Não exige mudança no núcleo da rede (núcleo simples, periferia complexa)
- ☐ Cliente-servidor, P2P, híbrida
- ☐ O protocolo especifica como se dá a troca de mensagens (Sockets, portas)
- ☐ Utiliza os serviços da camada de transporte (TCP, UDP)

Camada de aplicação

▷ Visão geral de
sockets

Softwares

Visão geral de sockets

Sockets

Camada de aplicação

Visão geral de sockets

Softwares

- ☐ Permitem a escrita de programas que se comunicam via rede
- ☐ Utilizam serviços da camada de transporte
- ☐ Necessário mais algumas informações das camadas inferiores

Códigos

Camada de aplicação

Visão geral de sockets

Softwares

- ☐ Cliente daytime
- ☐ Servidor daytime

Camada de aplicação

Visão geral de sockets

▷ Softwares

Softwares

- ☐ <http://nmap.org/>
- ☐ Lista as portas TCP e UDP abertas em computadores
- ☐ Na execução padrão ele busca as portas mais conhecidas para não demorar demais. A busca por todas as portas (de 0 até 65535) consome muito tempo mesmo se executada em uma rede local
- ☐ Se ele diz que o serviço é por exemplo SSH, ele sabe disso porque ele entende como um servidor SSH se apresenta. Ele não olha simplesmente a porta no arquivo `/etc/services`

Camada de aplicação

Visão geral de sockets

Softwares

- ☐ Lista o estado de todos os sockets do computador onde o comando é executado
- ☐ Se você escreve um programa servidor usando a API de sockets e executa ele, mesmo sem ter nenhum cliente conectado, a porta utilizada pelo servidor tem que aparecer na saída do netstat

Camada de aplicação

Visão geral de sockets

Softwares

- ☐ Lista todos os descritores abertos na máquina, inclusive sockets
- ☐ Permite descobrir qual o processo que está associado a cada porta apresentada na saída do netstat
- ☐ A depender dos parâmetros passados na linha de comando, ao invés do número da porta mostra o nome do protocolo (http, ftp, etc...). Mas ele não tem certeza que o protocolo é aquele, como o nmap faz, pois ele só olha o /etc/services