

Camada de aplicação

Prof. Jean Lima

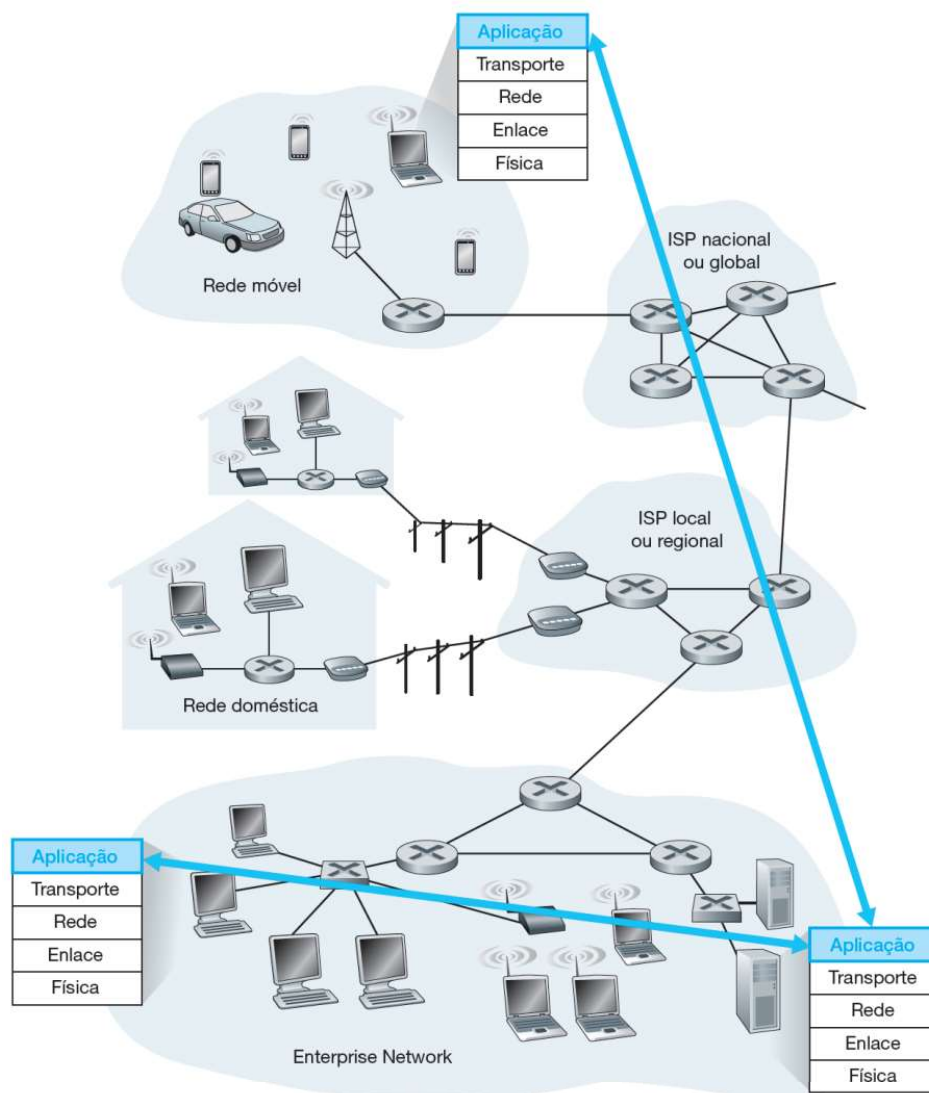
Princípios de aplicações de rede

- O núcleo do desenvolvimento de aplicação de rede é escrever programas que rodem em sistemas finais diferentes e se comuniquem entre si.
- Ao desenvolver sua nova aplicação, você precisará escrever um software que rode em vários sistemas finais.
- Esse software poderia ser criado, por exemplo, em C, Java ou Python.
- Você não precisará escrever programas que executem nos elementos do núcleo de rede, como roteadores e comutadores.

Arquiteturas de aplicação de rede

- A **arquitetura de rede** é fixa e provê um conjunto específico de serviços.
- A **arquitetura da aplicação** é projetada pelo programador e determina como a aplicação é organizada nos vários sistemas finais.
- Em uma **arquitetura cliente-servidor** há um hospedeiro sempre em funcionamento, denominado *servidor*, que atende a requisições de muitos outros hospedeiros, denominados *clientes*.

Arquiteturas de aplicação de rede

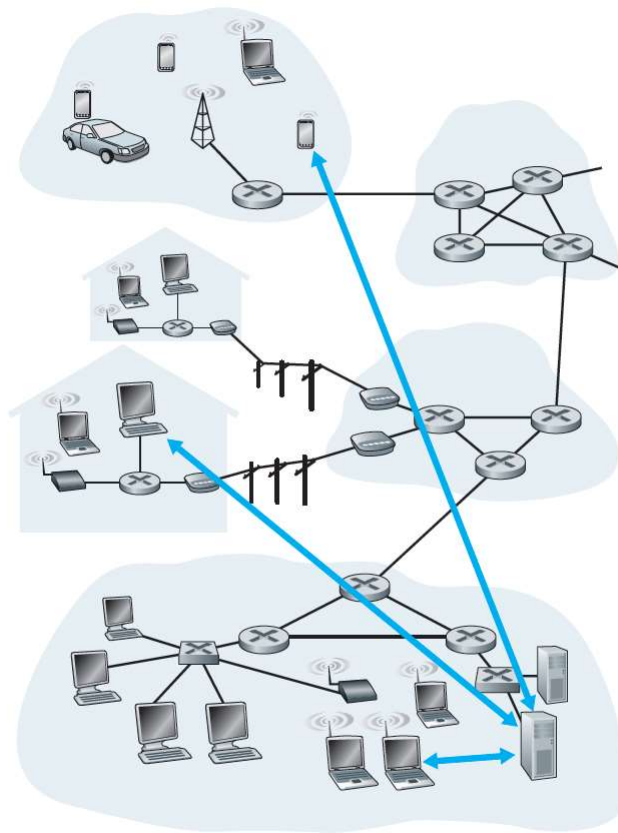


- A comunicação de uma aplicação de rede ocorre entre sistemas finais na camada de aplicação.

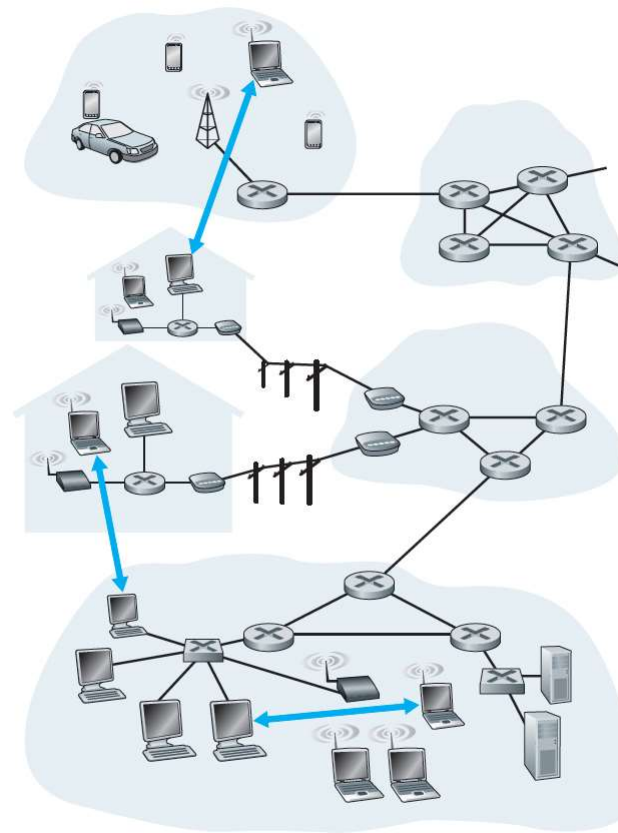
Arquiteturas de aplicação de rede

- A **arquitetura P2P** utiliza a comunicação direta entre duplas de hospedeiros conectados alternadamente, denominados *pares*.
- Uma das características mais fortes da arquitetura P2P é sua **autoescalabilidade**.
- As futuras aplicações P2P estão diante de três principais desafios:
 1. ISP Amigável.
 2. Segurança.
 3. Incentivos.

Arquiteturas de aplicação de rede



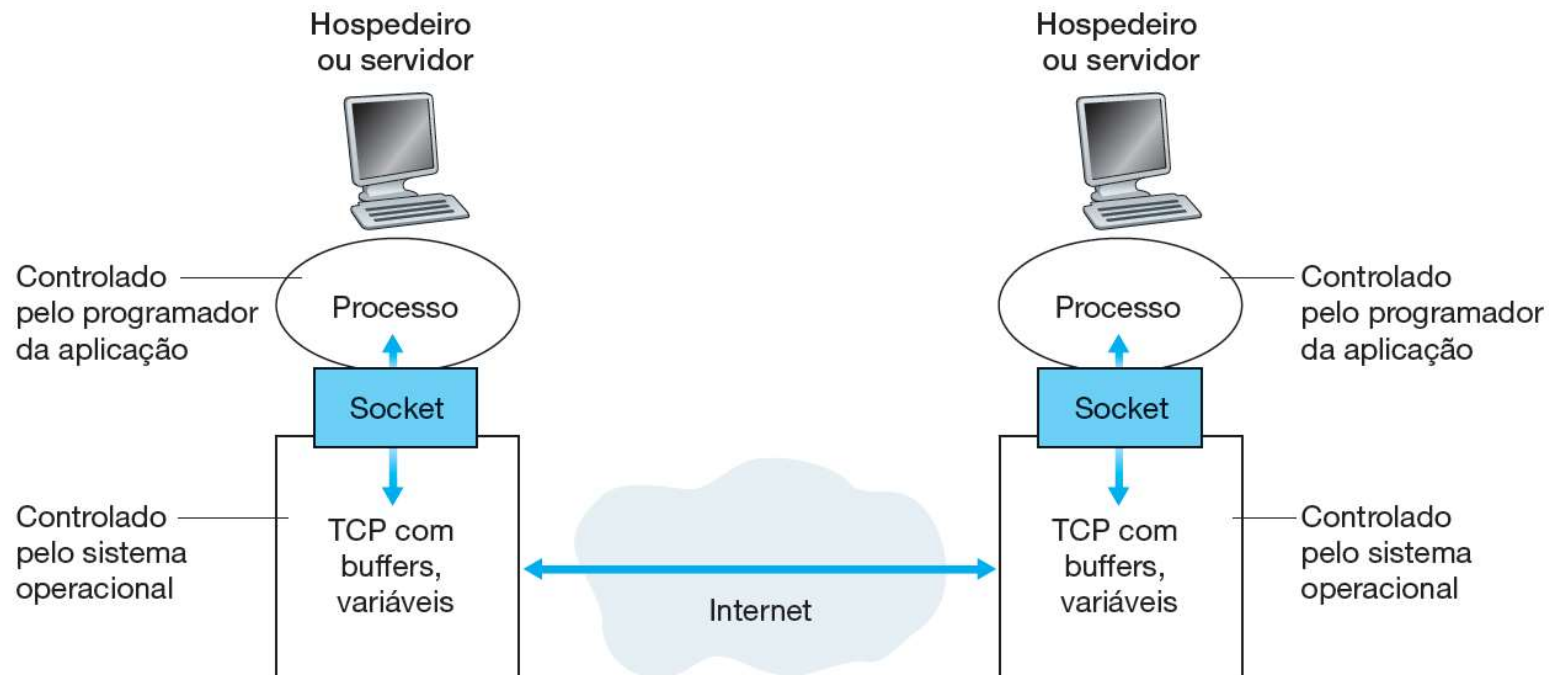
a. Arquitetura cliente-servidor



b. Arquitetura P2P

Comunicação entre processos

- Processos de aplicação, *sockets* e protocolo de transporte subjacente.



Comunicação entre processos

- Uma aplicação de rede consiste em pares de processos que enviam mensagens uns para os outros por meio de uma rede.
- Um processo envia mensagens para a rede e recebe mensagens dela através de uma interface de software denominada *socket*.
- Para identificar o processo receptor, duas informações devem ser especificadas:
 1. o endereço do hospedeiro e
 2. um identificador que especifica o processo receptor no hospedeiro de destino.

Serviços de transporte disponíveis para aplicações

- Transferência confiável de dados
- Vazão
- Temporização
- Segurança

Serviços de transporte providos pela Internet

- A Internet disponibiliza dois protocolos de transporte para aplicações, o UDP e o TCP.
- Requisitos de aplicações de rede selecionadas:

Aplicação	Perda de dados	Vazão	Sensibilidade ao tempo
Transferência / download de arquivo	Sem perda	Elástica	Não
E-mail	Sem perda	Elástica	Não
Documentos Web	Sem perda	Elástica (alguns kbits/s)	Não
Telefonia via Internet/ videoconferência	Tolerante à perda	Áudio: alguns kbits/s – 1Mbit/s Vídeo: 10 kbits/s – 5 Mbits/s	Sim: décimos de segundo
Áudio/vídeo armazenado	Tolerante à perda	Igual acima	Sim: alguns segundos
Jogos interativos	Tolerante à perda	Poucos kbits/s – 10 kbits/s	Sim: décimos de segundo
Mensagem instantânea	Sem perda	Elástico	Sim e não

Serviços de transporte providos pela Internet

- Aplicações populares da Internet, seus protocolos de camada de aplicação e seus protocolos de transporte subjacentes:

Aplicação	Protocolo de camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP [RFC 5321]	TCP
Acesso a terminal remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transferência de arquivos	FTP [RFC 959]	TCP
Multimídia em fluxo contínuo	HTTP (por exemplo, YouTube)	TCP
Telefonia por Internet	SIP [RFC 3261], RTP [RFC 3550] ou proprietária (por exemplo, Skype)	UDP ou TCP

Protocolos de camada de aplicação

Um protocolo de camada de aplicação define:

- Os tipos de mensagens trocadas.
- A sintaxe dos vários tipos de mensagens, tais como os campos da mensagem e como os campos são delineados.
- A semântica dos campos, isto é, o significado da informação nos campos.
- Regras para determinar quando e como um processo envia mensagens e responde a mensagens.

A Web e o HTTP

- Talvez o que mais atraia a maioria dos usuários da Web é que ela funciona por demanda.
- O HTTP — Protocolo de Transferência de Hipertexto (HyperText Transfer Protocol) —, o protocolo da camada de aplicação da Web, está no coração da Web e é definido no [RFC 1945] e no [RFC 2616].
- O HTTP é executado em dois programas:
 1. um cliente e
 2. outro servidor.

A Web e o HTTP

- Uma página Web é constituída de objetos.
- Um objeto é apenas um arquivo que se pode acessar com um único URL.
- A maioria das páginas Web é constituída de um arquivo-base HTML e diversos objetos referenciados.
- O HTTP usa o TCP como seu protocolo de transporte subjacente.
- O HTTP é denominado um protocolo sem estado.

Conexões persistentes e não persistentes

- Quando a interação cliente-servidor acontece por meio de conexão TCP, o programador da aplicação precisa tomar uma importante decisão:
- **Conexões não persistentes** — cada par de requisição/resposta deve ser enviado por uma conexão TCP distinta.
- **Conexões persistentes** — todas as requisições e suas respostas devem ser enviadas por uma mesma conexão TCP.

Formato da mensagem HTTP

Mensagem de requisição HTTP

- Apresentamos a seguir uma mensagem de requisição HTTP típica:

GET /somedir/page.html HTTP/1.1

Host: www.someschool.edu

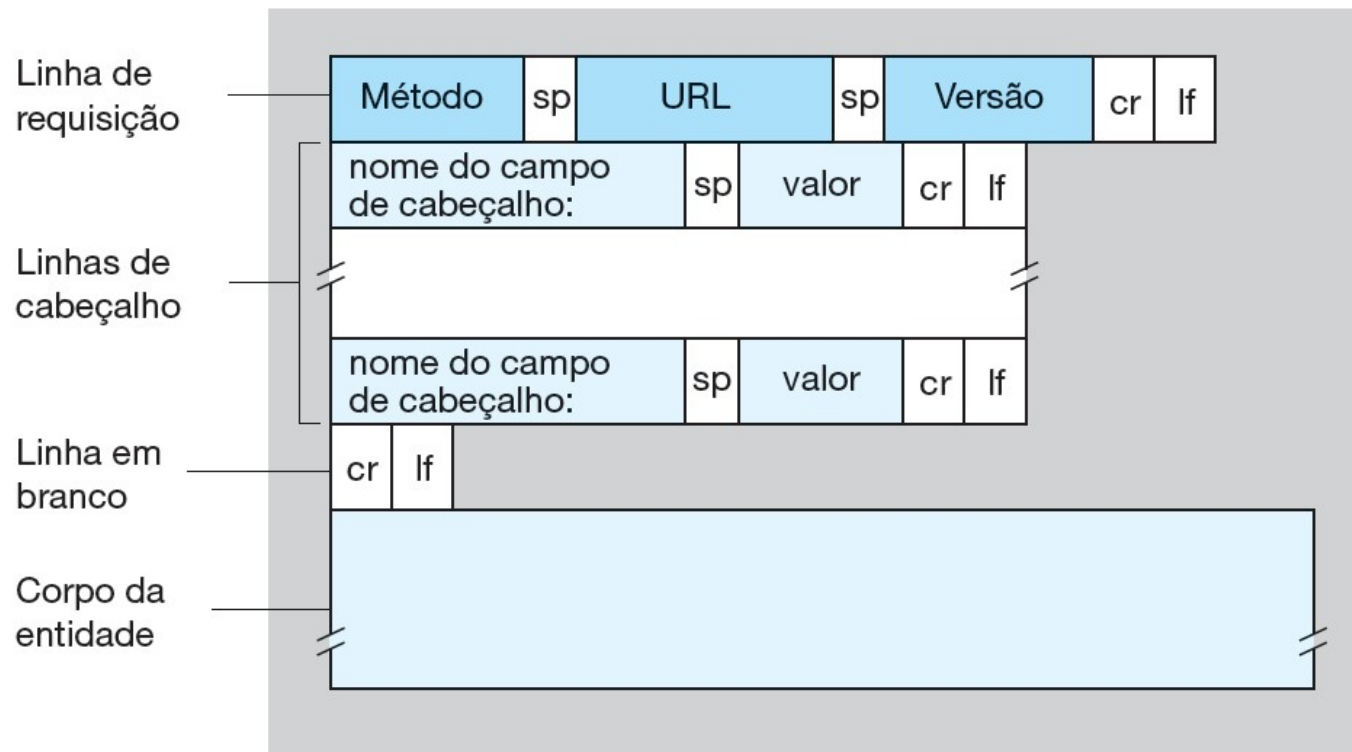
Connection: close

User-agent: Mozilla/5.0

Accept-language: fr

Formato da mensagem HTTP

- Formato geral de uma mensagem de requisição HTTP



Formato da mensagem HTTP

Mensagem de resposta HTTP

- Apresentamos a seguir uma mensagem de resposta HTTP típica:

HTTP/1.1 200 OK

Connection: close

Date: Tue, 09 Aug 2011 15:44:04 GMT

Server: Apache/2.2.3 (CentOS)

Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT

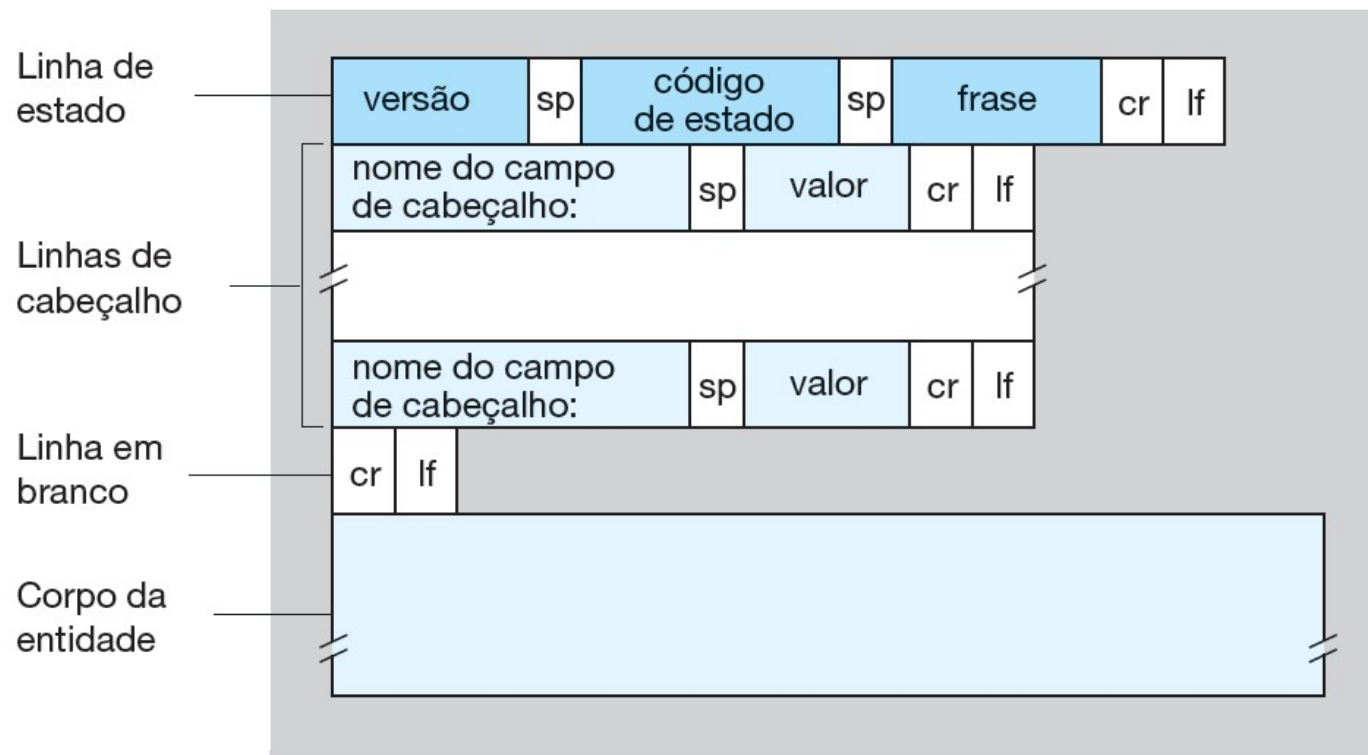
Content-Length: 6821

Content-Type: text/html

(dados dados dados dados dados ...)

Formato da mensagem HTTP

- Formato geral de uma mensagem de resposta HTTP



Interação usuário-servidor:

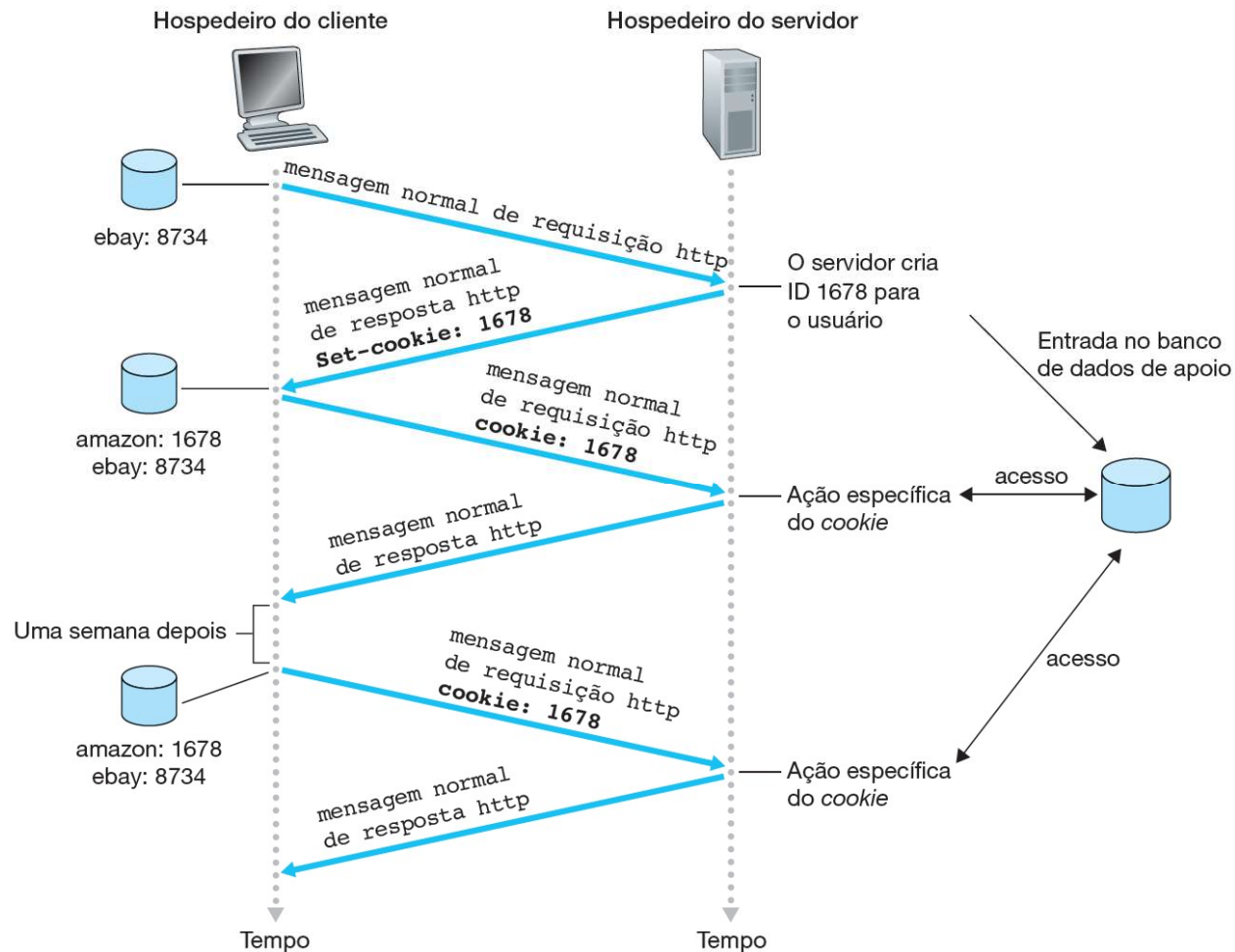
cookies

Cookies, definidos no [RFC 6265], permitem que sites monitorem seus usuários.

A tecnologia dos cookies tem quatro componentes:

1. uma linha de cabeçalho de cookie na mensagem de resposta HTTP;
2. uma linha de cabeçalho de cookie na mensagem de requisição HTTP;
3. um arquivo de cookie mantido no sistema final do usuário e gerenciado pelo navegador do usuário;
4. um banco de dados de apoio no site.

Interação usuário-servidor: *cookies*

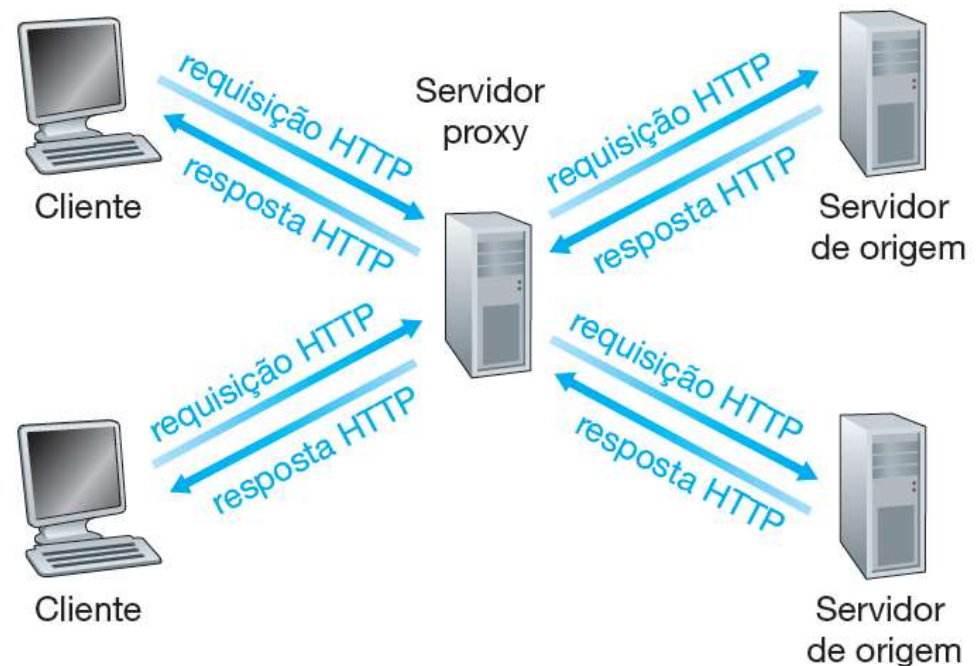


- Mantendo o estado do usuário com *cookies*.

Caches Web

- Um **cache Web** — também denominado **servidor proxy** — é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem.

Clientes requisitando objetos por meio de um *cache Web*:



GET condicional

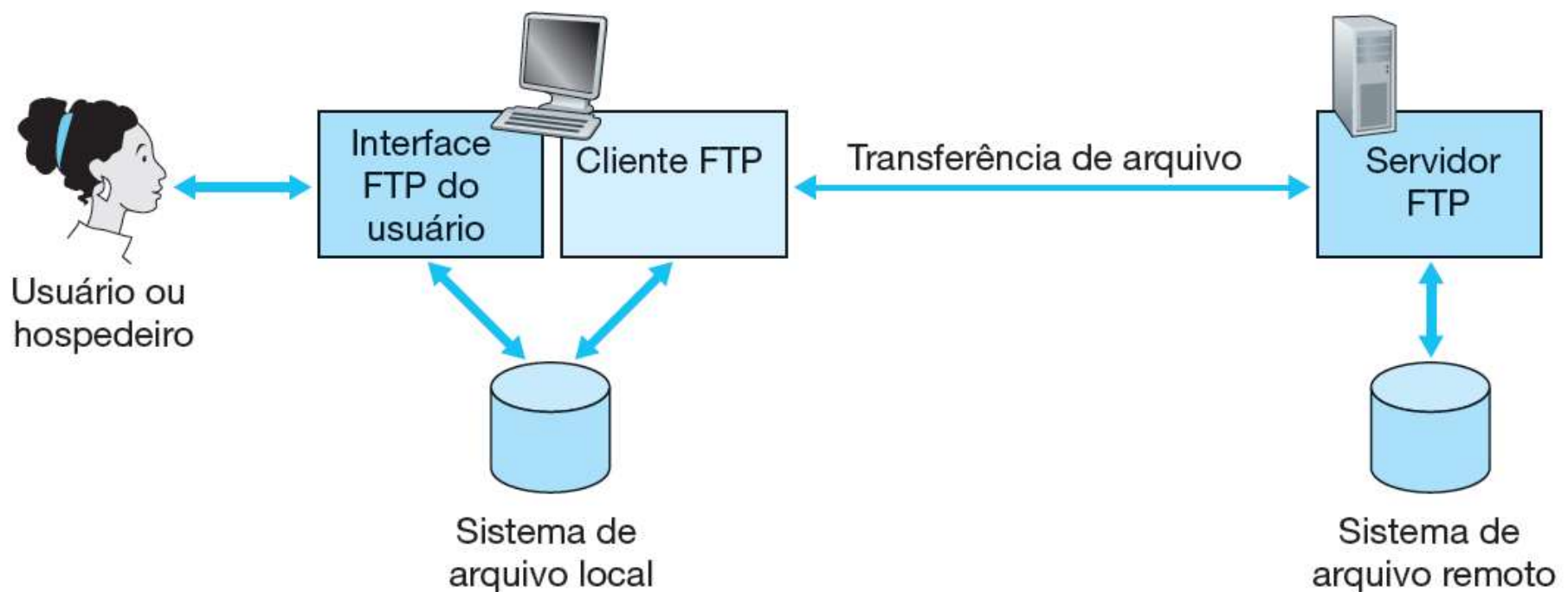
- GET condicional – mecanismo que permite que um *cache* verifique se seus objetos estão atualizados.

Transferência de arquivo: FTP

- Em uma sessão FTP típica, o usuário quer transferir arquivos de ou para um hospedeiro remoto.
- HTTP e FTP são protocolos de transferência de arquivos e têm muitas características em comum.

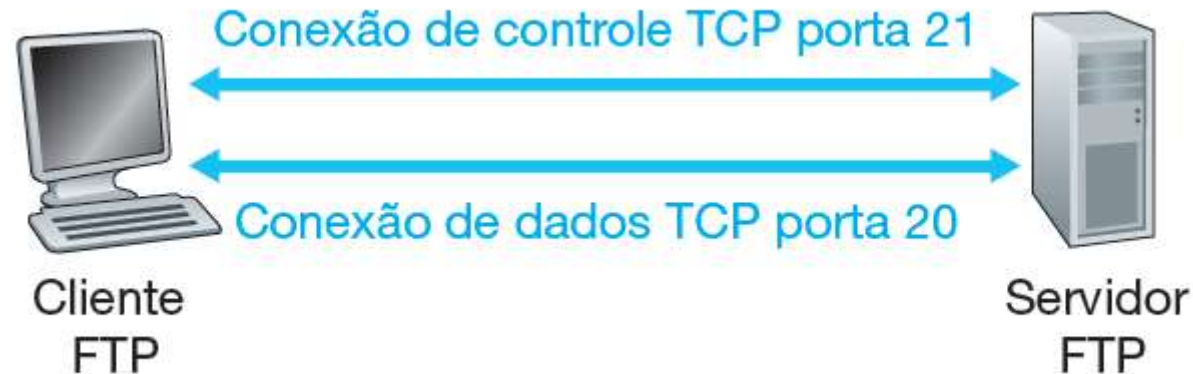
Transferência de arquivo: FTP

- FTP transporta arquivos entre sistemas de arquivo local e remoto:



Transferência de arquivo: FTP

- Conexões de controle e de dados:



Camadas e respostas FTP

Alguns dos comandos mais comuns são descritos a seguir:

- USER username: usado para enviar identificação do usuário ao servidor.
- PASS password: usado para enviar a senha do usuário ao servidor.
- LIST: usado para pedir ao servidor que envie uma lista com todos os arquivos existentes no atual diretório remoto.
- RETR filename: usado para extrair um arquivo do diretório atual do hospedeiro remoto.

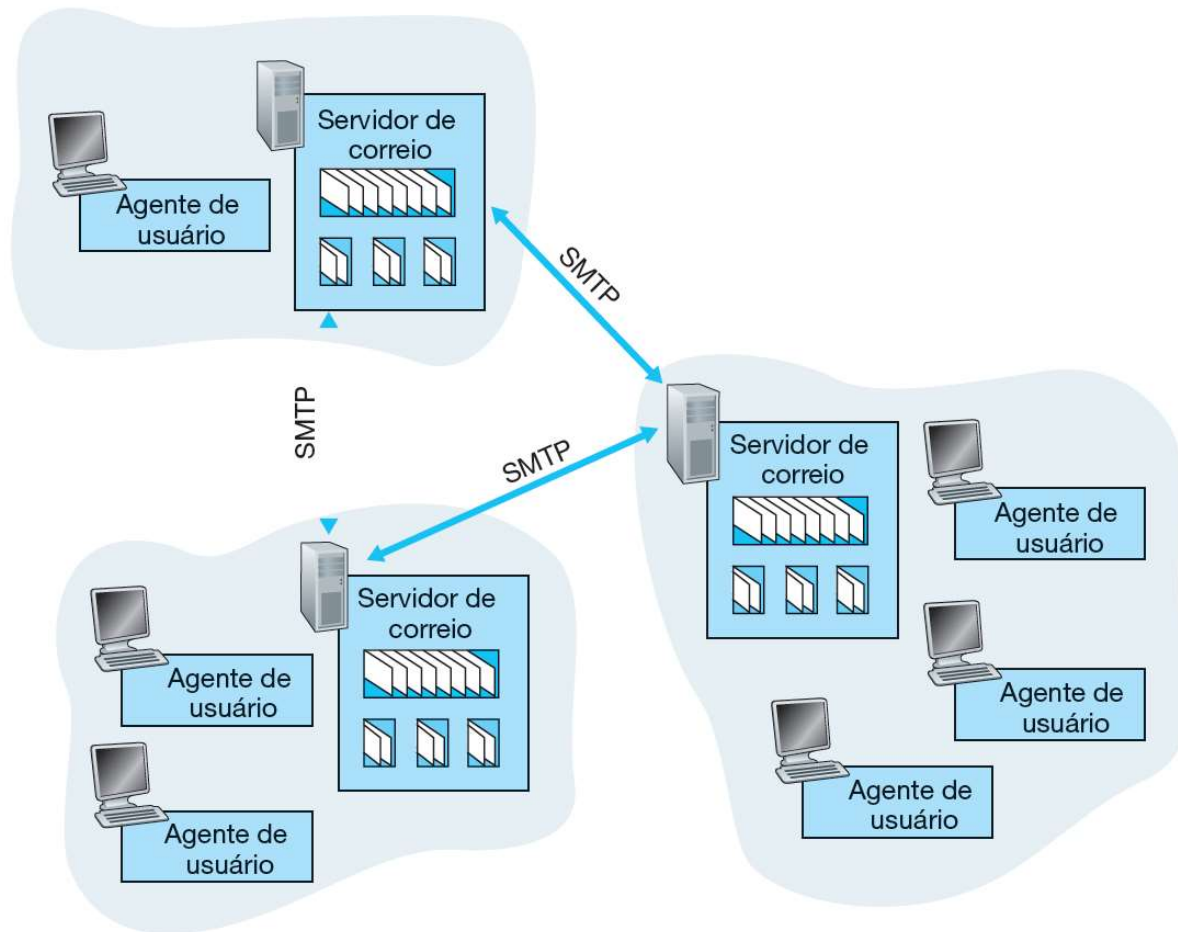
Camadas e respostas FTP

- STOR filename: usado para armazenar um arquivo no diretório atual do hospedeiro remoto.

Algumas respostas típicas, junto com suas possíveis mensagens, são as seguintes:

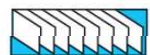
- 331 Nome de usuário OK, senha requisitada
- 125 Conexão de dados já aberta; iniciando transferência
- 425 Não é possível abrir a conexão de dados
- 452 Erro ao escrever o arquivo

Correio eletrônico na Internet



- Uma visão do sistema de e-mail da Internet.

Legenda:



Fila de mensagem de saída

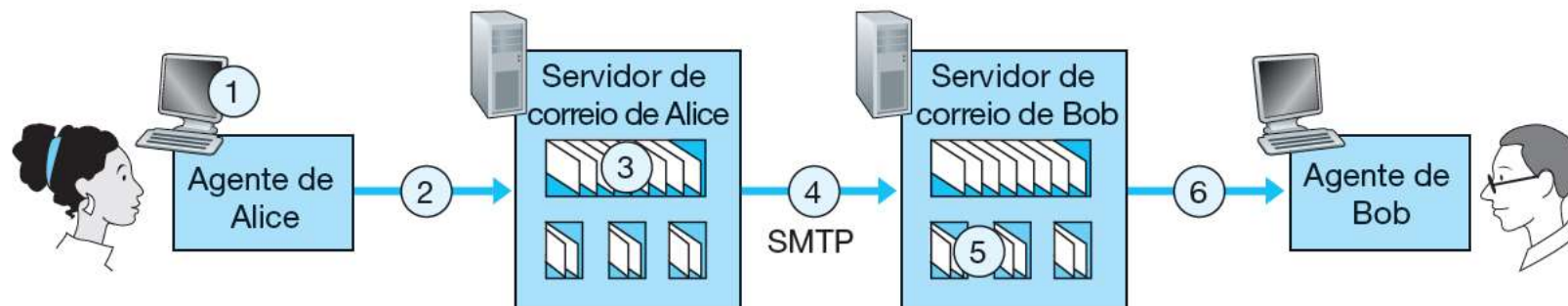


Caixa de entrada do usuário

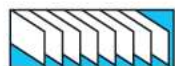
SMTP

- O SMTP transfere mensagens de servidores de correio remetentes para servidores de correio destinatários.

Alice envia uma mensagem a Bob:



Legenda:



Fila de mensagem



Caixa postal do usuário

Formatos de mensagem de correio

- Um cabeçalho de mensagem típico é semelhante a:

From: alice@crepes.fr

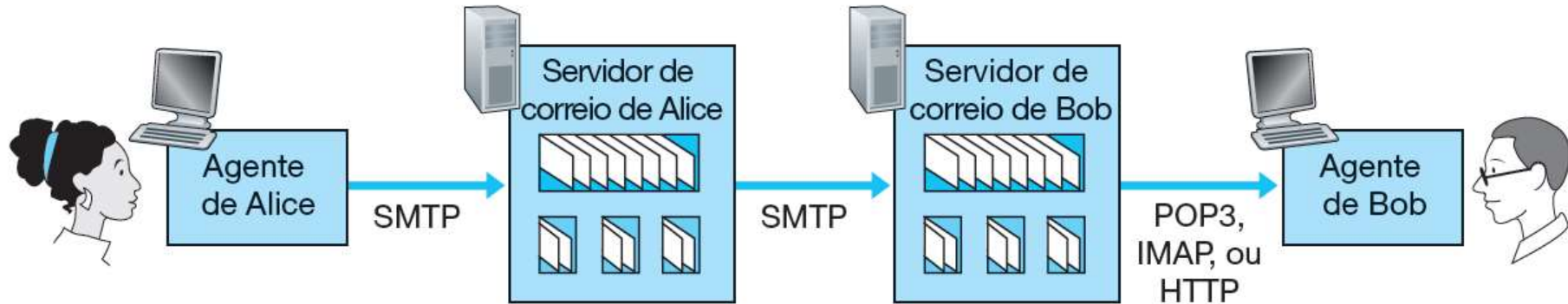
To: bob@hamburger.edu

Subject: Searching for the meaning of life.

- Após o cabeçalho da mensagem, vem uma linha em branco e, em seguida, o corpo da mensagem (em ASCII).
- Você pode usar o Telnet para enviar a um servidor de correio uma mensagem que contenha algumas linhas de cabeçalho, inclusive Subject:. Para tal, utilize o comando telnet serverName 25.

Protocolos de acesso ao correio

- Protocolos de e-mail e suas entidades comunicantes



DNS: o serviço de diretório da Internet

- Há duas maneiras de identificar um hospedeiro — por um nome de hospedeiro e por um endereço IP.
- Para conciliar isso, é necessário um serviço de diretório que traduza nomes de hospedeiro para endereços IP.
- Esta é a tarefa principal do DNS da Internet.
- O DNS é (1) um banco de dados distribuído executado em uma hierarquia de servidores de DNS, e (2) um protocolo de camada de aplicação que permite que hospedeiros consultem o banco de dados distribuído.

DNS: o serviço de diretório da Internet

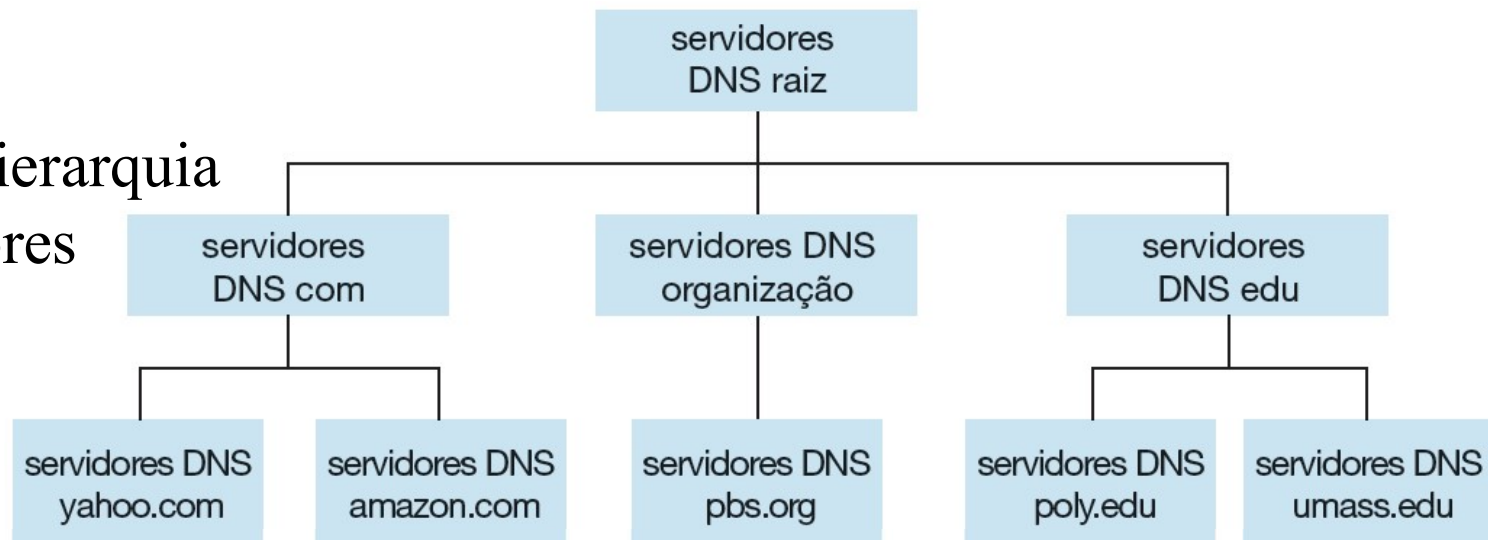
O DNS provê alguns outros serviços importantes além da tradução de nomes de hospedeiro para endereços IP:

- Apelidos (*aliasing*) de hospedeiro.
- Apelidos de servidor de correio.
- Distribuição de carga.

DNS: o serviço de diretório da Internet

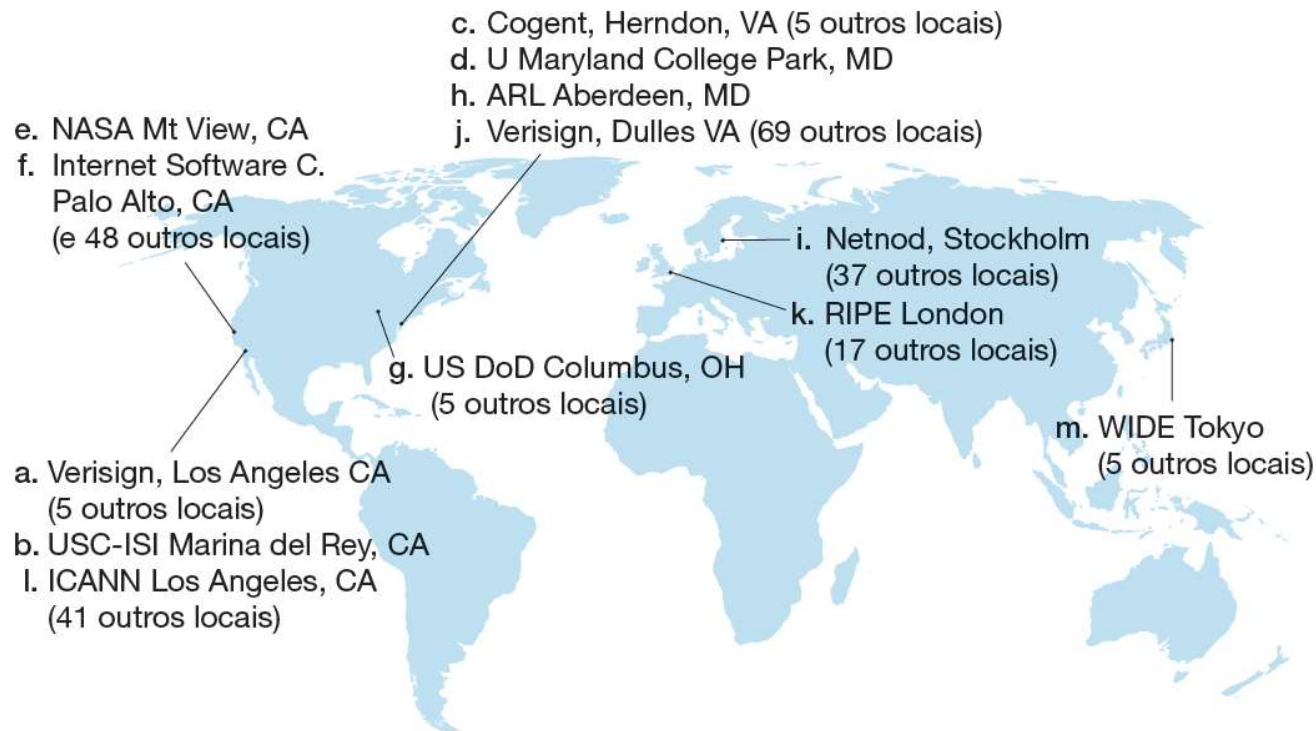
- Nenhum servidor DNS isolado tem todos os mapeamentos para todos os hospedeiros da Internet.
- Em vez disso, os mapeamentos são distribuídos pelos servidores DNS.

Parte da hierarquia de servidores DNS

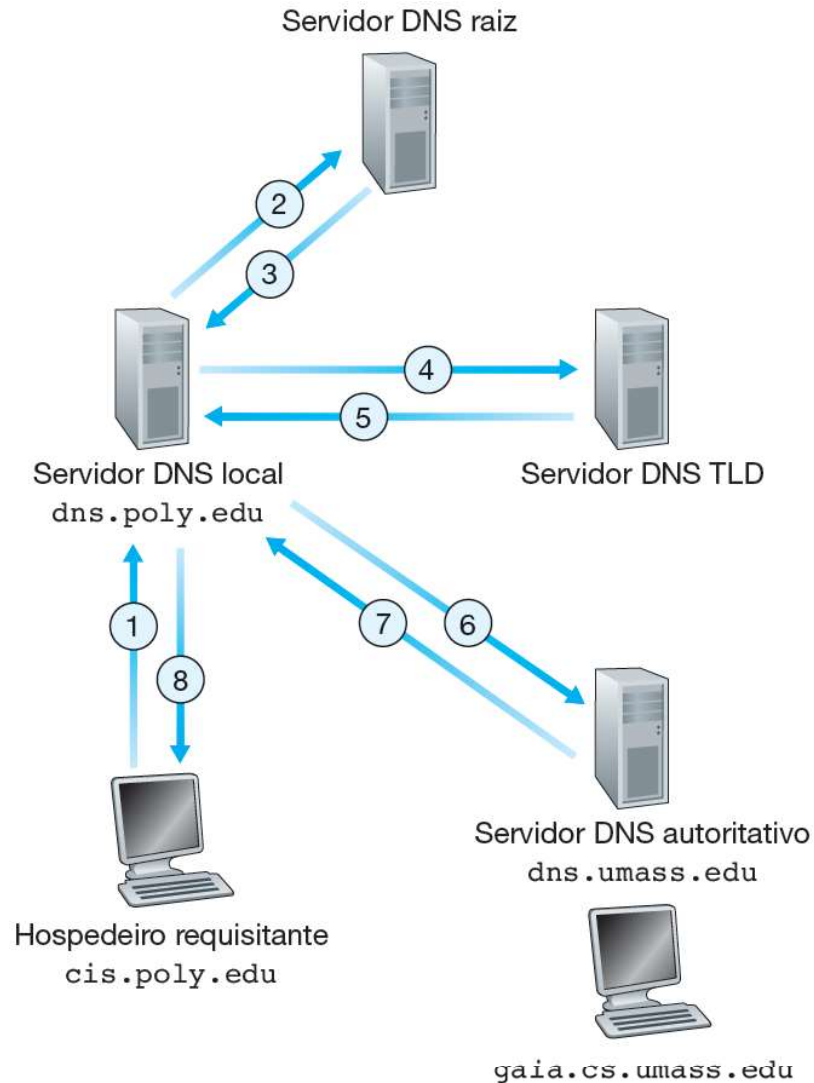


DNS: o serviço de diretório da Internet

- Servidores DNS raiz em 2012 (nome, organização, localização)



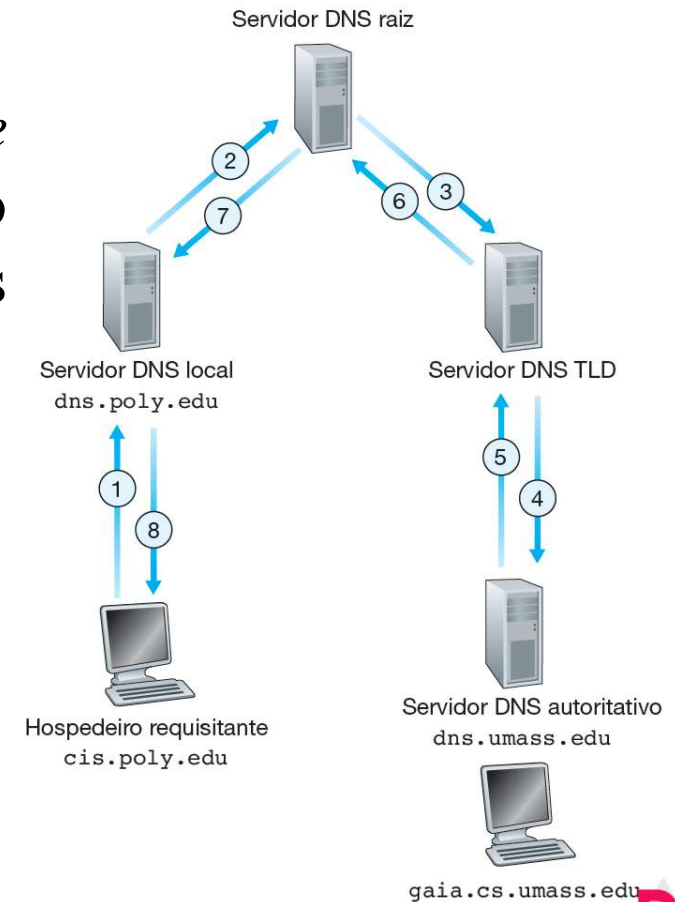
DNS: o serviço de diretório da Internet



- Interação dos diversos servidores DNS:

DNS: o serviço de diretório da Internet

- O DNS explora extensivamente o *cache* para melhorar o desempenho quanto ao atraso e reduzir o número de mensagens DNS que dispara pela Internet.
- Consultas recursivas em DNS:



Registros e mensagens DNS

- Um registro de recurso é uma tupla de quatro elementos que contém os seguintes campos:

(Name, Value, Type, TTL)

- Formato da mensagem DNS

Identificação	Flags	12 bytes
Número de perguntas	Número de RRs de resposta	
Número de RRs autoritativos	Número de RRs adicionais	
Perguntas (número variável de perguntas)		Nome, campos de tipo para uma consulta
Respostas (número variável de registros de recursos)		RRs de resposta à consulta
Autoridade (número variável de registros de recursos)		Registros para servidores com autoridade
Informação adicional (número variável de registros de recursos)		Informação adicional 'útil', que pode ser usada

Aplicações P2P

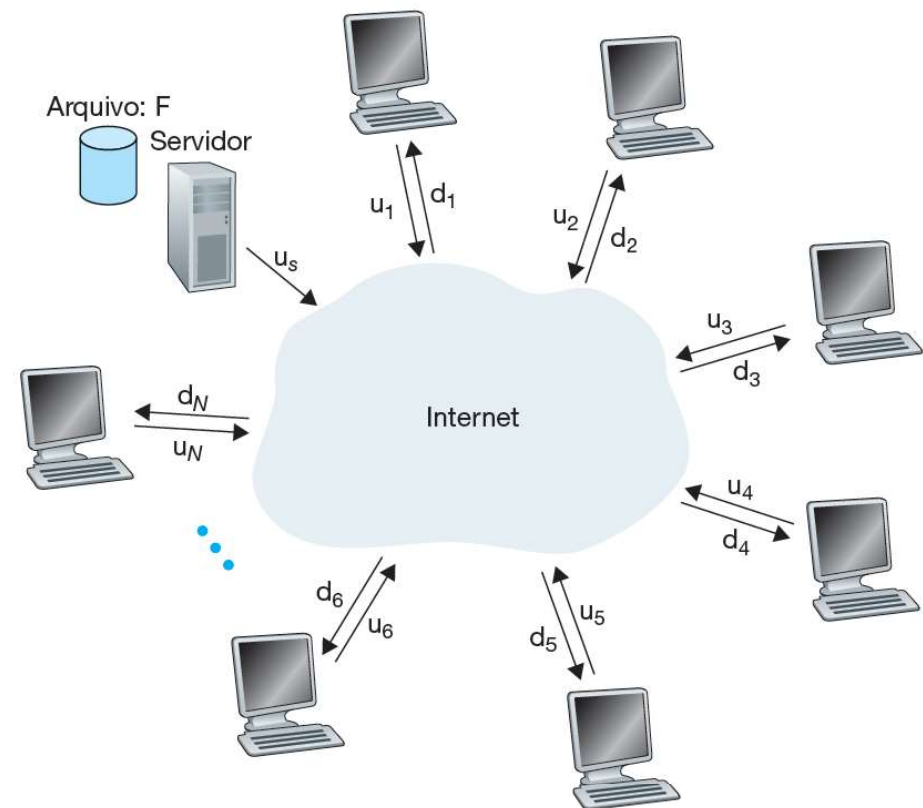
Distribuição de arquivos P2P

- Na distribuição de arquivos P2P, cada par pode redistribuir qualquer parte do arquivo recebido para outros pares, auxiliando, assim, o servidor no processo de distribuição.
- O tempo de distribuição é o tempo necessário para que todos os N pares obtenham uma cópia do arquivo.
- O BitTorrent é um protocolo P2P popular para distribuição de arquivos.

Aplicações P2P

Distribuição de arquivos P2P

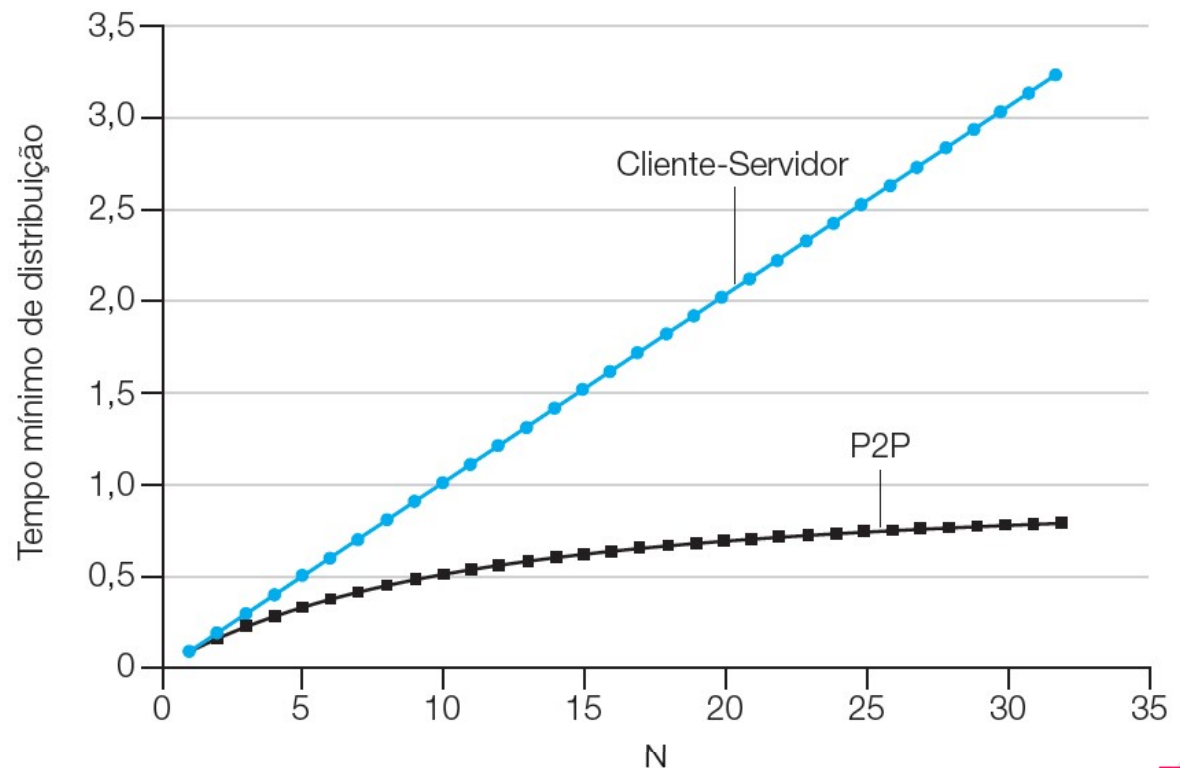
Um problema ilustrativo de distribuição de arquivo



Aplicações P2P

Distribuição de arquivos P2P

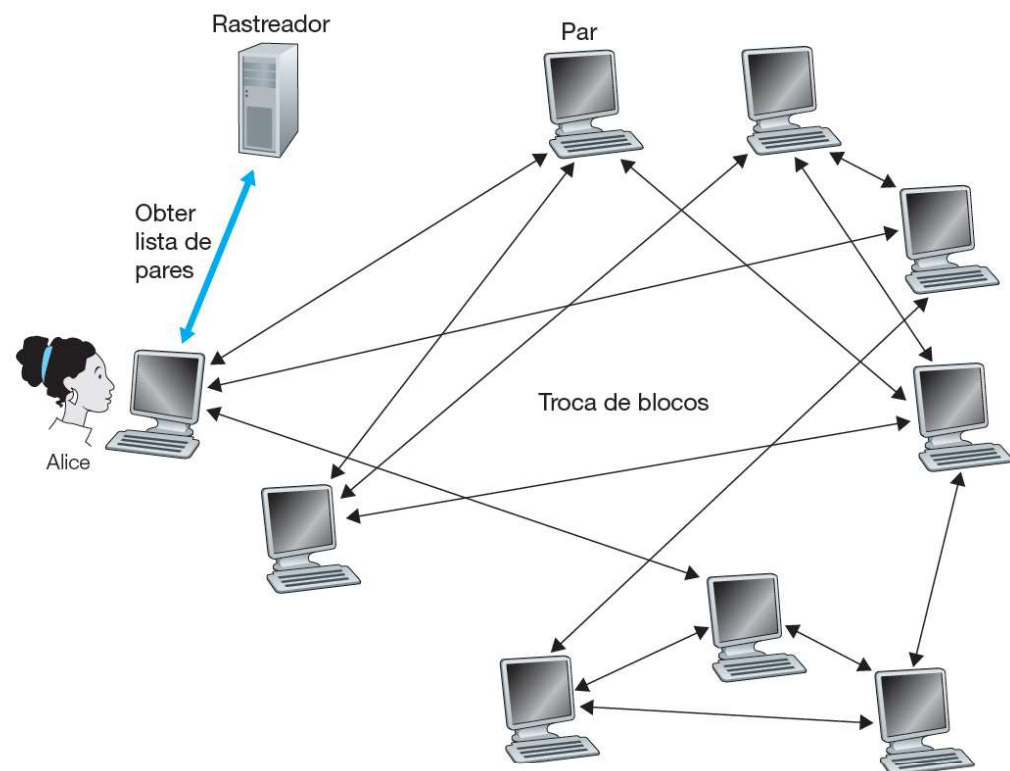
Tempo de distribuição
para arquiteturas P2P
e cliente-servidor



Aplicações P2P

Distribuição de arquivos P2P

Distribuição de arquivos com o BitTorrent



Aplicações P2P

Distributed Hash Tables (DHTs)

- Vamos considerar como montar uma versão distribuída, P2P, de um banco de dados, que guardará os pares (chave, valor) por milhões.
- No sistema P2P, cada par só manterá um pequeno subconjunto da totalidade (chave, valor).
- Permitiremos que qualquer par consulte o banco de dados distribuído com uma chave em particular.

Aplicações P2P

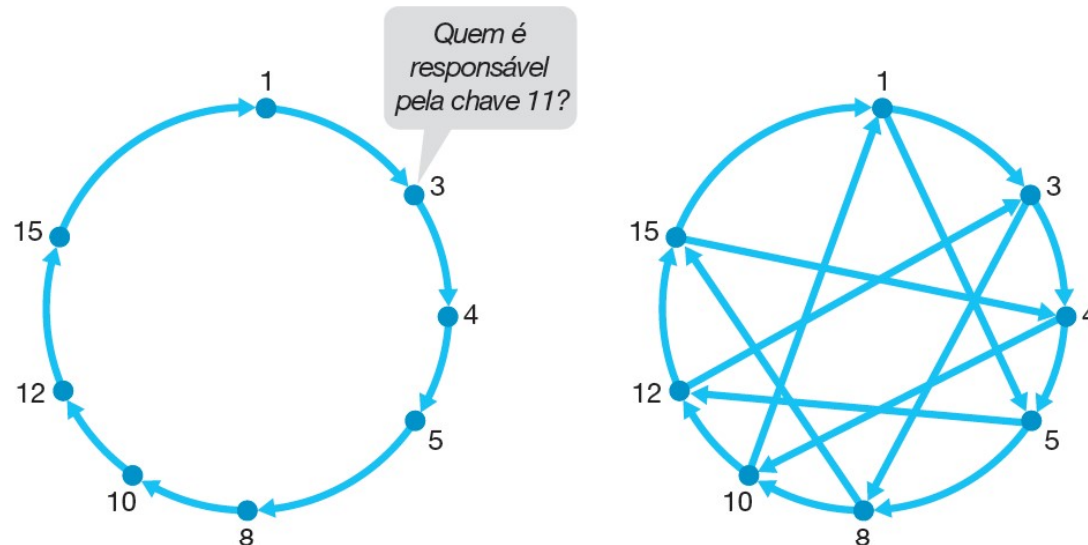
Distributed Hash Tables (DHTs)

- O banco de dados distribuído, então, localizará os pares que possuem os pares (chave, valor) correspondentes e retornará os pares chave-valor ao consultante.
- Qualquer par também poderá inserir novos pares chave-valor no banco de dados.
- Esse banco de dados distribuído é considerado como uma **tabela *hash* distribuída (DHT — Distributed Hash Table)**.

Aplicações P2P

Distributed Hash Tables (DHTs)

- O DHT circular oferece uma solução bastante elegante para reduzir a quantidade de informação sobreposta que cada par deve gerenciar.



Aplicações P2P

Distributed Hash Tables (DHTs)

- Em sistemas P2P, um par pode vir ou ir sem aviso.
- Suponha que o par 5 da figura anterior saia de modo abrupto.
- Os dois pares precedentes ao que saiu (4 e 3) saberão que o par saiu, pois não responde mais às mensagens de ping.
- Os pares 4 e 3 precisam, portanto, atualizar as informações do estado de seu sucessor.

Aplicações P2P

Distributed Hash Tables (DHTs)

- Consideraremos agora como o par 4 atualiza seu estado:
 1. O par 4 substitui seu primeiro sucessor (par 5) por seu segundo sucessor (par 8).
 2. O par 4, então, pergunta a seu novo primeiro sucessor (par 8) o identificador e o endereço IP de seu sucessor imediato (par 10). O par 4, então, torna o par 10 seu segundo sucessor.

Programação de *sockets*: criando aplicações de rede

- Há dois tipos de aplicações de rede.
- Um deles é uma execução cuja operação é especificada em um padrão de protocolo.
- O outro tipo de aplicação de rede é uma aplicação de rede proprietária.

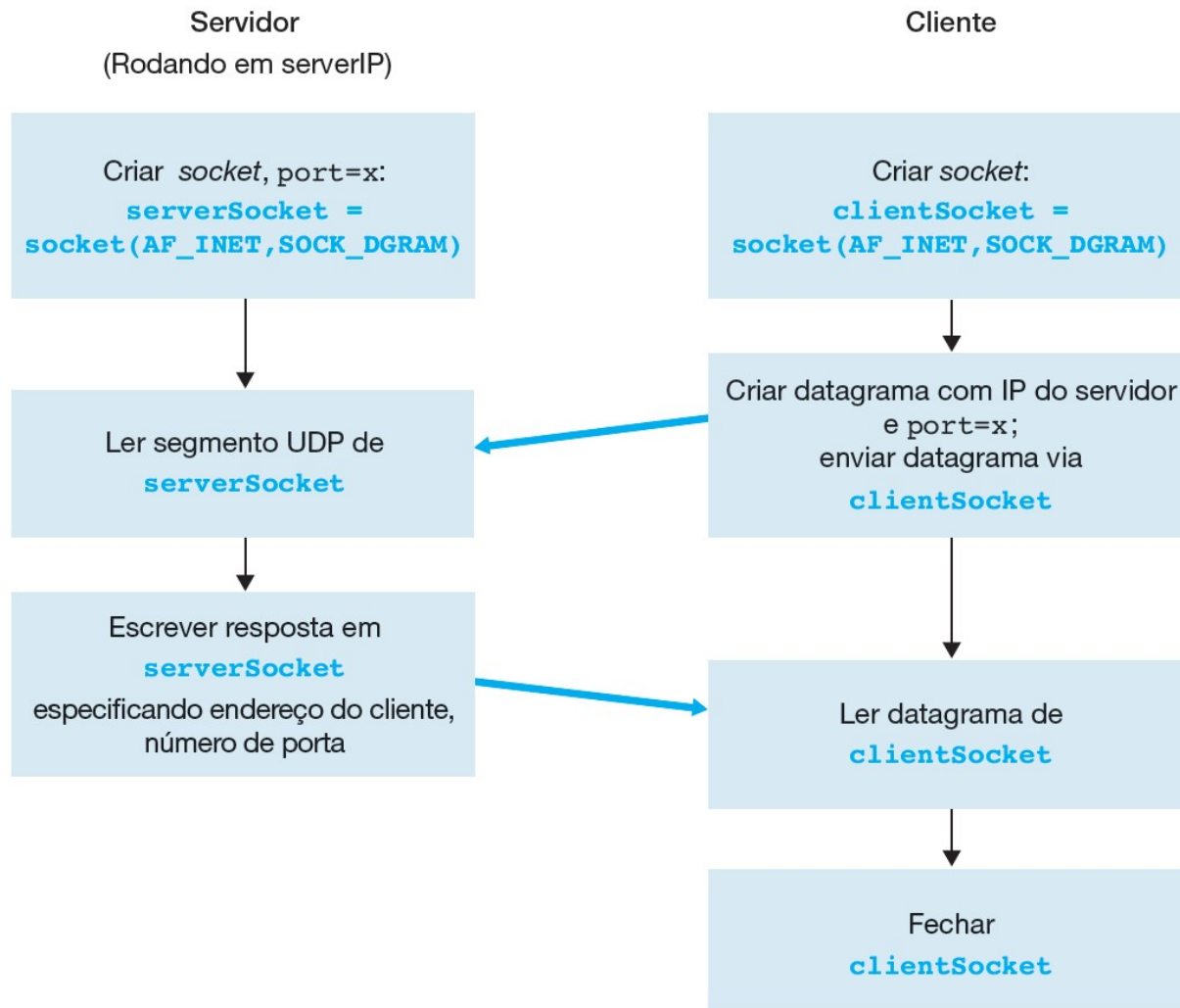
Programação de sockets com UDP

- Usaremos a aplicação cliente-servidor simples a seguir para demonstrar a programação de socket para UDP e TCP:

Programação de sockets com UDP

1. Um cliente lê uma linha de caracteres (dados) do teclado e a envia para o servidor.
2. O servidor recebe os dados e converte os caracteres para maiúsculas.
3. O servidor envia os dados modificados ao cliente.
4. O cliente recebe os dados modificados e apresenta a linha em sua tela.

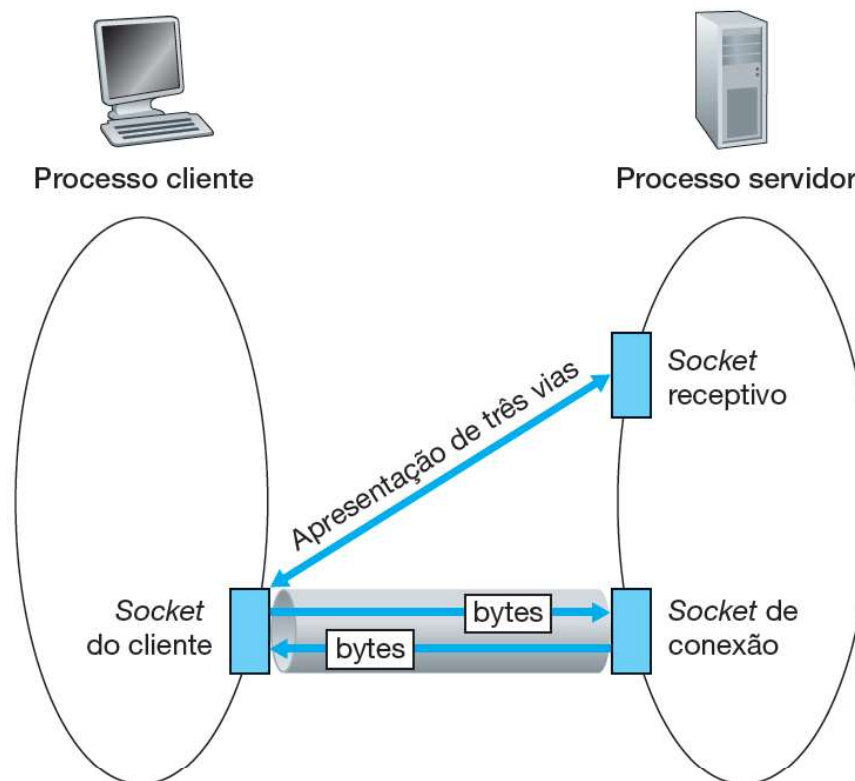
Programação de sockets com UDP



A aplicação
cliente-servidor
usando UDP

Programação de sockets com TCP

- O processo TCPServer tem dois *sockets*



Programação de sockets com TCP

- A aplicação cliente-servidor usando TCP

