Redes de Computadores

Camada de Aplicação

Prof. Me. Ricardo Girnis Tombi

Conteúdo adaptado de:

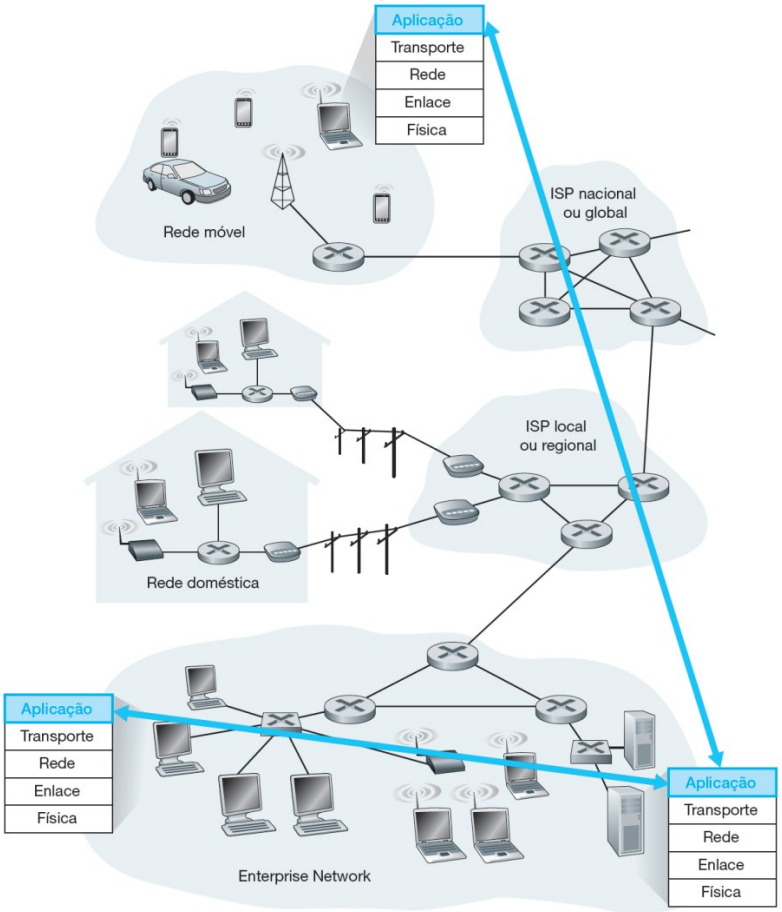
Redes de Computadores e a Internet. Ed. Pearson

Camada de aplicação

Aplicações

- E-mail
- Web
- Mensagem instantânea
- Login remoto
- P2P file sharing
- Jogos de rede multiusuário
- Telefonia via Internet
- Videoconferência em tempo real

Aplicações na rede



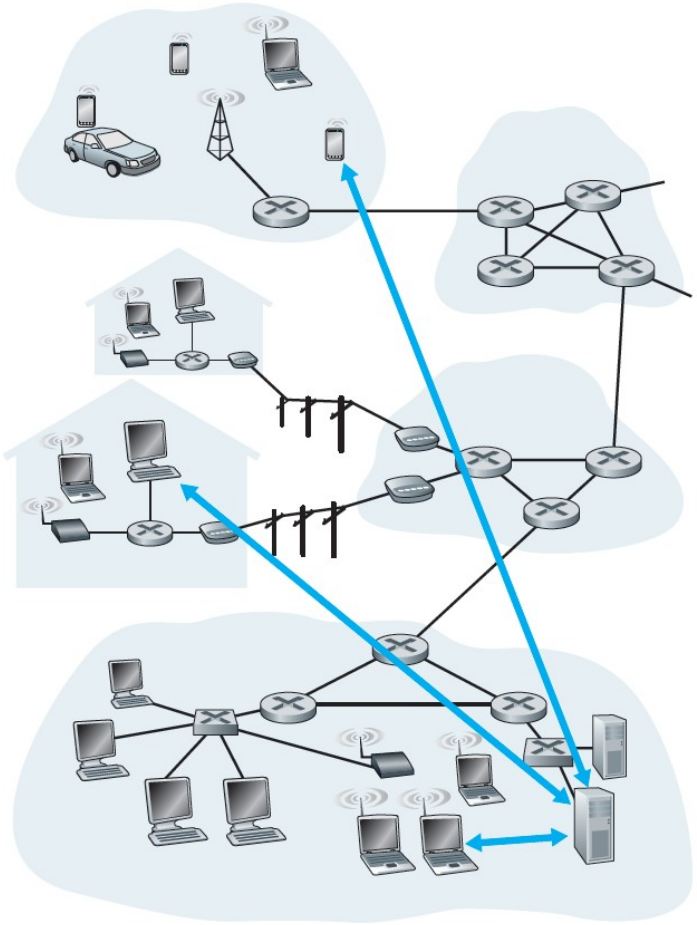
Escrever programas que

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

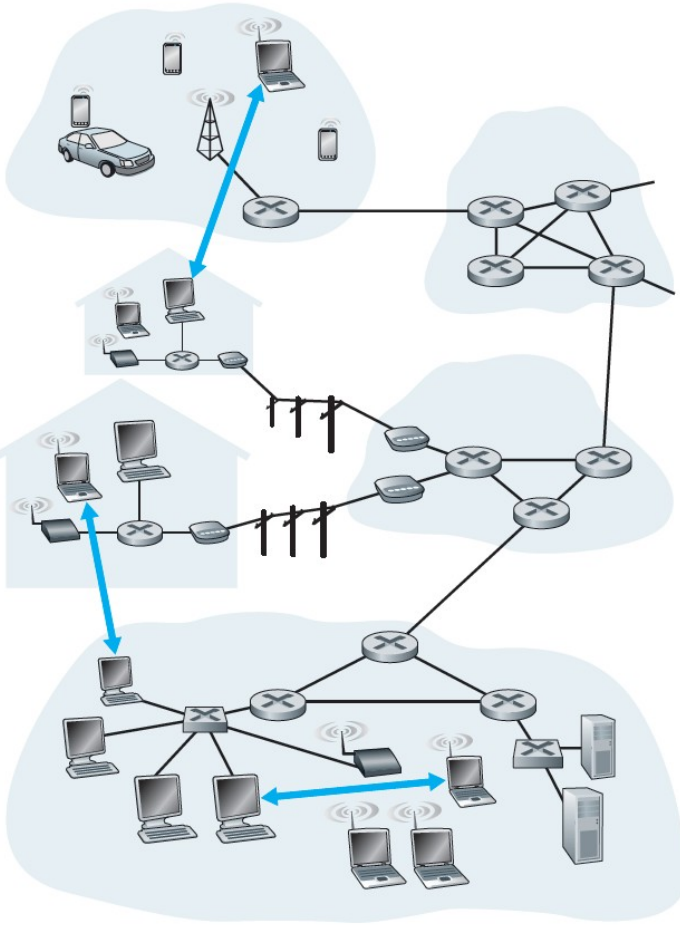
Nenhum software é escrito para dispositivos no núcleo da rede

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

Arquiteturas aplicação



a. Arquitetura cliente-servidor



b. Arquitetura P2P

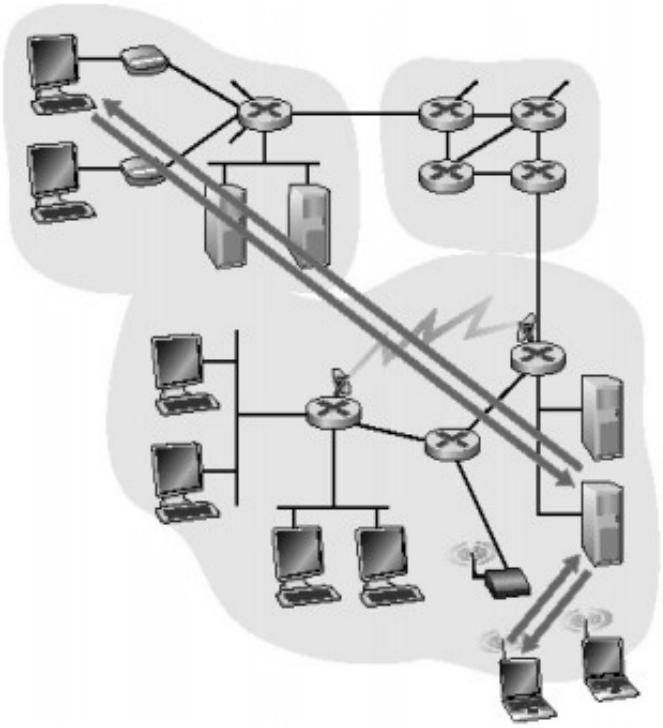
Cliente - Servidor

Servidor:

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

Clientes:

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



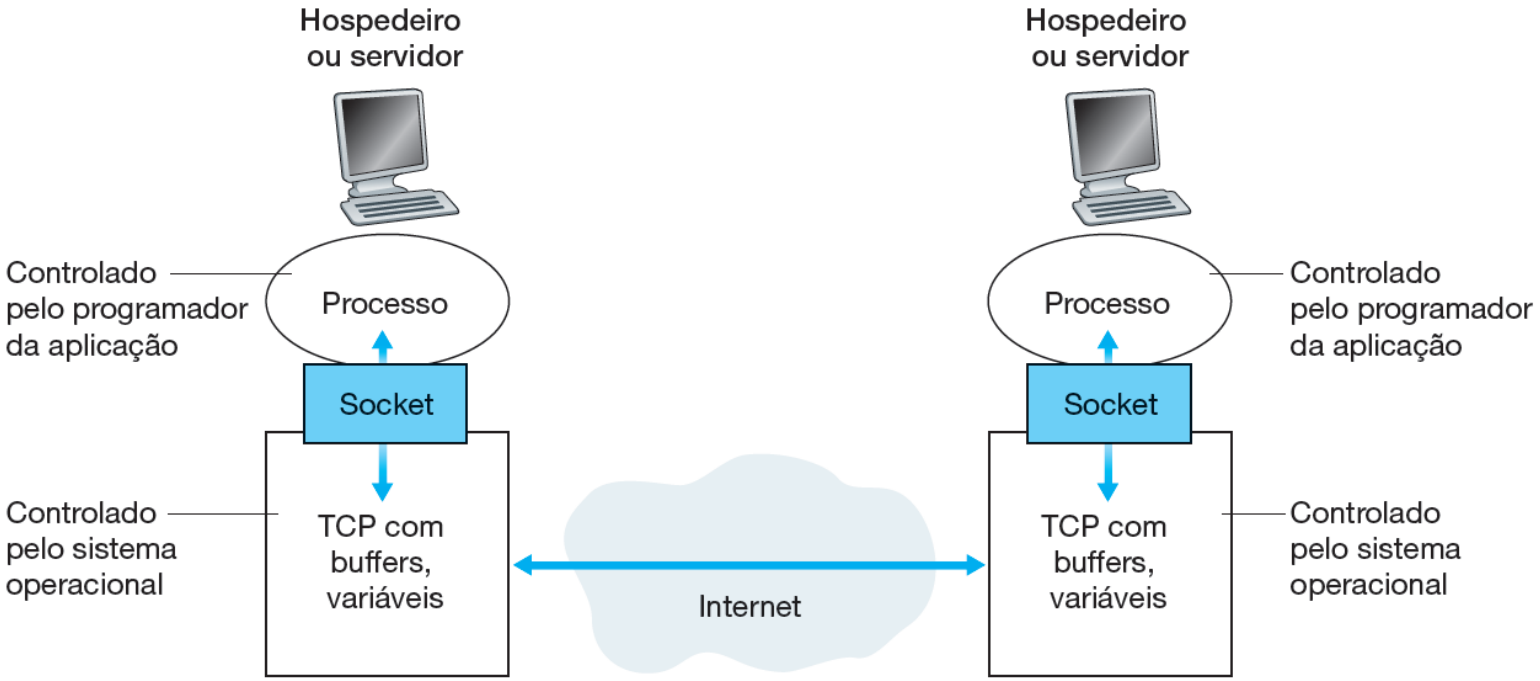
a. Aplicação cliente-servidor

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

Comunicação de processos



- 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

Endereçamento de processos

- Para um processo receber mensagens, ele deve ter um identificador
- Um hospedeiro possui um único endereço IP de 32 bits

O endereço IP do hospedeiro onde o processo está executando é suficiente para identificar o processo?

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

Serviços de transporte disponíveis para aplicações

- Transferência confiável de dados
- Vazão
- Temporização
- Segurança

Relembrando ...

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

A Internet disponibiliza dois protocolos de transporte para aplicações, o UDP e o TCP.

Relembrando ...

A Internet disponibiliza dois protocolos de transporte para aplicações, o UDP e o TCP.

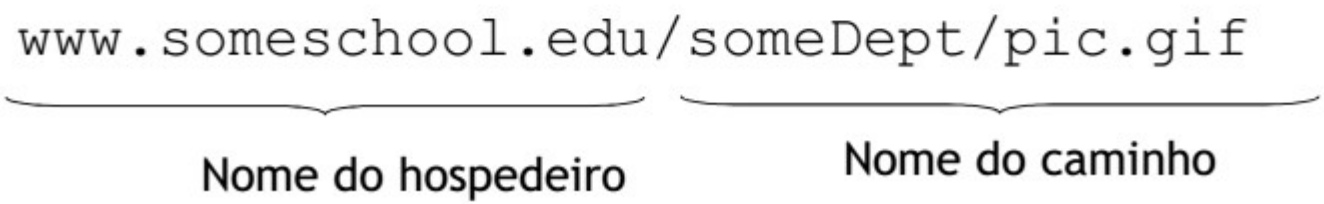
Aplicações populares da Internet, seus protocolos de camada de aplicação e seus protocolos de transporte subjacentes

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

Web e HTTP

Primeiro alguns jargões

- Página Web consiste de objetos
- Objeto pode ser arquivo HTML, imagem JPEG, Java applet, arquivo de áudio,...
- A página Web consiste de arquivo-HTML base, que inclui vários objetos referenciados
- Cada objeto é endereçado por uma URL
- Exemplo de URL:



HTTP

HTTP: hypertext transfer protocol

- Protocolo da camada de aplicação da Web
- Modelo cliente/servidor
  - Cliente: browser que solicita, recebe e apresenta objetos da Web
  - Servidor: envia objetos em resposta a pedidos
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2068

HTTP

Utiliza TCP:

- Cliente inicia conexão TCP (cria socket) para o servidor na porta 80
- Servidor aceita uma conexão TCP do cliente
- Mensagens HTTP (mensagens do protocolo de camada de aplicação) são trocadas entre o browser (cliente HTTP) e o servidor Web (servidor HTTP)
- A conexão TCP é fechada

HTTP é "stateless"

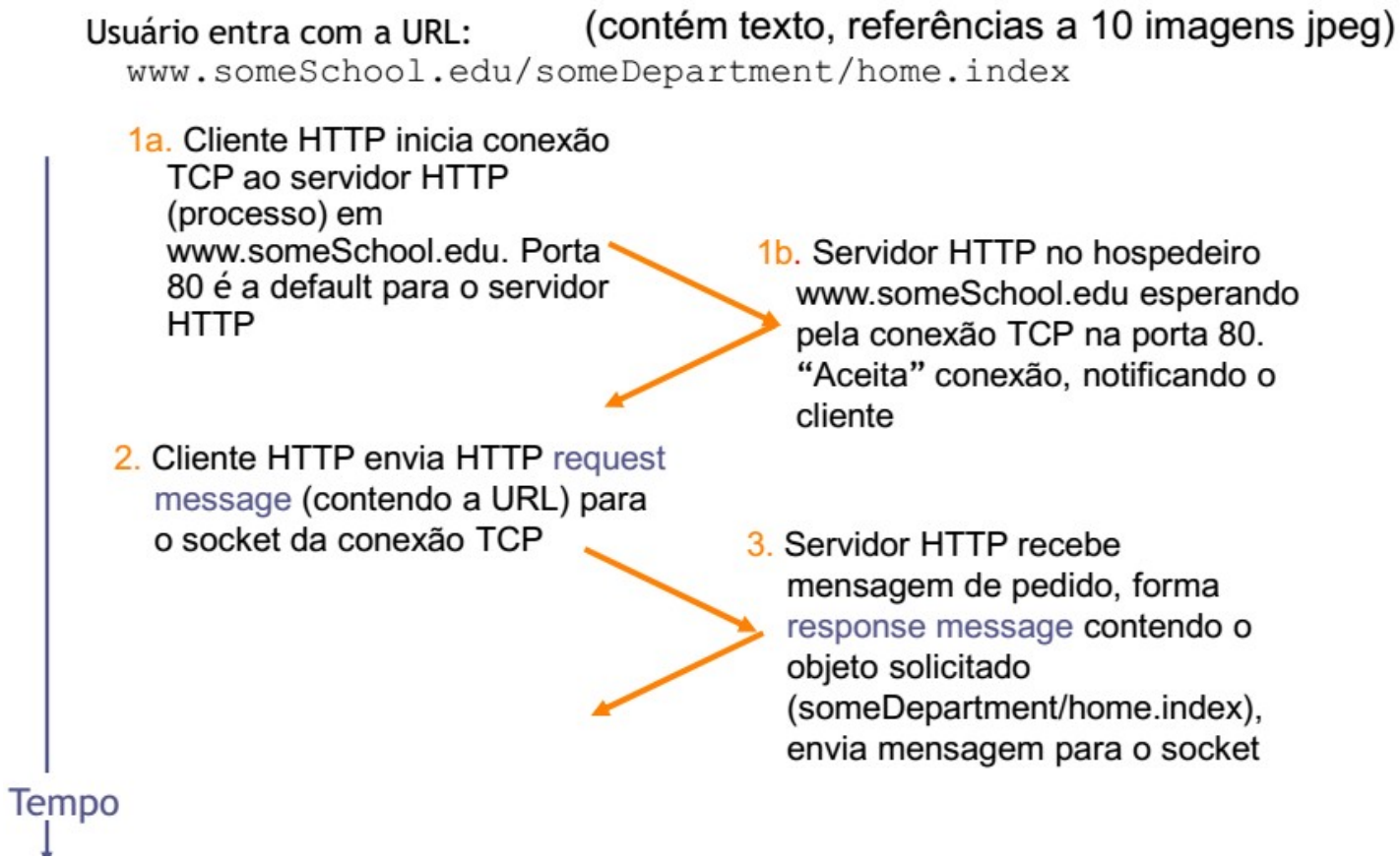
- O servidor não mantém informação sobre os pedidos passados pelos clientes

Protocolos que mantêm informações de "estado" são complexos!

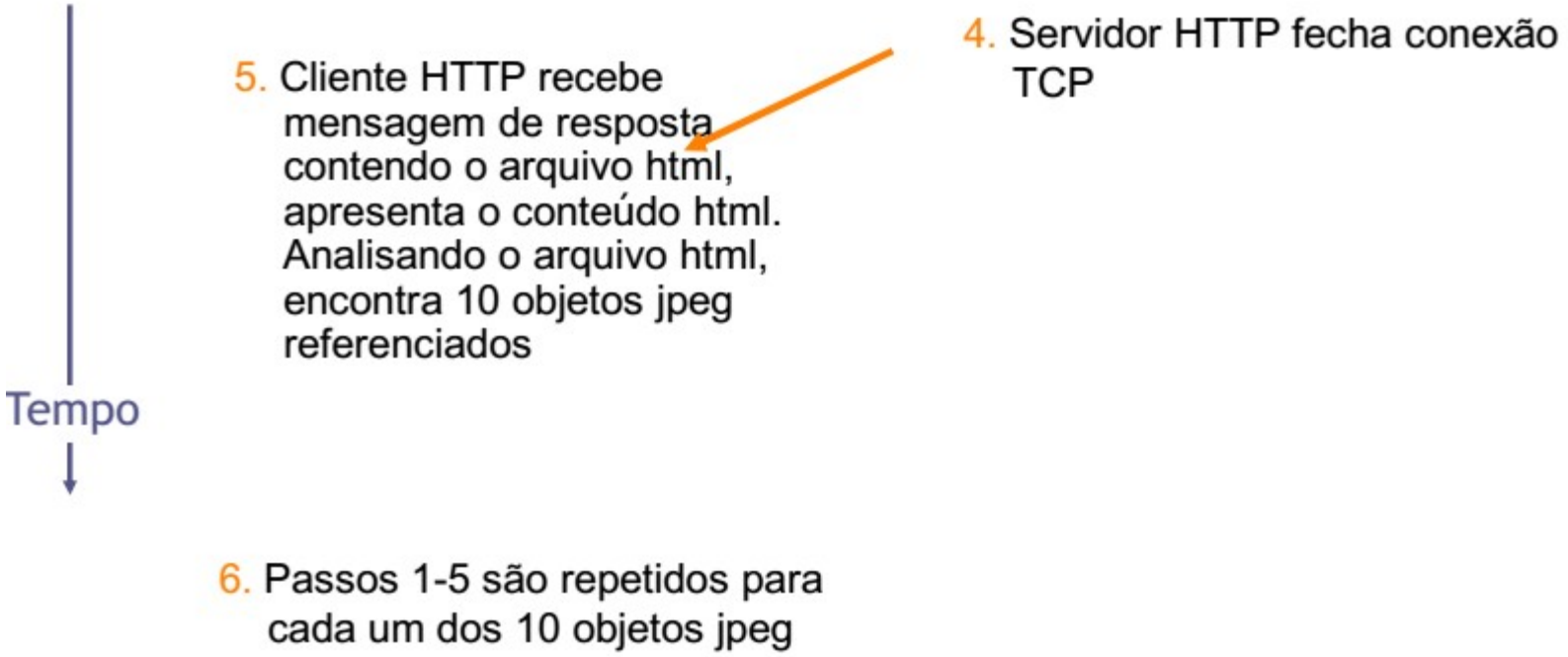
- Histórico do passado (estado) deve ser mantido
- Se o servidor/cliente quebra, suas visões de "estado" podem ser inconsistentes, devendo ser reconciliadas

HTTP não persistente

- No máximo, um objeto é enviado sobre uma conexão TCP
- O HTTP/1.0 utiliza HTTP não persistente



HTTP não persistente



HTTP não persistente – Tempo de Resposta

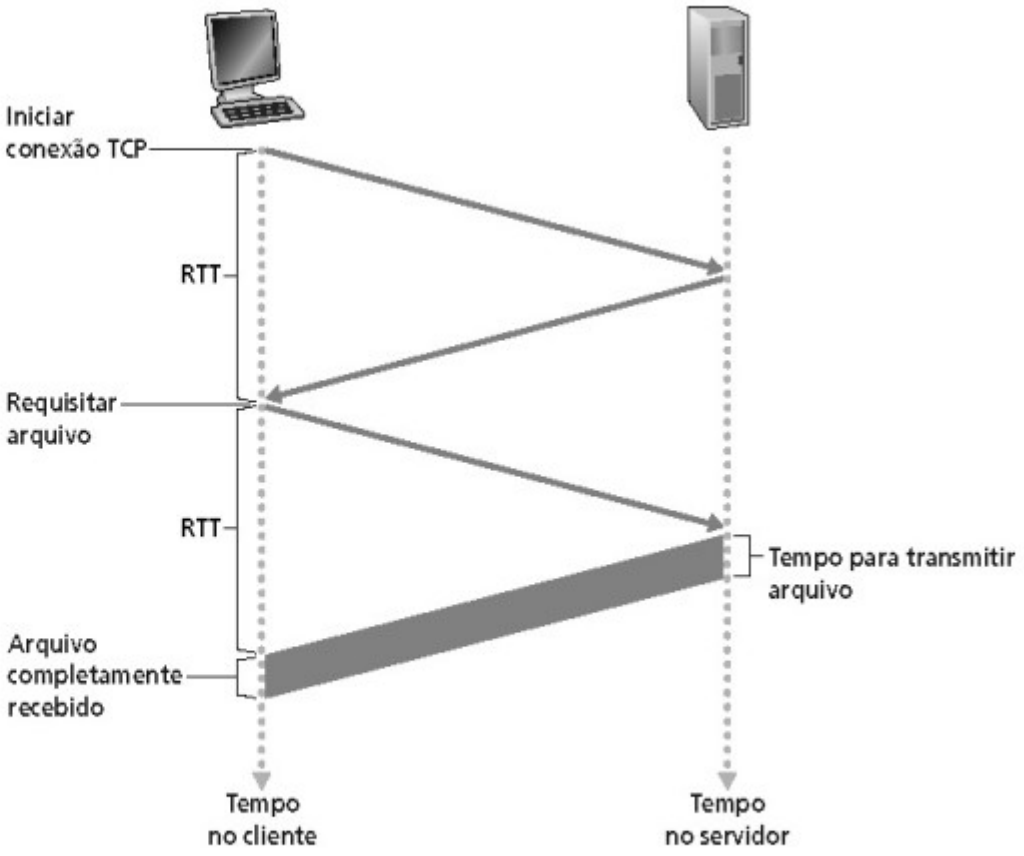
Definição de RTT: tempo para enviar um pequeno pacote que vai do cliente para o servidor e retorna

Tempo de resposta:

- Um RTT para iniciar a conexão TCP
- Um RTT para requisição HTTP e primeiros bytes da resposta HTTP para retorno
- Tempo de transmissão de arquivo

Total = 2RTT+ tempo de transmissão

RTT = Round Trip Time



HTTP persistente

Características do HTTP persistente:

- Requer 2 RTTs por objeto
- OS deve manipular e alocar recursos do hospedeiro para cada conexão TCP. Mas os browsers freqüentemente abrem conexões TCP paralelas para buscar objetos referenciados

HTTP persistente

- Servidor deixa a conexão aberta após enviar uma resposta
- Mensagens HTTP subseqüentes entre o mesmo cliente/servidor são enviadas pela conexão

Persistente sem **pipelining**:

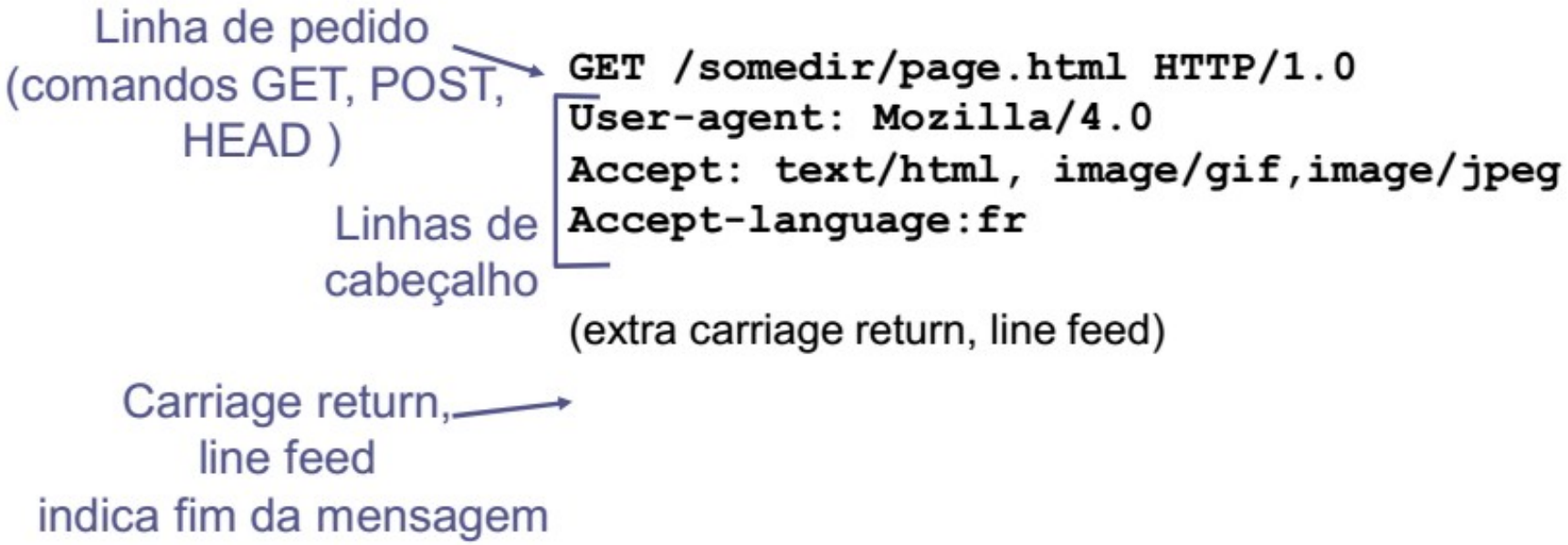
- O cliente emite novas requisições apenas quando a resposta anterior for recebida
- Um RTT para cada objeto referenciado

Persistente com **pipelining**:

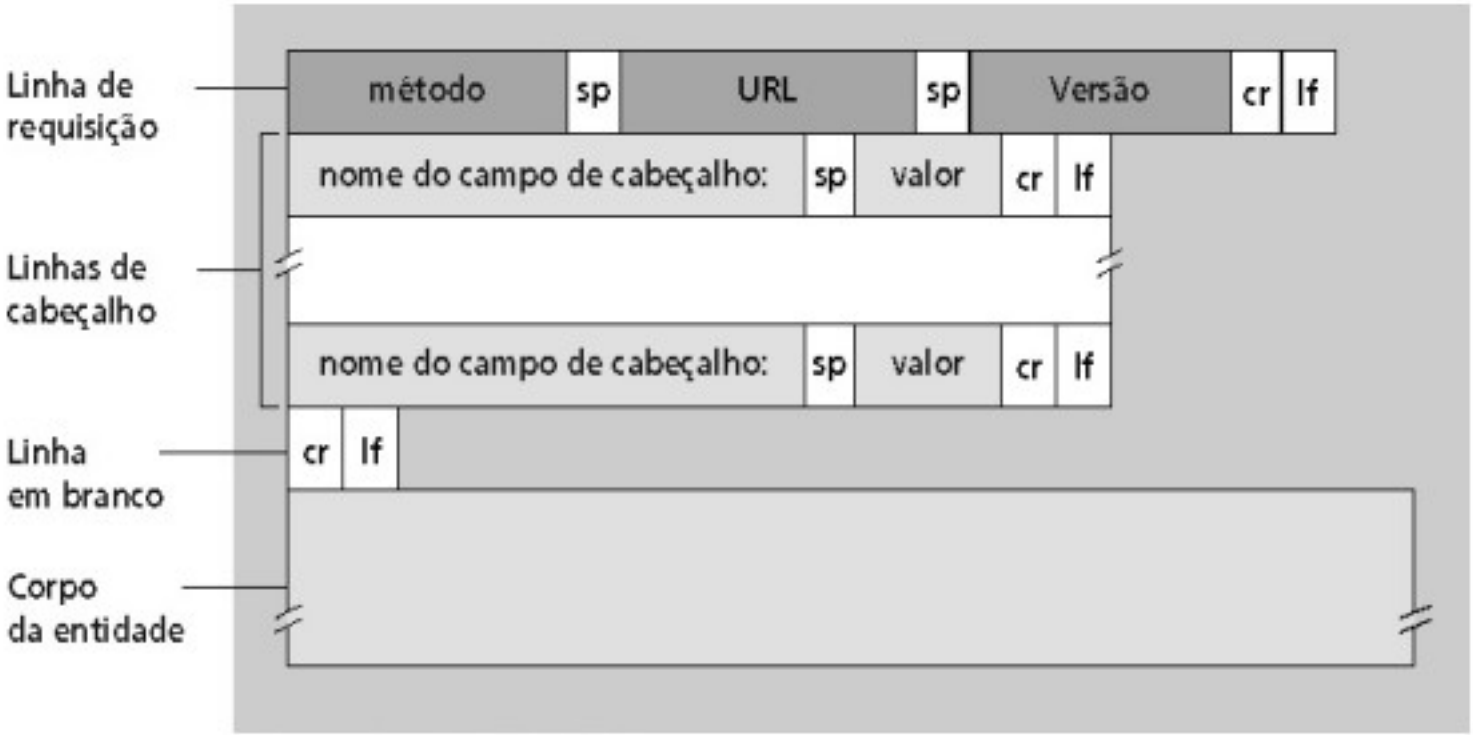
- Padrão no HTTP/1.1
- O cliente envia requisições assim que encontra um objeto referenciado
- Tão pequeno como um RTT para todos os objetos referenciados

Mensagens HTTP

- Dois tipos de mensagens HTTP: request, response
- HTTP request message:
  - ASCII (formato legível para humanos)



Mensagens HTTP



Obs.: cr = carriage return; lf = line feed

Métodos HTTP

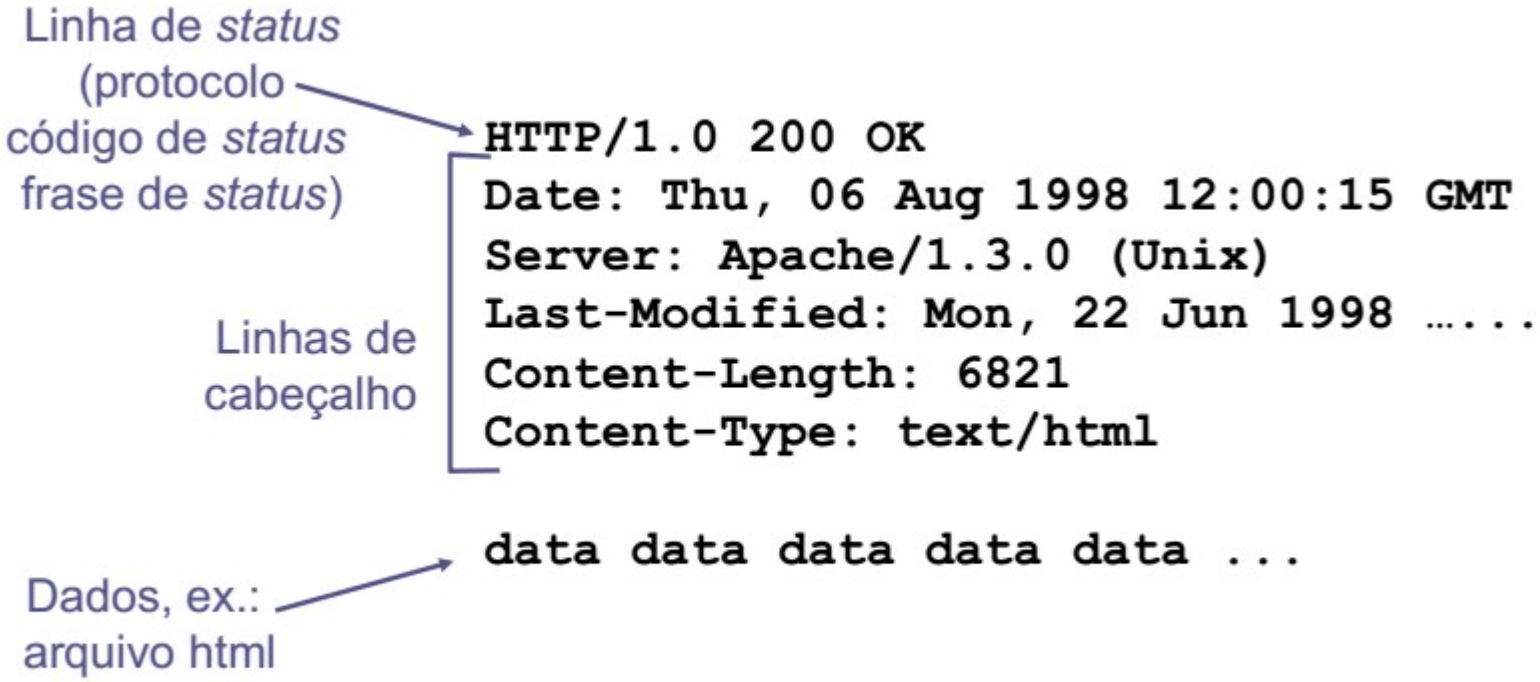
HTTP/1.0

- GET
- POST
- HEAD
  - Pede para o servidor deixar o objeto requisitado fora da resposta

HTTP/1.1

- GET, POST, HEAD
- PUT
  - Envia o arquivo no corpo da entidade para o caminho especificado no campo de URL
- DELETE
  - Apaga o arquivo especificado no campo de URL

Response HTTP



Status das respostas HTTP

Na primeira linha da mensagem de resposta servidor → cliente.
Alguns exemplos de códigos:

200 OK
- Requisição bem-sucedida, objeto requisitado a seguir nesta mensagem

301 Moved permanently
- Objeto requisitado foi movido, nova localização especificada a seguir nesta mensagem (Location:)

400 Bad request
- Mensagem de requisição não compreendida pelo servidor

404 Not Found
- Documento requisitado não encontrado neste servidor

505 HTTP version not supported

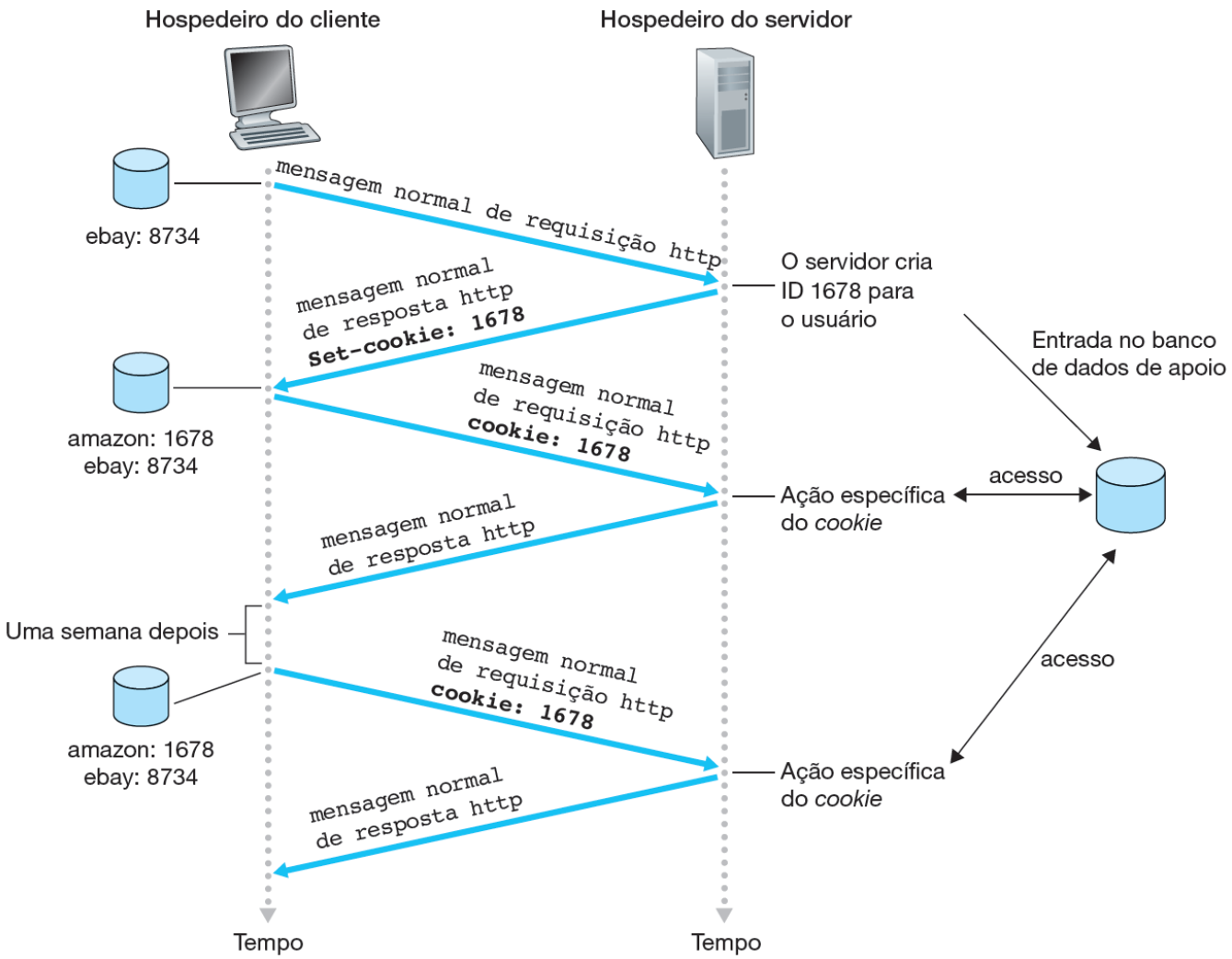
Interação usuário-servidor: cookies

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

A tecnologia dos cookies tem quatro componentes:

- Uma linha de cabeçalho de cookie na mensagem de resposta HTTP;
- Uma linha de cabeçalho de cookie na mensagem de requisição HTTP;
- Um arquivo de cookie mantido no sistema final do usuário e gerenciado pelo navegador do usuário;
- Um banco de dados de apoio no site.

Interação usuário-servidor: cookies

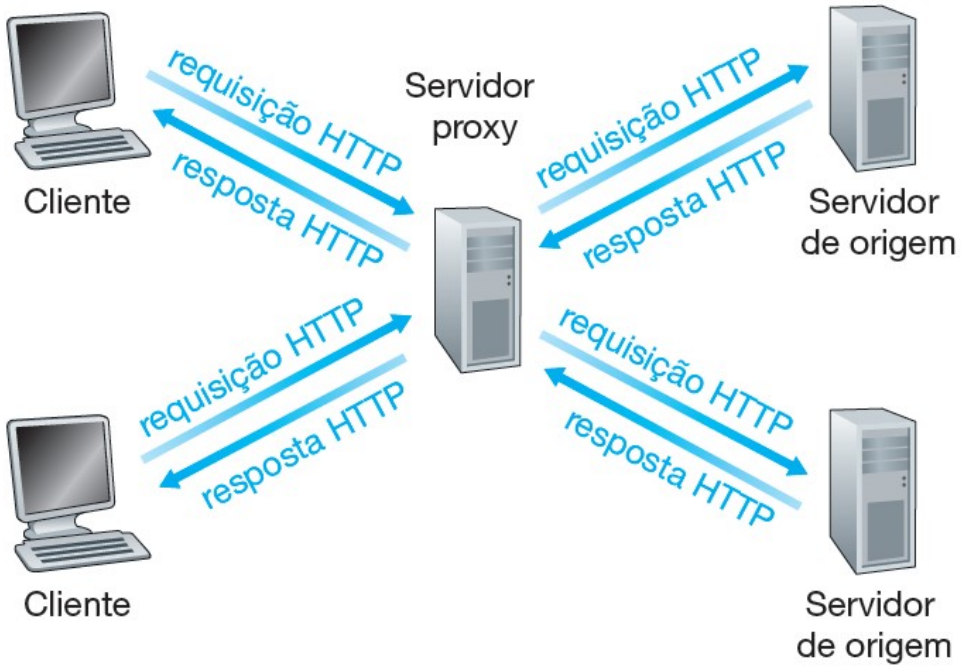


Mantendo o estado do usuário com *cookies*.

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:

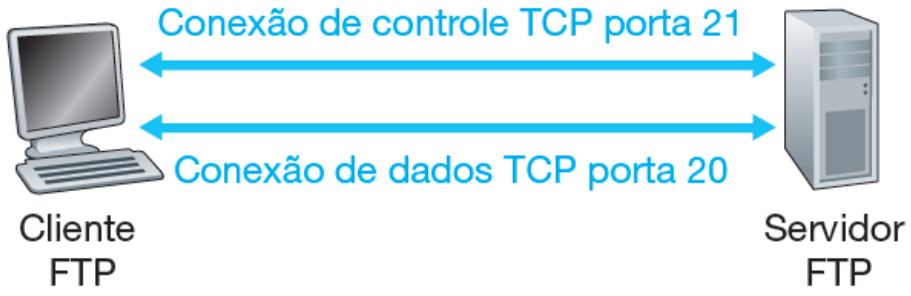


FTP – File Transfer Protocol

- Transferência de arquivos de/para o computador remoto
- Modelo cliente/servidor
  - **Cliente:** lado que inicia a transferência
  - **Servidor:** hospedeiro remoto
- FTP: RFC 959
- FTP servidor: porta 21

FTP – File Transfer Protocol

- Cliente FTP contata o servidor FTP na porta 21 especificando o TCP como protocolo de transporte
- Cliente obtém autorização pela conexão de controle
- Cliente procura o diretório remoto enviando comandos pela conexão de controle
- Quando o servidor recebe um comando para uma transferência de arquivo, ele abre uma conexão de dados TCP para o cliente
- Após a transferência de um arquivo, o servidor fecha a conexão
- Servidor abre uma segunda conexão de dados TCP para transferir outro arquivo
- Conexão de controle: "out-of-band"
- Servidor FTP mantém "estado": diretório atual, autenticação anterior



DNS – Domain Name Service

- Há duas maneiras de identificar um hospedeiro:
  - ✓ - Nome de hospedeiro
  - ✓ - 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.

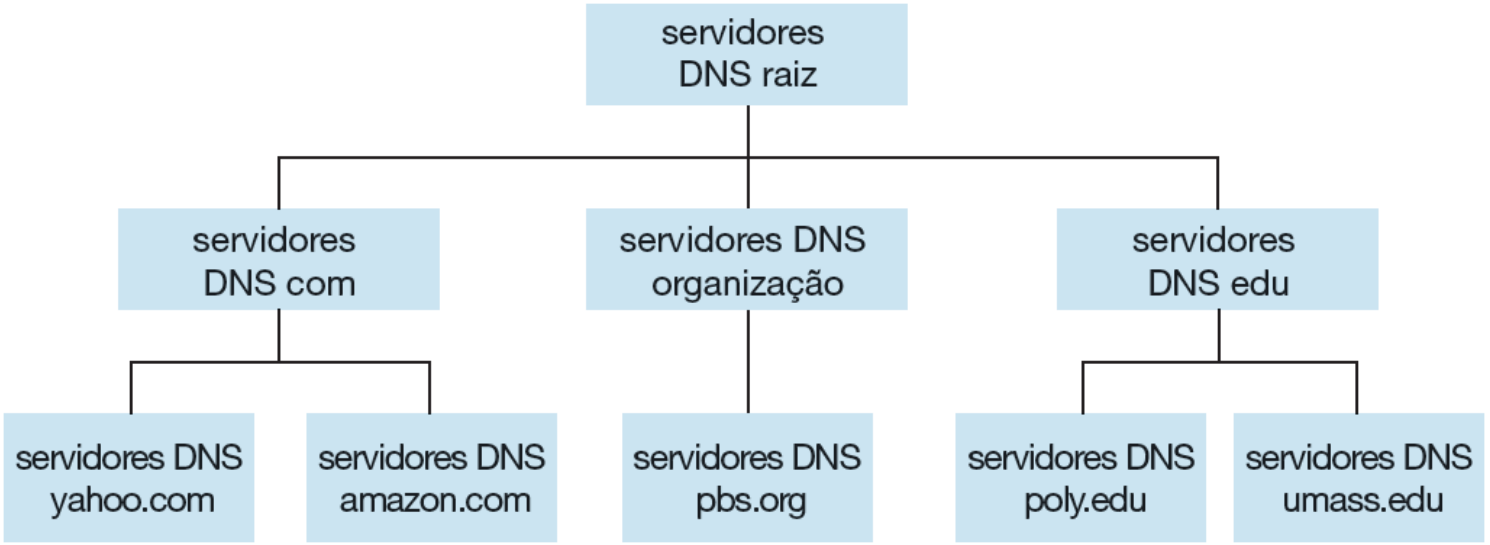
- O DNS:
  - (1) Banco de dados distribuído executado em uma hierarquia de servidores de DNS
  - (2) Protocolo de camada de aplicação que permite que hospedeiros consultem o banco de dados distribuído.

DNS – Sistema distribuído

Nenhum servidor DNS isolado tem todos os mapeamentos para todos os hospedeiros da Internet.

Em vez disso, os mapeamentos são distribuídos pelos servidores DNS.

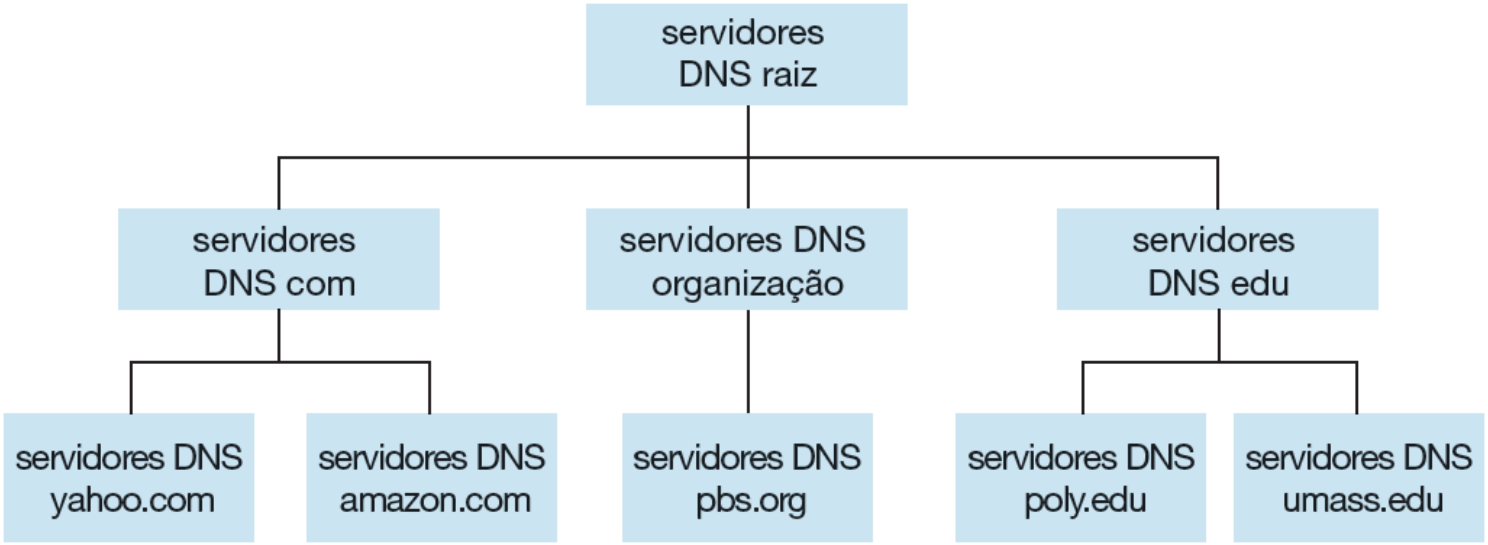
Parte da hierarquia de servidores DNS



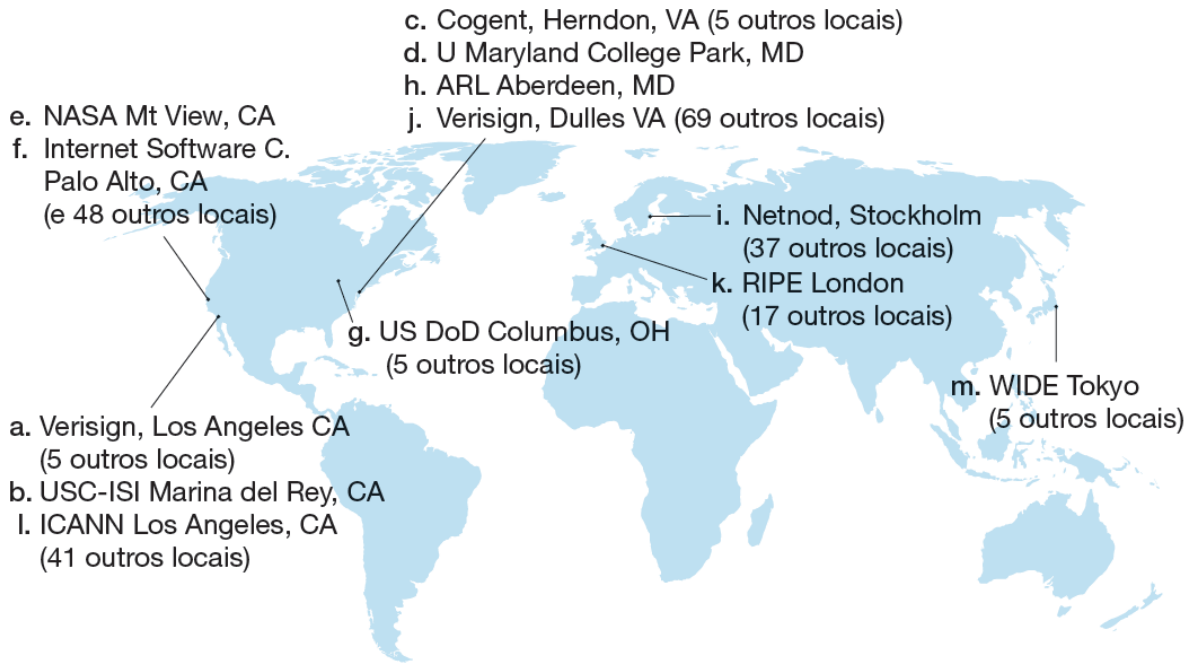
DNS – Sistema distribuído

Cliente quer o IP para www.amazon.com; 1ª aprox.:

- Cliente consulta um servidor de raiz para encontrar o servidor DNS com
- Cliente consulta o servidor DNS com para obter o servidor DNS amazon.com
- Cliente consulta o servidor DNS amazon.com para obter o endereço IP para www.amazon.com



Servidores DNS raíz



- São contatados pelos servidores de nomes locais que não podem resolver um nome
- Servidores de nomes raiz:
  - Buscam servidores de nomes autorizados se o mapeamento do nome não for conhecido
  - Conseguem o mapeamento
  - Retornam o mapeamento para o servidor de nomes local

Servidores TLD, autorizados e locais

Servidores top-level domain (TLD):

- Responsáveis pelos domínios com, org, net, edu etc. e todos os domínios top-level nacionais br, uk, fr, ca, jp.

Servidores DNS autoritativo:

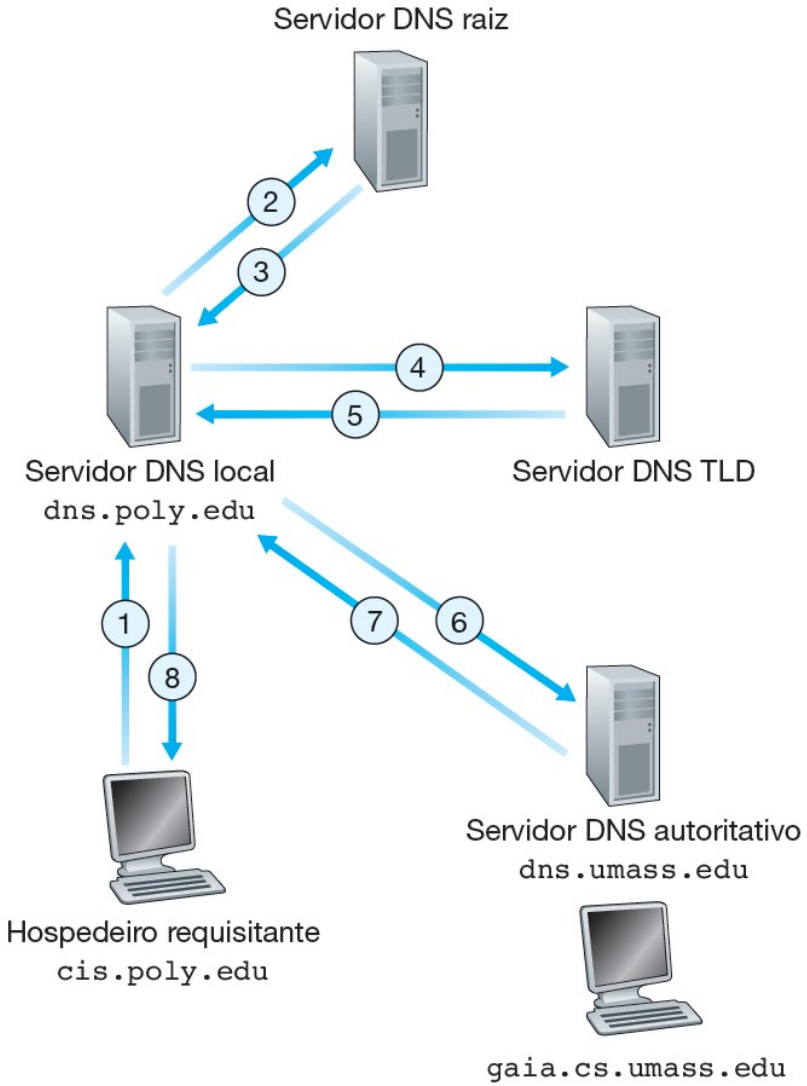
- Servidores DNS de organizações, provêem nome de hospedeiro autorizado para mapeamentos IP para servidores de organizações (ex.: Web e mail).

- Podem ser mantidos por uma organização ou provedor de serviços

Servidores DNS locais

- Não pertence estritamente a uma hierarquia
- Cada ISP (ISP residencial, companhia, universidade) possui um
- Também chamado de “servidor de nomes default”
- Quando um hospedeiro faz uma pergunta a um DNS, a pergunta é
- enviada para seu servidor DNS local
- Age como um proxy, encaminhando as perguntas para dentro da
- hierarquia

Interação dos diversos servidores DNS



Outros serviços da camada de aplicação

SMTP – Simple Mail Transfer Protocol

POP3 – Post Office Protocol

SNMP – Simple Network Management Protocol

DHCP – Dynamic Host Control Protocol

P2P – Peer to Peer

Telnet

PERGUNTAS ?