

# Capítulo 2: Camada de Aplicação

## Metas do capítulo:

- ❑ aspectos conceituais e de implementação de protocolos de aplicação em redes
  - modelos de serviço da camada de transporte
  - paradigma cliente servidor
  - paradigma *peer-to-peer*
- ❑ aprenda sobre protocolos através do estudo de protocolos populares da camada de aplicação:
  - HTTP
  - FTP
  - SMTP/ POP3/ IMAP
  - DNS

# Aula 6!

## Camada de aplicação

- ❑ Princípios
- ❑ Arquiteturas
- ❑ Comunicação entre processos
- ❑ Serviços de transporte disponíveis as aplicações
- ❑ exercício
- ❑ Protocolos da camada de aplicação

# Algumas aplicações de rede

- ❑ E-mail
- ❑ Web
- ❑ Instant messaging
- ❑ Login remoto
- ❑ Compartilhamento de arquivos P2P
- ❑ Jogos de rede multi-usuários
- ❑ Vídeo-clipes armazenados
- ❑ Voz sobre IP
- ❑ Vídeo conferência em tempo real

# Princípios de Aplicações de Rede

O núcleo do desenvolvimento de aplicação de rede é escrever programas que rodem em sistemas finais diferentes e se comuniquem entre si.

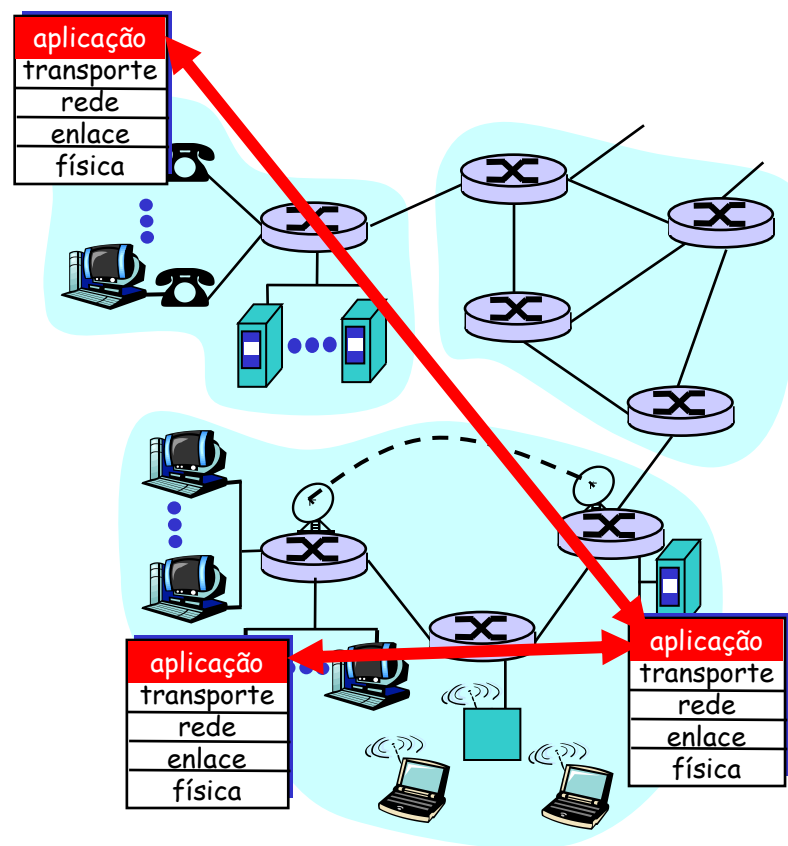
# Criando uma aplicação de rede

## Programas que

- Executam em diferentes sistemas finais
- Comunicam-se através da rede
- p.ex., Web: servidor Web se comunica com o navegador

## Programas não relacionados ao núcleo da rede

- Dispositivos do núcleo da rede não executam aplicações de usuários
- Aplicações nos sistemas finais permite rápido desenvolvimento e disseminação



# Arquiteturas de aplicação de rede

- ❑ Diferente da arquitetura de rede!
- ❑ A arquitetura de rede é fixa e provê um conjunto específico de serviços
  - Exemplo : 5 camadas da Internet - aplicação, transporte, Rede, enlace e física
- ❑ A arquitetura da aplicação é projetada pelo programador e determina como a aplicação é organizada nos vários sistemas finais.
  - Exemplo: cliente-servidor; P2P

# Arquiteturas das aplicações

- ❑ Cliente-servidor
- ❑ Peer-to-peer (P2P)

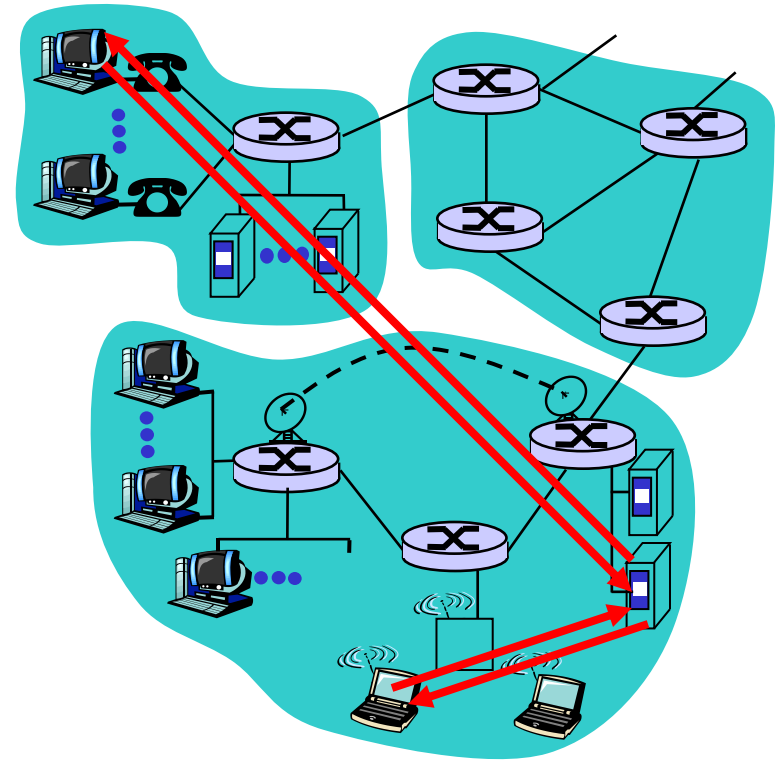
# Arquitetura cliente-servidor

## Servidor:

- ❑ Sempre ligado
- ❑ Endereço IP permanente
- ❑ Escalabilidade com *server farms* (conj, hospedeiros)

## Cliente:

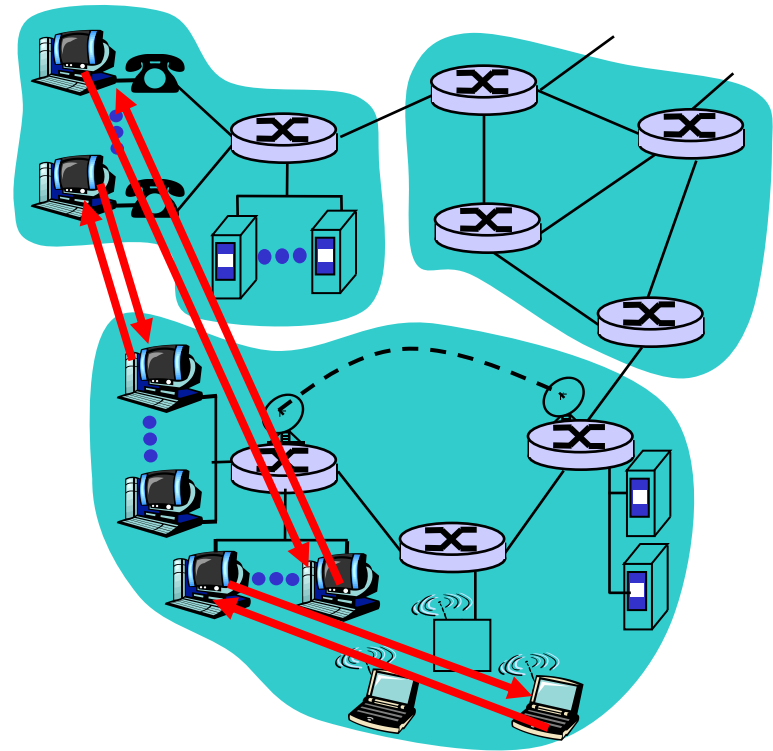
- ❑ Comunica-se com o servidor
- ❑ Pode estar conectado intermitentemente
- ❑ Pode ter endereços IP dinâmicos
- ❑ Não se comunica diretamente com outros clientes
- ❑ Exemplos: Web; FTP; Telnet e E-mail.





# Arquitetura P2P pura

- ❑ Não há servidor sempre ligado
- ❑ Sistemas finais arbitrários se comunicam diretamente
- ❑ Pares estão conectados intermitentemente e mudam endereços IP



Altamente escalável  
Porém, difícil de gerenciar

# Aplicações atuais baseadas na arquitetura P2P

- ❑ Compartilhamento de arquivos (BitTorrent)
- ❑ Aceleração de download assistida por par (Xunlei)
- ❑ Telefonia por internet (Skype)
- ❑ IPTV (KanKan e Ppstrean)
- ❑ Algumas aplicações possuem arquiteturas híbridas combinando elementos de ambas.

# Processos em comunicação

**Processo:** programa que executa num hospedeiro

- processos no mesmo hospedeiro se comunicam usando **comunicação entre processos** definida pelo sistema operacional (SO)
- processos em hospedeiros distintos se comunicam trocando **mensagens através da rede**

**Processo cliente:**

processo que inicia a comunicação

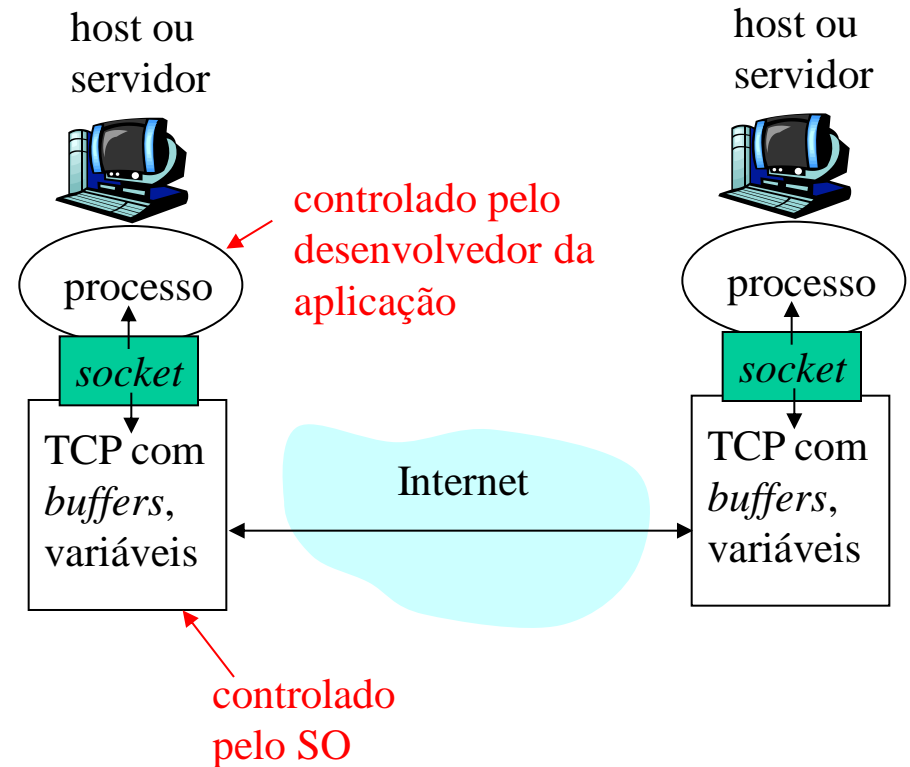
**Processo servidor:**

processo que espera para ser contatado

- Nota: aplicações com arquiteturas P2P possuem processos clientes e processos servidores

# Sockets

- ❑ Os processos enviam/recebem mensagens para/dos seus *sockets*
- ❑ Um *socket* é análogo a uma porta
  - Processo transmissor envia a mensagem através da porta
  - O processo transmissor assume a existência da infraestrutura de transporte no outro lado da porta que faz com que a mensagem chegue ao *socket* do processo receptor



- ❑ API- os únicos controles que o desenvolvedor da aplicação tem do lado da camada de transporte são: (1) escolha do protocolo de transporte; (2) talvez, a habilidade para fixar alguns parâmetros da camada de transporte .

# De que serviço de transporte uma aplicação precisa?

- ❑ A aplicação do lado remetente envia mensagens através da porta.
- ❑ Do outro lado, o protocolo da camada de transporte tem a responsabilidade de levar as mensagens pela rede até a porta do processo destinatário.

# De que serviço de transporte uma aplicação precisa?

- ❑ O desenvolvedor da aplicação escolhe um dos protocolos da camada de transporte disponíveis.
- ❑ Como fazer essa escolha?
- ❑ Avaliando os serviços providos disponíveis pelo protocolo de transporte, ou seja o que melhor atenda às necessidades!

# De que serviço de transporte uma aplicação precisa?

## Transferência confiável de dados

- ❑ algumas apls (p.ex. áudio) podem tolerar algumas perdas
- ❑ outras (p.ex., transf. de arquivos, telnet) requerem transferência 100% confiável

## Largura de banda (vazão)

- ❑ algumas apls (p.ex., multimídia) requerem quantia mínima de banda para serem "efetivas"
- ❑ outras apls ("apls elásticas") conseguem usar qq quantia de banda disponível

## Temporização

- ❑ algumas apls (p.ex., telefonia Internet, jogos interativos) requerem baixo retardo para serem "viáveis", ou seja, um longo atraso entre realizar uma ação e ver a reação faz com que a aplicação pareça menos realista.

## Segurança

## Requisitos do serviço de transporte de apls comuns

<b>Aplicação</b>	<b>Perdas</b>	<b>Banda</b>	<b>Sensibilidade temporal</b>
transferência de arqs	sem perdas	elástica	não
correio	sem perdas	elástica	não
documentos WWW	sem perdas	elástica	não
áudio/vídeo de tempo real	tolerante	áudio: 5Kb-1Mb vídeo: 10Kb-5Mb	sim, décimos de segundo
áudio/vídeo gravado	tolerante	como anterior	sim, alguns segs
jogos interativos	tolerante	> alguns Kbps	sim, décimos de..
apls financeiras	sem perdas	elástica	sim e não



# Serviços providos por protocolos de transporte Internet (redes TCP/IP)

## qual protocolo usar TCP ou UDP?

### Serviço TCP:

- ❑ *orientado a conexão:*  
inicialização requerida entre cliente e servidor
- ❑ *transporte confiável* entre processos remetente e receptor (dados sem erro e na ordem correta)
- ❑ *controle de fluxo:* remetente não vai "afogar" receptor
- ❑ *controle de congestionamento:*  
estrangular remetente quando a rede estiver carregada
- ❑ *não provê:* garantias temporais ou de banda mínima

### Serviço UDP:

- ❑ transferência de dados não confiável entre processos remetente e receptor
- ❑ não provê: estabelecimento da conexão, confiabilidade, controle de fluxo, controle de congestionamento, garantias temporais ou de banda mínima

P: Qual é o interesse em ter um UDP?

# Serviços não providos pelos protocolos de transporte da Internet

- Garantias de vazão
- Temporização

# Aplicações Internet: seus protocolos e seus protocolos de transporte

<b>Aplicação</b>	<b>Protocolo da camada de apl</b>	<b>Protocolo de transporte usado</b>
correio eletrônico	SMTP [RFC 2821]	TCP
acesso terminal remoto	telnet [RFC 854]	TCP
WWW	HTTP [RFC 2616]	TCP
transferência de arquivos	ftp [RFC 959]	TCP
streaming multimídia	proprietário (p.ex. RealNetworks)	TCP ou UDP
telefonia Internet	proprietário (p.ex., Dialpad)	tipicamente UDP

# Exercício

1. Qual a diferença entre aplicações de rede e protocolos da camada de aplicação?
2. De que modo mensagem instantânea é um híbrido das arquiteturas cliente-servidor e P2P?
3. O que é um socket?
4. Para uma sessão de comunicação entre um par de processos, qual processo é o cliente e qual é o servidor?
5. De que serviços de transporte uma aplicação precisa?
6. Explique o modelo de serviço do TCP.
7. Explique o modelo de serviço do UDP.
8. Cite algumas aplicações de rede e descreva qual o protocolo da camada de aplicação ela utiliza?