
APLICAÇÕES TCP/IP

REDES DE COMPUTADORES

Eriko Porto

eriko_porto@uol.com.br

Última atualização – agosto/2005

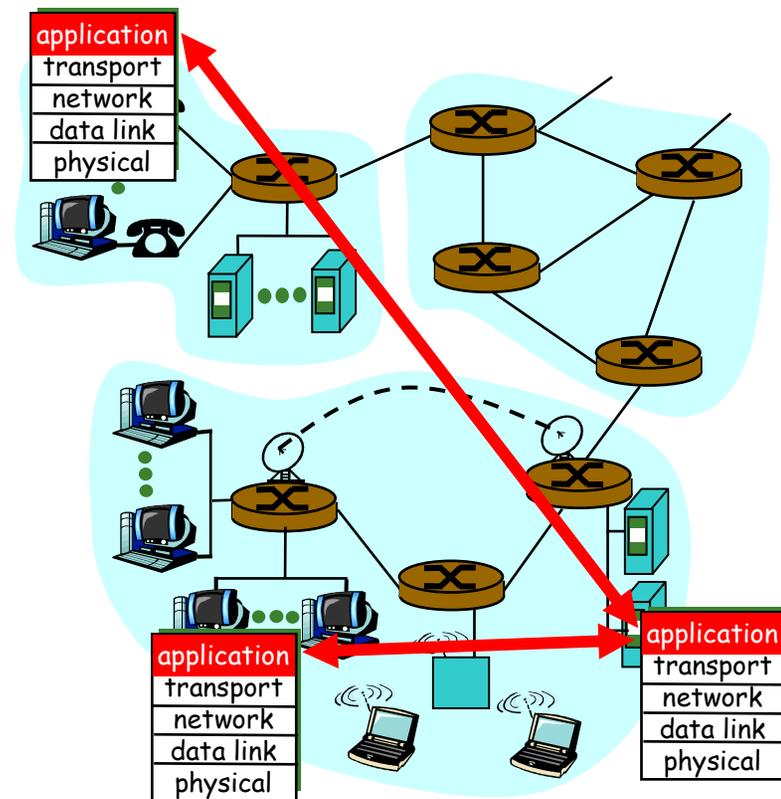
NÍVEL DE APLICAÇÃO

■ Aplicações

- ❑ comunicação entre processos distribuídos
- ❑ executam nos hosts no “espaço do usuário”
- ❑ baseadas em troca de mensagens
- ❑ exemplo: e-mail, transferência de arquivos, Web

■ Protocolos (aplicação)

- ❑ são parte da aplicação
- ❑ definem o tipo de mensagens trocadas e as ações a serem tomadas pelos processos



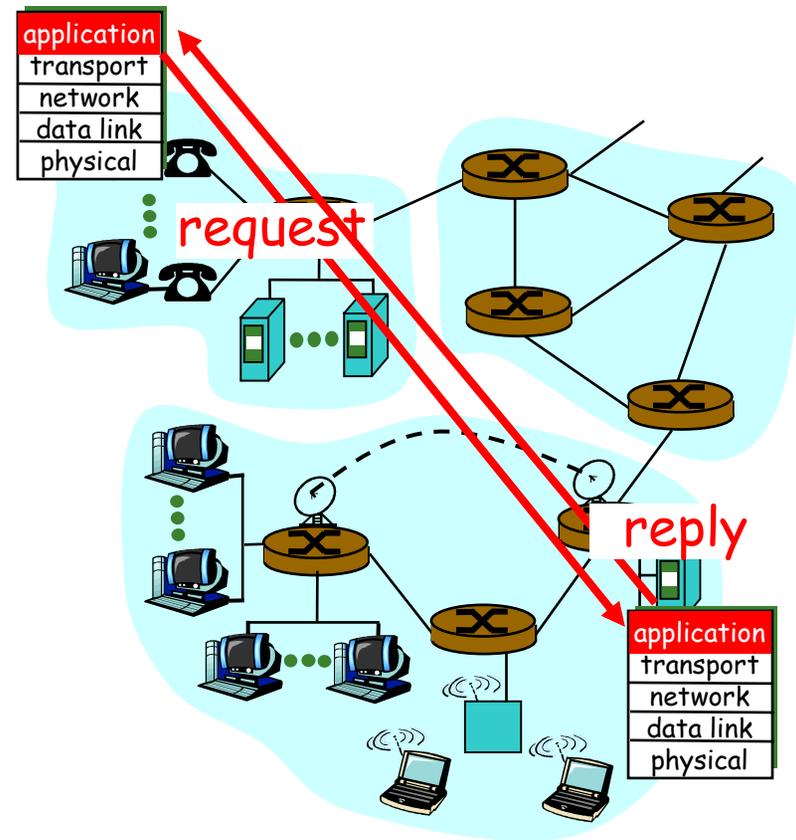
NÍVEL DE APLICAÇÃO

■ Cliente

- ❑ inicia o contato com o servidor
- ❑ geralmente requisita serviços ao servidor
- ❑ Web – o cliente é implementado no browser
- ❑ e-mail – programa leitor de correio

■ Servidor

- ❑ provê os serviços requisitados pelos clientes
- ❑ exemplo, servidor Web envia as páginas Web requisitadas, servidor de correio entrega os e-mails



APLICAÇÕES INTERNET

PROTOCOLO	APLICAÇÃO	PROTOCOLO DE TRANSPORTE
dns [RFC 1035]	domínios na Internet	TCP e UDP
http [RFC 2068]	web (www)	TCP
ftp [RFC 959]	transferência de arquivos	TCP
tftp [RFC 1350]	transferência de arquivos não confiável	UDP
smtp [RFC 821]	e-mail	TCP
pop3 [RFC 1225]	e-mail	TCP
imap [RFC 1064]	e-mail	TCP

APLICAÇÕES INTERNET

PROTOCOLO	APLICAÇÃO	PROTOCOLO DE TRANSPORTE
telnet [RFC 854]	acesso a um terminal remoto	TCP
nfs [RFC 1094]	acesso a um sistema de arquivos	TCP
snmp [RFC 1067]	gerência de equipamentos	UDP
bootp [RFC 959]	obtenção de endereços IP	UDP
dhcp [RFC 1531]	obtenção de configuração TCP/IP	UDP
nntp [RFC 977]	grupos de discussão	TCP

DOMÍNIOS

- Números IP representam uma forma compacta e eficiente de endereçamento
 - Entretanto, usuários em geral preferem fazer referência a nomes, ao invés de números
 - Os nomes devem ser preferencialmente curtos e mnemônicos
 - É necessário, portanto, estruturar um esquema de correspondência entre nomes e números IP
-

DOMÍNIOS

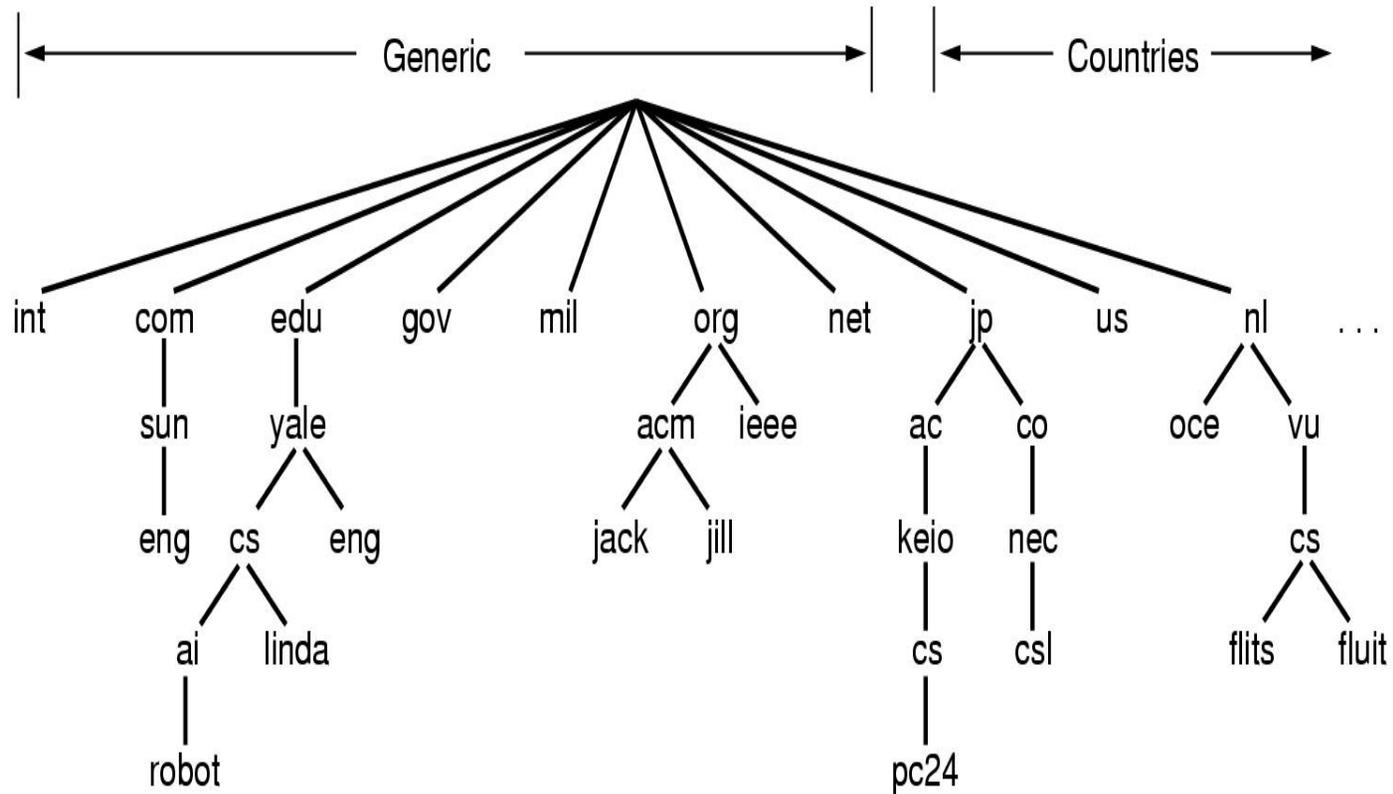
- Assim como no esquema de endereçamento IP, não pode haver nomes duplicados dentro da internet
 - Portanto a atribuição de nomes também deve ser centralizada por uma autoridade única
 - Entretanto, a escolha de nomes não duplicados é um processo mais difícil de ser realizado de que a atribuição de números IP
 - É preciso definir como será feita esta escolha de nomes
-

DOMÍNIOS

- A solução é a adoção de uma estrutura hierárquica de nomes
 - O topo da escala hierárquica é dividido em partições, cada uma delas responsável pela atribuição de nomes dentro de seu contexto
 - Cada partição pode ser subdividida em novas partições, caso haja necessidade
-

DOMÍNIOS

- Hierarquia de domínios na Internet



DOMÍNIOS

- No exemplo anterior, o nome completo do nó *robot* seria ***robot.ai.cs.yale.edu***
 - A sintaxe utilizada especifica que cada subdivisão deve ser separada por um ponto das demais
 - Quanto mais a direita no nome, mais global é a subdivisão
 - O topo é administrado pelo InterNIC, e cada partição é administrada por uma instituição, com poderes delegados pelo topo
-

DOMÍNIOS

- A estrutura hierárquica de nomes, ou domínio, reflete a estrutura organizacional das instituições que fazem parte da internet, e não a estrutura física das redes que as interconectam
 - Em geral, temos até quatro níveis na estrutura hierárquica a partir do topo
 - Apenas pela sintaxe, não é possível distinguir um subdomínio de um nó
-

DOMÍNIOS

- Exemplos de domínios globais na Internet

COM	organizações comerciais
EDU	instituições educacionais
GOV	instituições governamentais
MIL	grupos militares
NET	centros de suporte à rede
ORG	demais organizações
ARPA	domínio Arpanet
xy	duas letras representando um país

DOMÍNIOS

■ Exemplos

- ❑ gatekeeper.dec.com
 - ❑ nic.cerf.net
 - ❑ ceop1.rederio.br
 - ❑ ipanema.nce.ufrj.br
 - ❑ netlib.org
 - ❑ nic.ddn.mil
 - ❑ wuarchive.wustl.edu
 - ❑ latusi.org.uy
-

DOMÍNIOS

- No protocolo IP, o roteamento é todo baseado nos endereços IP, e não nos nomes
 - Torna-se necessário um mecanismo de tradução de nomes em números IP
 - O mecanismo deve ser transparente para o usuário, e feito de forma automática
-

DOMÍNIOS

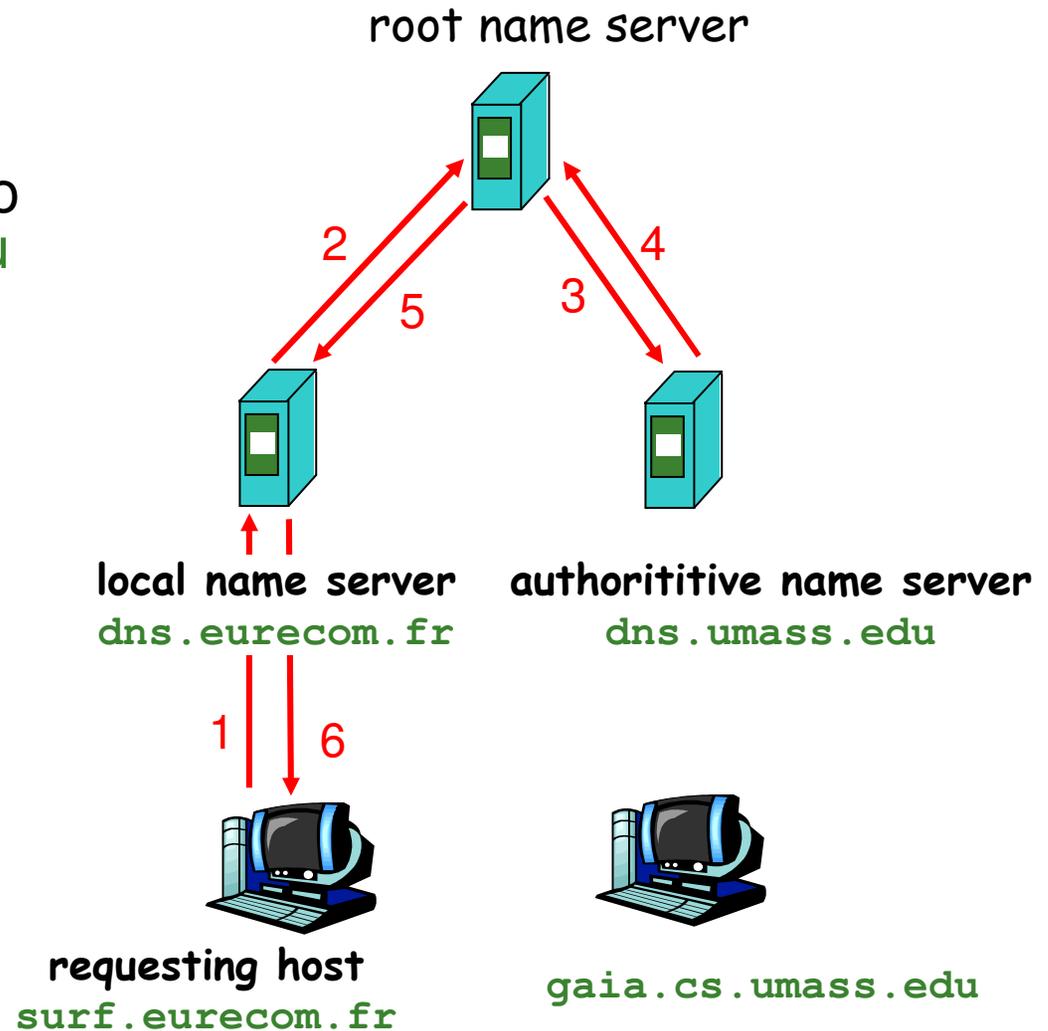
- Servidor de nomes (name server)
 - ❑ mecanismo usado na internet para a tradução de nomes em endereços IP
 - ❑ mecanismo distribuído – conjunto de máquinas cooperativas operando em diferentes lugares
 - ❑ mecanismo eficiente – a tradução é feita geralmente em modo local, só exigindo tráfego na rede quando necessário
 - ❑ mecanismo confiável – não há um único ponto de falha que impeça todo o sistema de funcionar
-

DOMÍNIOS

- Servidor de nomes (name server)
 - Os servidores de nomes se estruturam hierarquicamente, do mesmo modo que a estrutura de nomes da internet
 - Cada servidor é responsável pela tradução de nomes dentro de seu domínio, e contata os demais quando precisa traduzir nomes fora do seu contexto
 - O programa cliente que consulta um servidor de nomes é chamado resolvedor
 - Em uma máquina UNIX, cada servidor de nomes realiza a conversão nome-endereço através da execução de um daemon chamado in.named
-

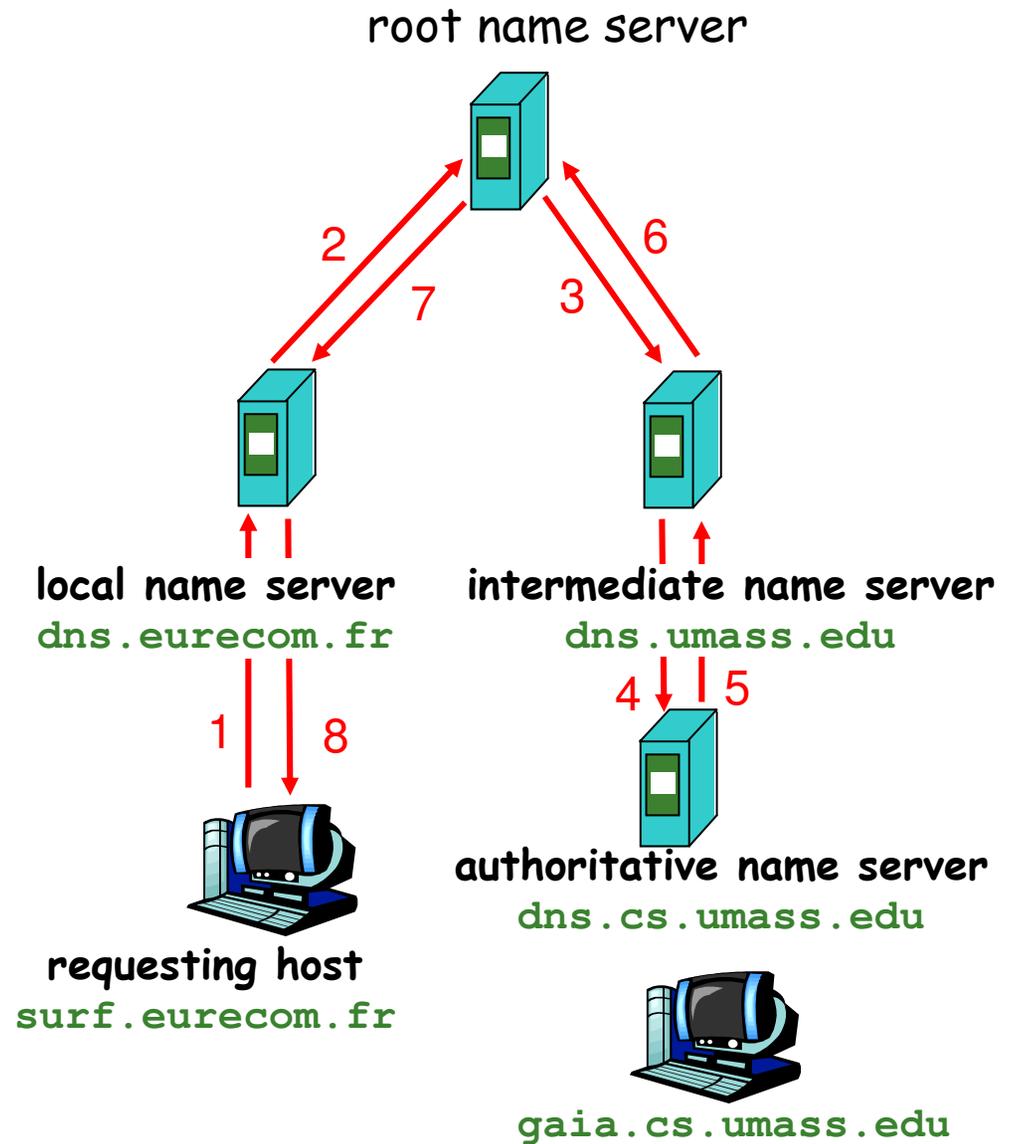
DOMÍNIOS

- O host `surf.eurecom.fr` quer resolver o endereço IP de `gaia.cs.umass.edu`
 - contata o servidor DNS local `dns.eurecom.fr`
 - `dns.eurecom.fr` contata o root name server, caso necessário
 - root name server contata o servidor autoritativo, `dns.umass.edu`, caso necessário



DOMÍNIOS

- Root name server
 - pode não conhecer um servidor autoritativo
 - conhece um servidor intermediário (intermediate name server) que pode ser contatado para encontrar o servidor autoritativo

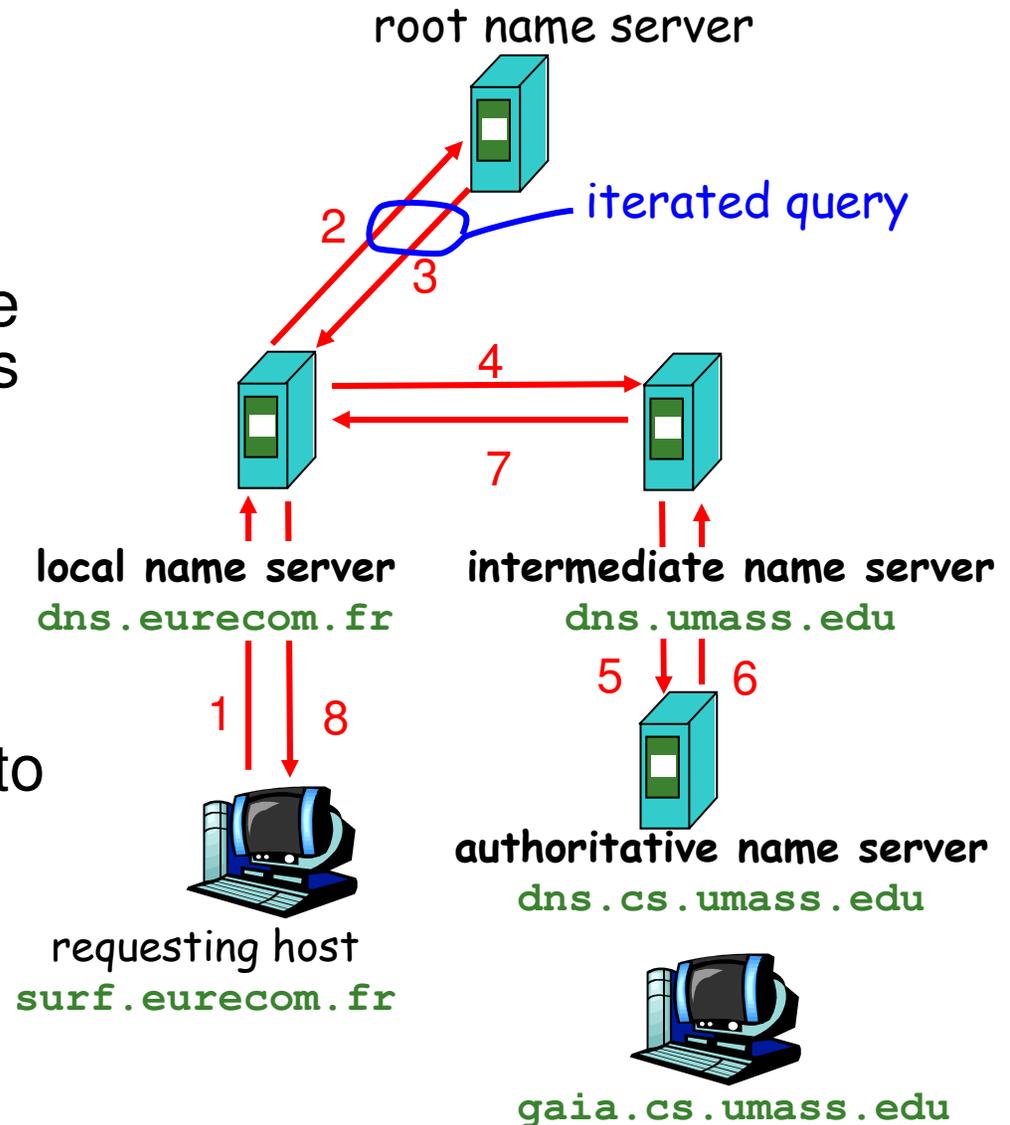


DOMÍNIOS

- A tradução de nomes se processa de cima para baixo, começando pelo servidor raiz, descendo depois para os servidores nas folhas da árvore.
 - Um servidor de nomes pode realizar dois tipos de tradução
 - **RECURSIVA** – o servidor vai contatando todos os demais até achar a tradução completa
 - **ITERATIVA** – se o servidor não puder fazer a tradução dentro do seu contexto, ele devolve o nome do servidor que pode traduzir
-

DOMÍNIOS

- consulta recursiva
 - coloca o trabalho de resolução de nomes no servidor local
- consulta iterativa
 - servidor contatado responde com o nome do próximo servidor para contato



DOMÍNIOS

- Um servidor de nomes pode manter dois tipos de dados
 - AUTORITATIVO
 - dados obtidos diretamente pelo servidor, que são atualizados periodicamente
 - NÃO-AUTORITATIVO
 - dados obtidos indiretamente (através de outro servidor)
 - podem estar incompletos ou desatualizados
-

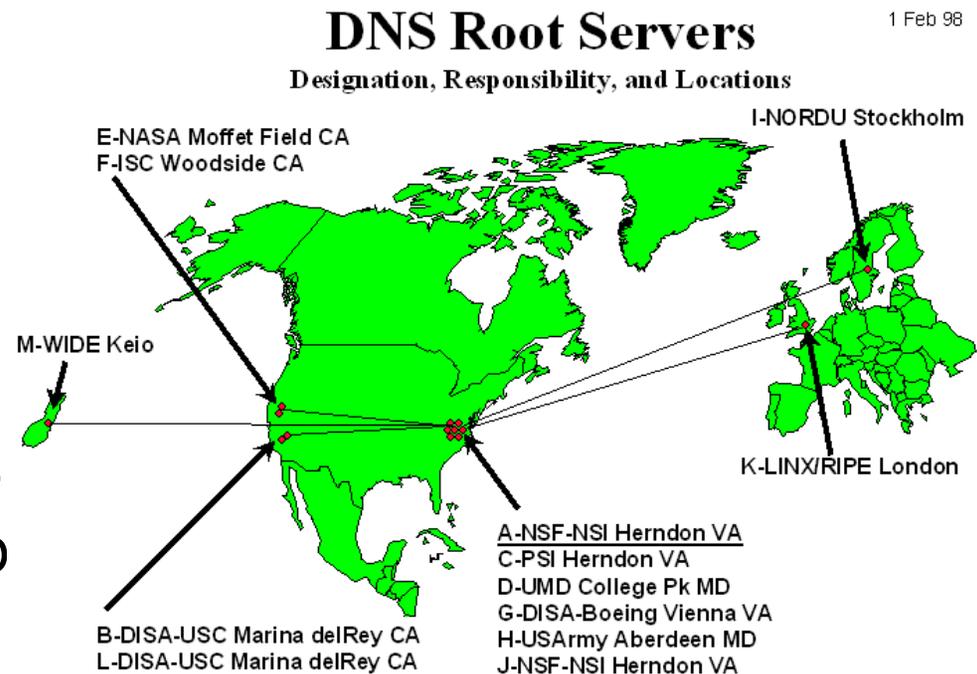
DOMÍNIOS

- Cada resolvedor precisa saber previamente o endereço IP de pelo menos um servidor de nomes
 - Cada servidor de nomes precisa saber previamente os endereços IP dos servidores raiz
 - O InterNIC mantém cerca de sete servidores raiz
-

DOMÍNIOS

■ Root Name Server

- ❑ contatado pelo servidor local que não consegue resolver o nome
- ❑ contata o servidor autoritativo
- ❑ recebe a resolução
- ❑ retorna a resolução para o servidor de nomes local
- ❑ apenas alguns servidores raiz ao redor do mundo



DOMÍNIOS

- Para evitar tráfegos desnecessários na rede, cada servidor tem um cache, guardando as traduções realizadas mais recentemente
 - Informações no cache podem ficar obsoletas, precisando ser atualizadas periodicamente a partir do servidor autoritativo
 - O usuário pode opcionalmente consultar diretamente o servidor autoritativo
 - O tempo de atualização periódica do cache é configurável (em geral 1 dia)
-

DOMÍNIOS

- Master Servers

- ❑ cada zona deve ter pelo menos 2 *master servers* que mantêm todos os dados correspondentes a essa zona
 - ❑ portanto, os dados de uma zona devem estar disponíveis em pelo menos 2 servidores
 - ❑ um deles deve ser designado como primário e pelo menos um servidor adicional como secundário, que serve como backup caso aconteça algum problema com o primário
-

DOMÍNIOS

- Servidor Primário

- ❑ servidor de nomes onde são feitas as mudanças relativas à zona
 - ❑ envia dados e delega sua autoridade ao servidor secundário
 - ❑ carrega a cópia de seus dados do disco quando o `in.named` é executado (UNIX)
 - ❑ pode delegar autoridade para outros servidores dentro ou fora de sua zona
-

DOMÍNIOS

- Servidor Secundário
 - ❑ servidor de nomes que mantém uma cópia dos dados de uma zona
 - ❑ quando o `in.named` é executado, ele pede todos os dados da respectiva zona para o primário
 - ❑ checa periodicamente com o primário para verificar se precisa atualizar seus dados
-

DNS

- **Domain Name System**

- ❑ Protocolo de aplicação usado para comunicação entre servidor e cliente
 - ❑ Utiliza UDP porta 53 para solicitações de resolução de nomes a partir do cliente (resolvedor)
 - ❑ Utiliza TCP porta 53 para sincronização entre servidores
 - ❑ Mensagens *query* e *reply* com mesmo formato
-

DNS

- Cabeçalho da mensagem
- **Identification**
 - ❑ número de 16 bits que identifica o query e o reply
- **flags**
 - ❑ query or reply
 - ❑ recursion desired
 - ❑ recursion available
 - ❑ reply is authoritative

identification	flags
number of questions	number of answer RRs
number of authority RRs	number of additional RRs
questions (variable number of questions)	
answers (variable number of resource records)	
authority (variable number of resource records)	
additional information (variable number of resource records)	



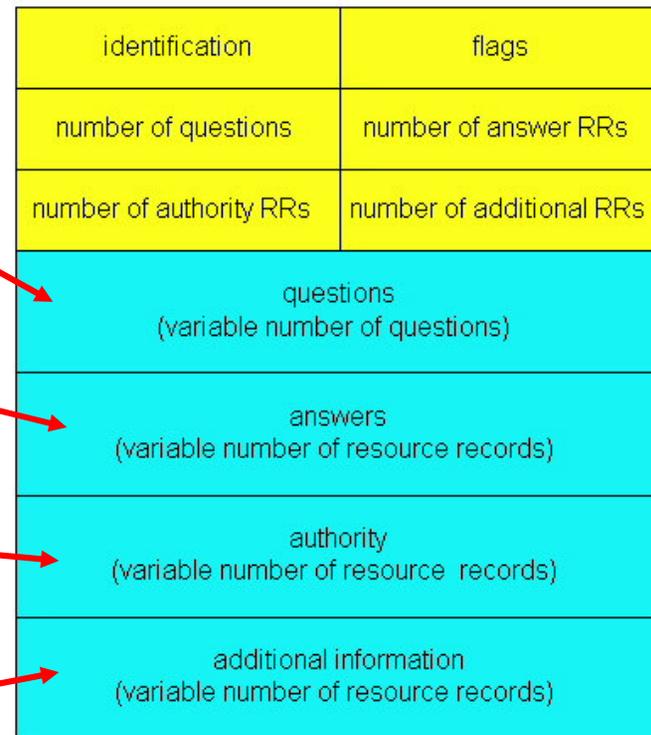
DNS

Nome, campo de tipo
para uma consulta

RRs da resposta
à consulta

registros para
servidores autoritativos

Informações adicionais
que podem ser úteis



↑
12 bytes
↓

DNS

DNS – BD distribuído armazenando *resource records* (RRs)

formato do RR - (name, value, type, ttl)

- Type = A
 - name – hostname
 - value – endereço IP
- Type = NS
 - name – domínio (exemplo foo.com)
 - value - endereço IP do servidor de nomes autoritativo do domínio
- Type = CNAME
 - name – um *alias* para o nome “canônico” (real)
 - value – nome canônico
- Type = MX
 - value – hostname do servidor de correio associado à name

DNS

Parte do BD de um servidor DNS

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA  star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.      86400  IN  TXT  "Divisie Wiskunde en Informatica."
cs.vu.nl.      86400  IN  TXT  "Vrije Universiteit Amsterdam."
cs.vu.nl.      86400  IN  MX   1 zephyr.cs.vu.nl.
cs.vu.nl.      86400  IN  MX   2 top.cs.vu.nl.

flits.cs.vu.nl. 86400  IN  HINFO Sun Unix
flits.cs.vu.nl. 86400  IN  A    130.37.16.112
flits.cs.vu.nl. 86400  IN  A    192.31.231.165
flits.cs.vu.nl. 86400  IN  MX   1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX   2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX   3 top.cs.vu.nl.
www.cs.vu.nl.   86400  IN  CNAME star.cs.vu.nl
ftp.cs.vu.nl.   86400  IN  CNAME zephyr.cs.vu.nl

rowboat                IN  A    130.37.56.201
                        IN  MX   1 rowboat
                        IN  MX   2 zephyr
                        IN  HINFO Sun Unix

little-sister          IN  A    130.37.62.23
                        IN  HINFO Mac MacOS

laserjet               IN  A    192.31.231.216
                        IN  HINFO "HP Laserjet IIISi" Proprietary
```

WWW – WORLD WIDE WEB

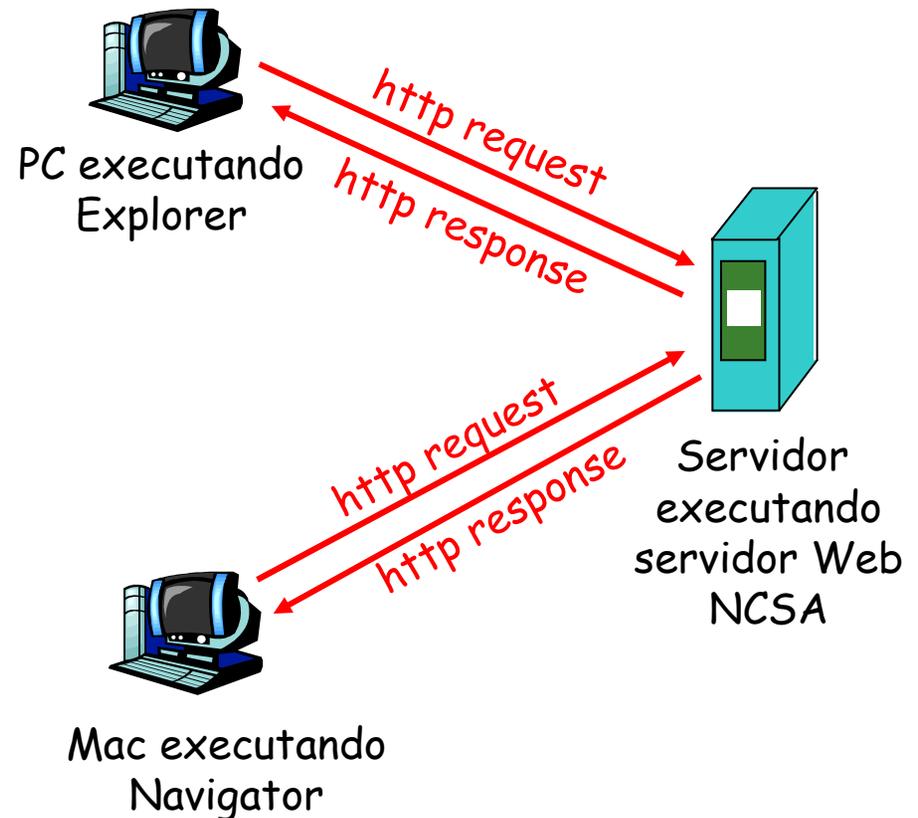
- Até a década de 90 a Internet era usada basicamente por pesquisadores, acadêmicos e estudantes universitários
 - No início da década de 90, entra em cena a aplicação chave da Internet – a WWW (World Wide Web)
 - A proposta inicial para uma teia de documentos vinculados veio de um físico do CERN (Centro Europeu para pesquisas nucleares), Tim Berners-Lee
-

WWW – WORLD WIDE WEB

- Rapidamente o desenvolvimento das aplicações WWW (browsers e servidores) chamou a atenção do público em geral
 - As aplicações Web elevaram a Internet do nível que esta ocupava, como apenas mais uma rede de dados, para o nível atual, de ser essencialmente a única rede de dados
 - A Internet está provocando uma drástica transformação na maneira como as pessoas interagem dentro e fora do seu ambiente de trabalho
-

HTTP

- **Hypertext Transfer Protocol**
 - protocolo da camada de aplicação utilizado na Web
 - modelo cliente/servidor
 - cliente – browser que requisita, recebe, e mostra objetos Web
 - servidor – servidor Web que envia objetos em resposta às requisições
- HTTP 1.0 – RFC 1945
- HTTP 1.1 – RFC 2068



HTTP

■ HTTP – utiliza o TCP

- ❑ cliente inicia conexão TCP (cria o socket) para o servidor na porta 80
- ❑ servidor aceita a conexão TCP do cliente
- ❑ mensagens HTTP são trocadas entre o browser (cliente HTTP) e o servidor Web (servidor HTTP)
- ❑ conexão TCP é finalizada

■ A conexão HTTP é “stateless”

- ❑ o servidor não mantém informações sobre as requisições anteriores do cliente
 - ❑ protocolos que mantêm informações de estado são complexos
 - ❑ estados devem ser armazenados
 - ❑ se a conexão falhar pode ocorrer inconsistências
-

HTTP

■ Conexão não-persistente

- HTTP/1.0
- servidor recebe a requisição, responde, e fecha a conexão TCP
- 2 RTTs para visualizar cada objeto
- cada transferência do browser enfrenta o slow start do TCP

■ Conexão persistente

- default para o HTTP/1.1
 - na mesma conexão TCP o servidor recebe a requisição, responde, e atende novas requisições
 - o cliente envia novas requisições para os objetos referenciados, tão logo recebe o documento base HTML
 - menos RTTs e slow starts
-

HTTP

- Mensagem HTTP *request* - ASCII

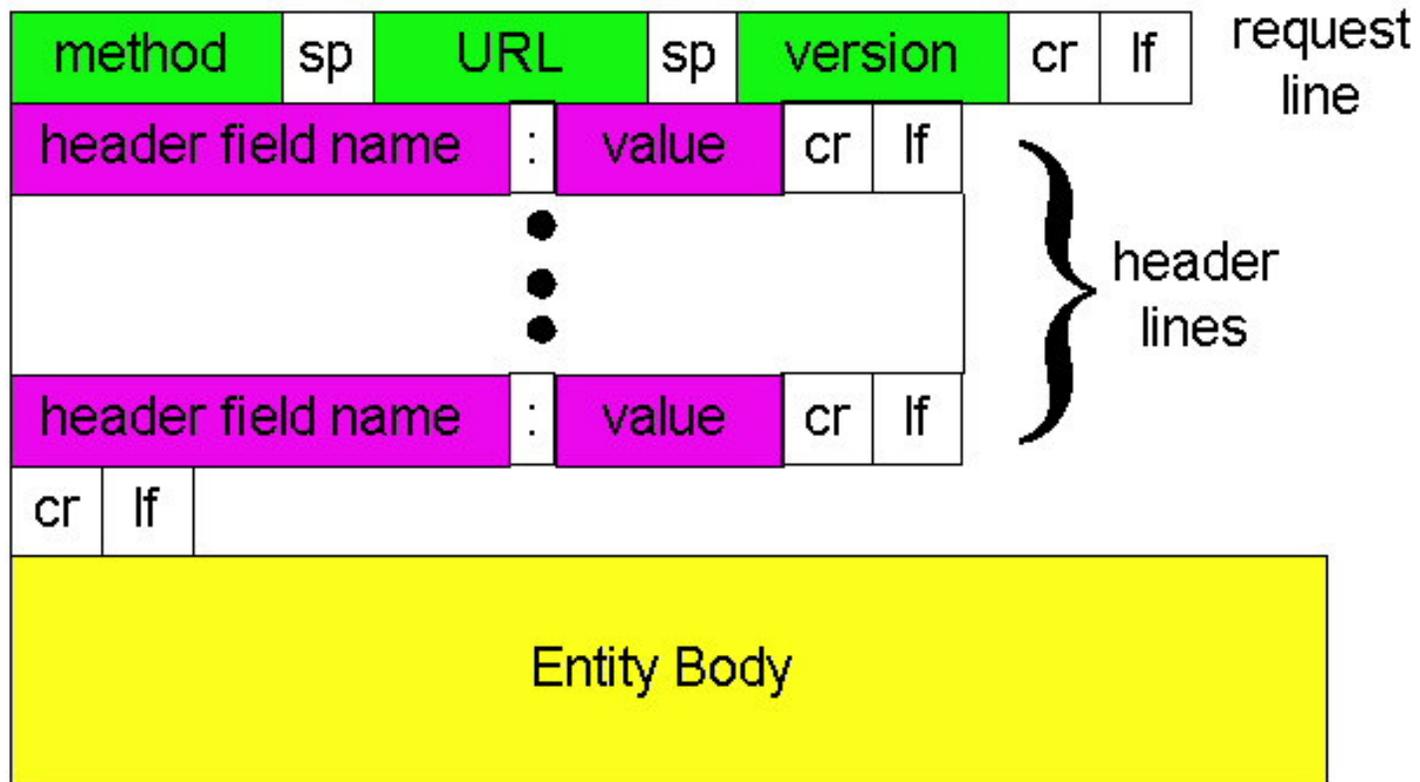
request line
(comandos GET, POST, HEAD) → GET /somedir/page.html HTTP/1.0

linhas do cabeçalho → User-agent: Mozilla/4.0
Accept: text/html, image/gif
Accept-language: fr

carriage return, line feed → (extra carriage return, line feed)
indicam fim da mensagem

HTTP

- Mensagem HTTP *request*



HTTP

- Mensagem HTTP *response* - ASCII

linha de status
(código de status
do protocolo e
comentário
do status)

linhas do
cabeçalho

```
HTTP/1.0 200 OK
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html
```

data, ex.,
arquivo html
requisitado

```
data data data data data ...
```

HTTP

■ Códigos de status

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

400 Bad Request

- request message not understood by server

404 Not Found

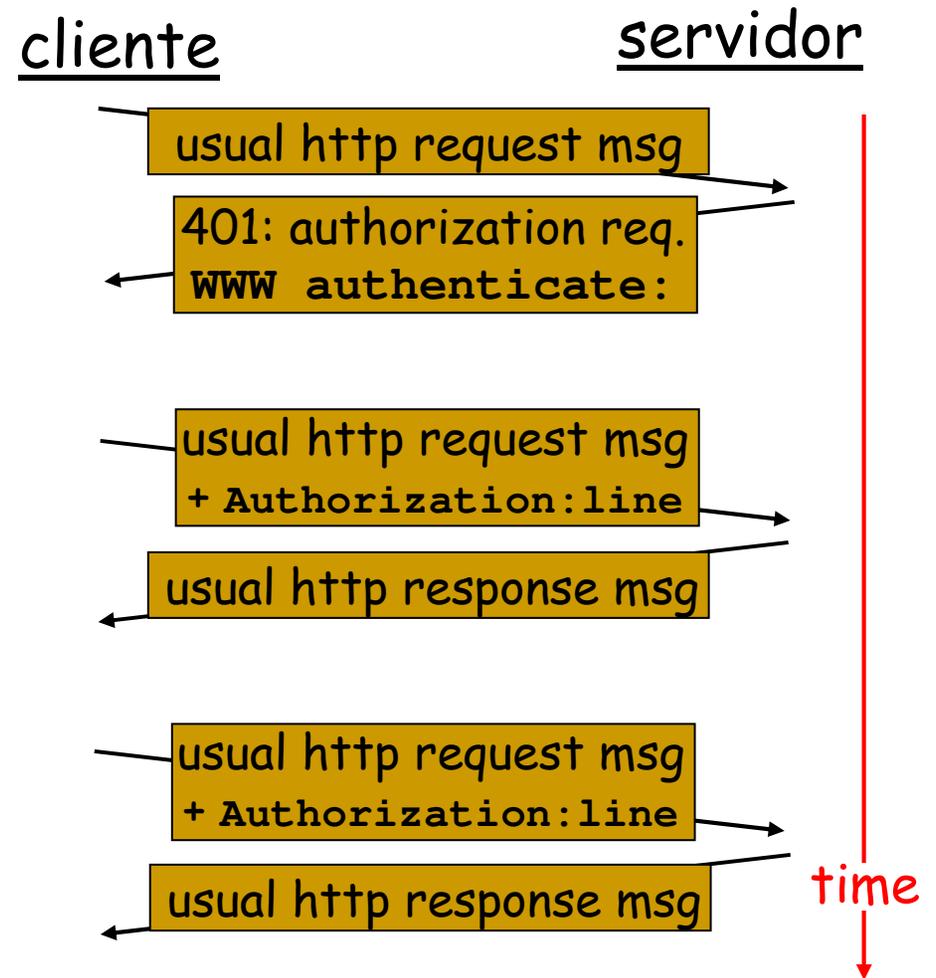
- requested document not found on this server

505 HTTP Version Not Supported

HTTP

■ Autenticação

- controle de acesso aos documentos do servidor
- o cliente deve apresentar autorização para cada requisição (stateless)
- Autorização – usualmente nome e senha



HTTP

■ Cookies

- servidor envia cookie para o cliente em resposta à requisição

Set-cookie: 1678453

- cliente apresenta o cookie nas requisições posteriores

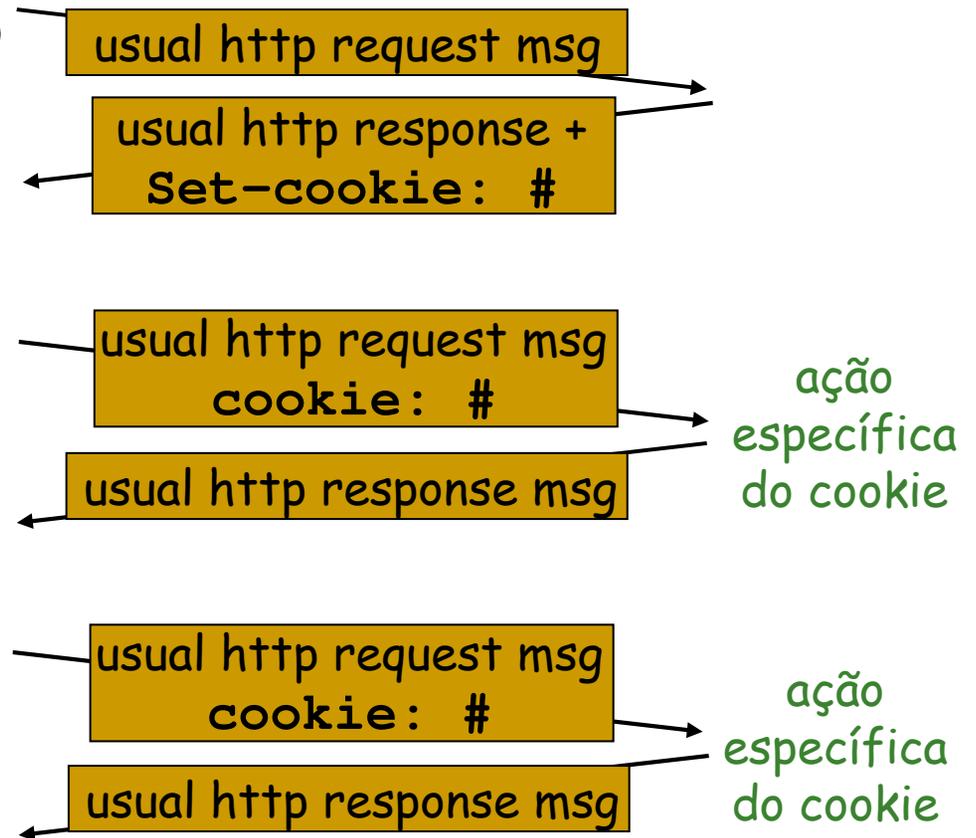
cookie: 1678453

- servidor compara o cookie apresentado com as informações do banco de dados

- autenticação
- preferências do usuário

cliente

servidor



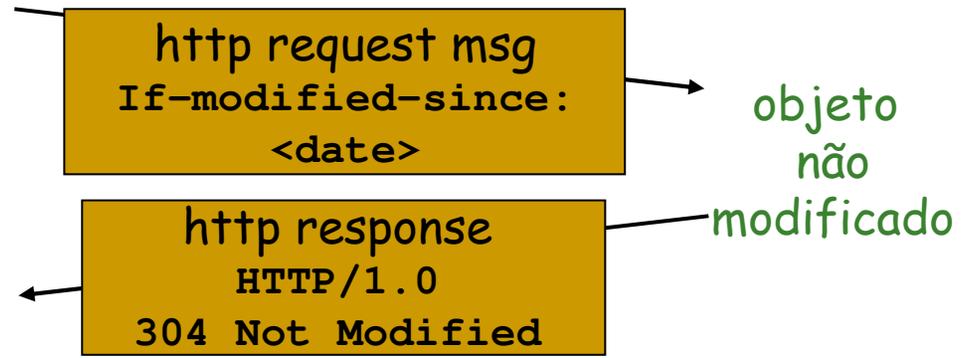
HTTP

■ Conditional GET

- não envia objeto se a informação está atualizada (cached)
- cliente especifica a data da cópia na mensagem http request

cliente

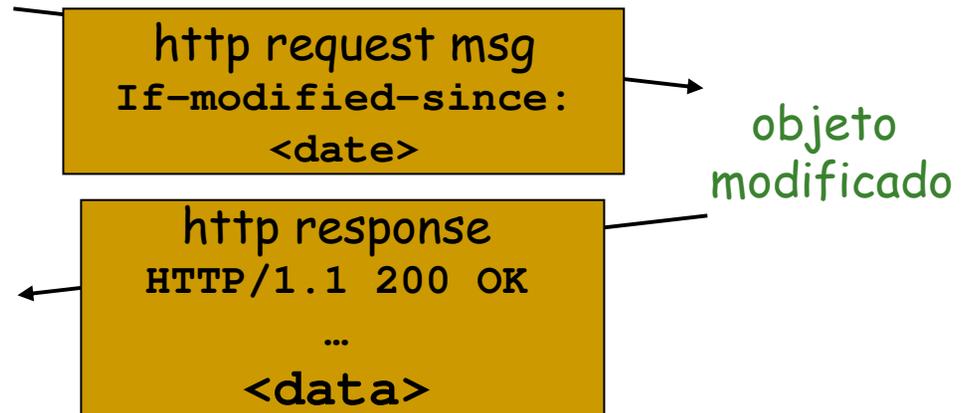
servidor



If-modified-since: <date>

- a resposta do servidor não contém o objeto caso a cópia esteja atualizada

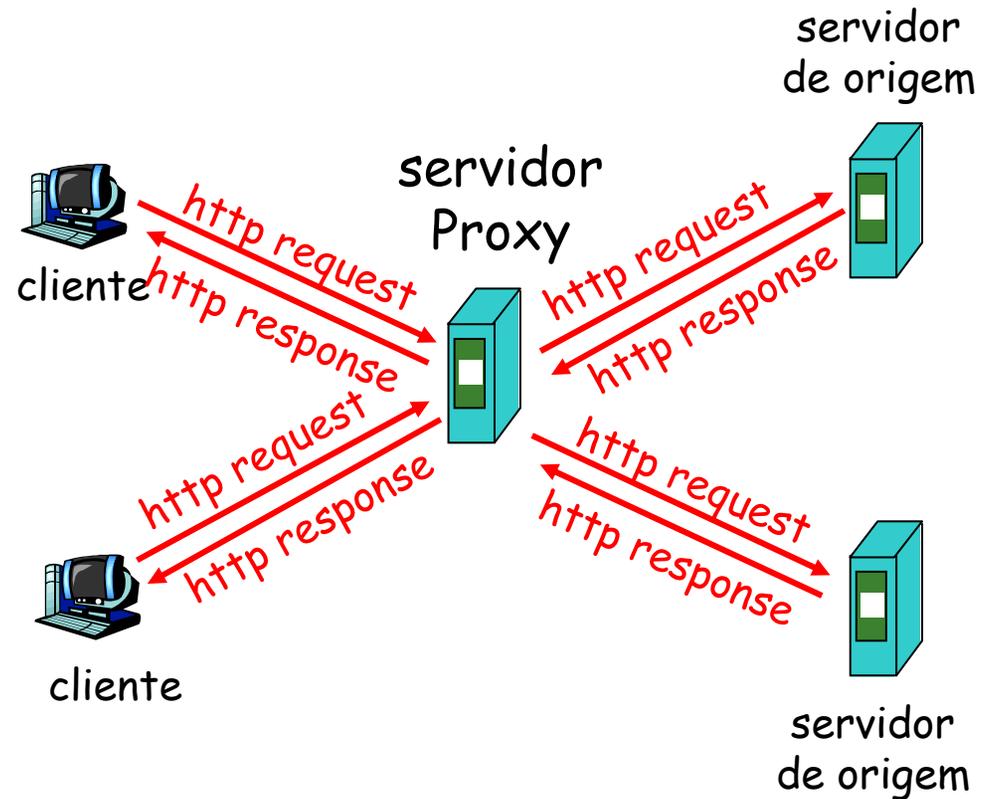
HTTP/1.0 304 Not Modified



HTTP

Web Caching (proxy server)

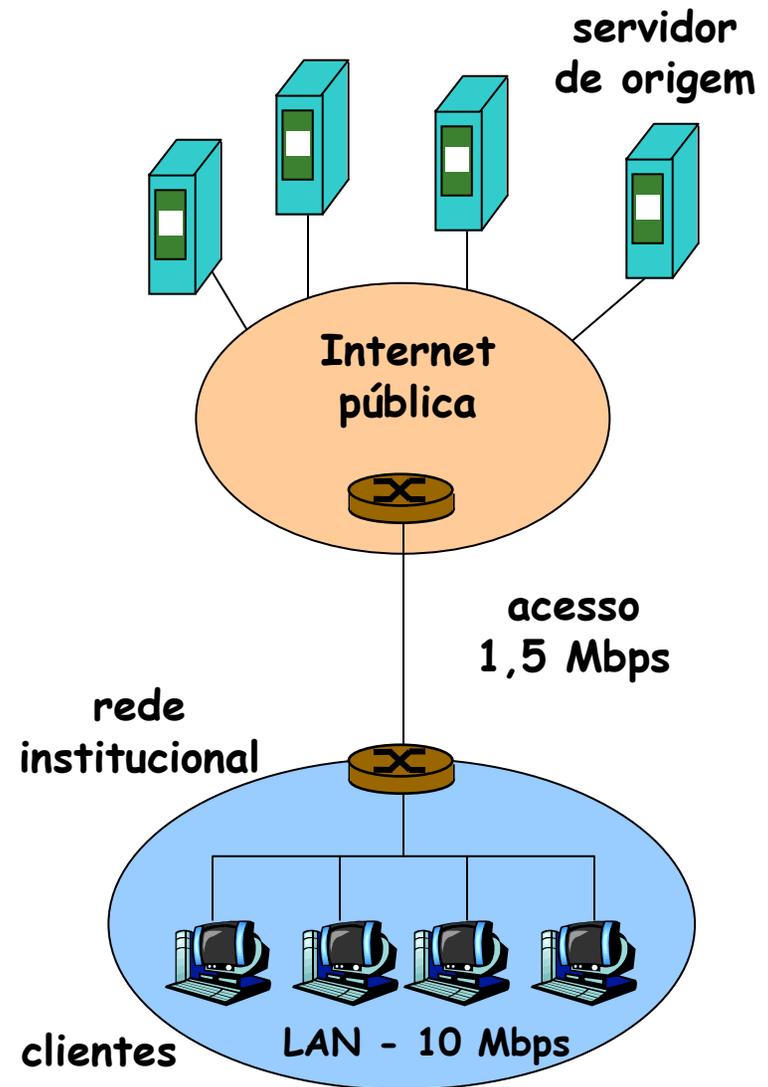
- usuário ativa o acesso à Web via web cache
- cliente envia todas as requisições HTTP para o web cache
- se o objeto está no web cache, este envia imediatamente o objeto na mensagem HTTP response
- caso contrário requisita o objeto ao servidor original e retorna ao cliente a mensagem HTTP response



HTTP

Benefícios

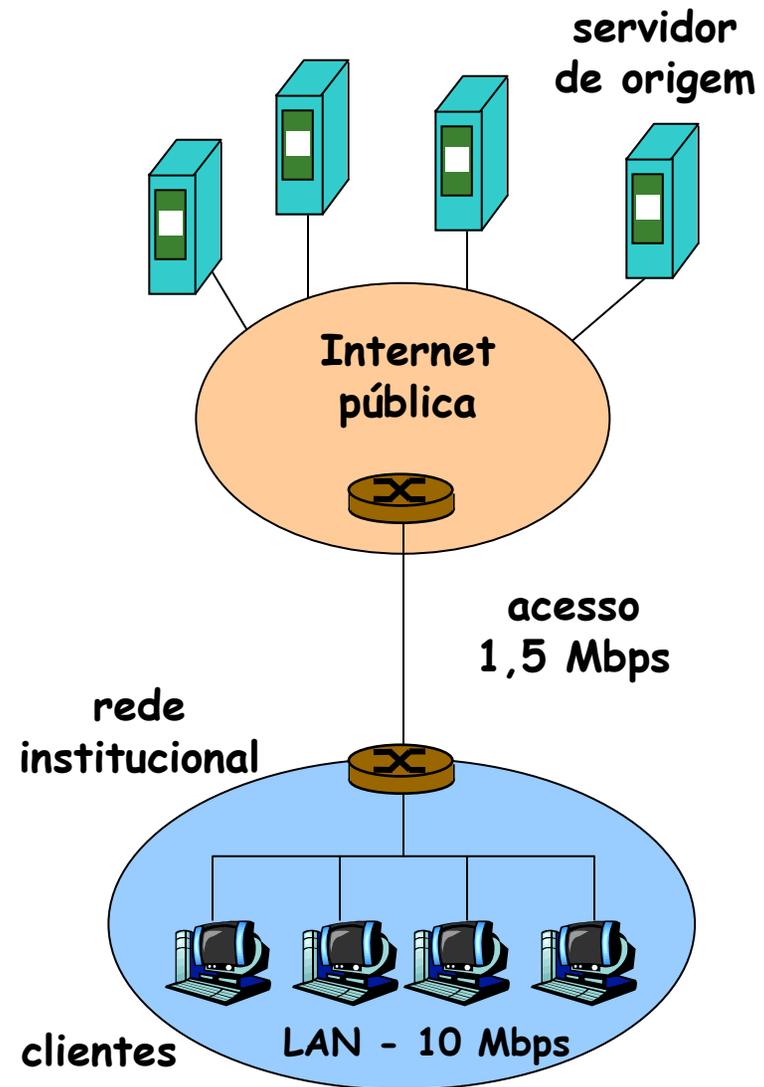
- redução do tempo de resposta
- redução do tráfego no enlace de acesso
- redução do tráfego geral na Web



HTTP

Exemplo

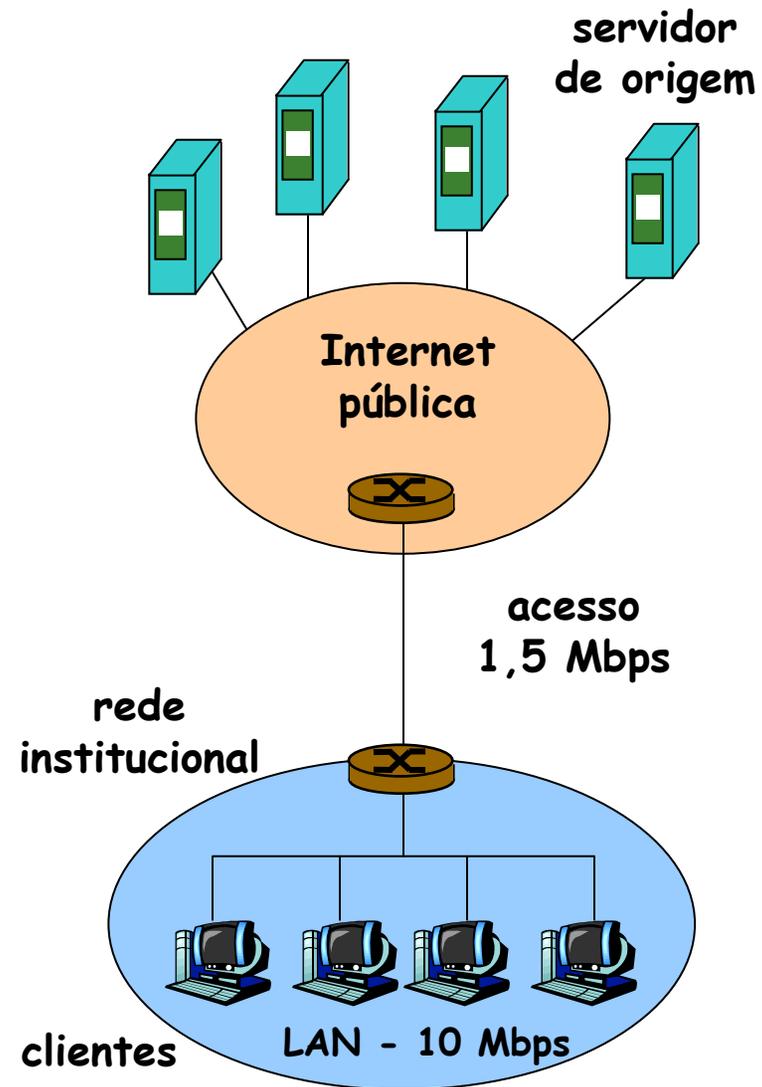
- considere a rede institucional da figura conectada à Internet através de um link de 1,5 Mbps
- suponha que o tamanho médio dos objetos seja de 100 Kbits e a taxa média de requisições dos browsers 15 req/s



HTTP

Exemplo

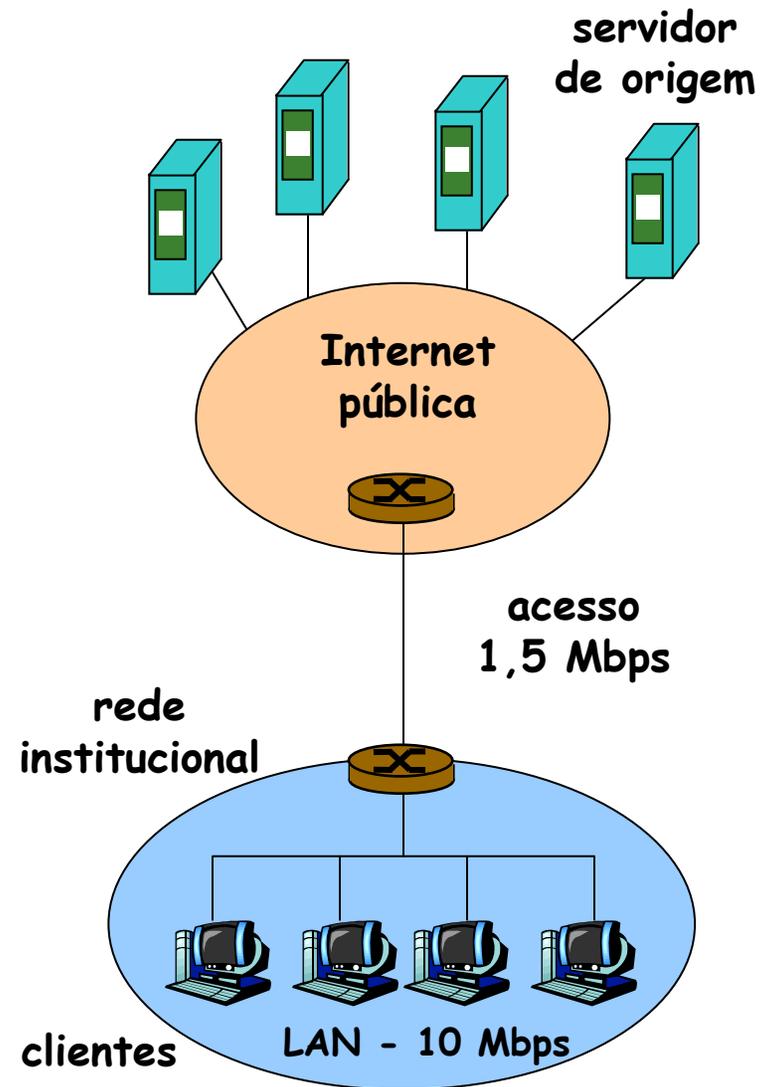
- vamos assumir um tempo de resposta de 2 segundos em média para a Internet
- o tempo de resposta total é a soma do atraso na LAN, do atraso de acesso e do atraso da Internet
- estimativa da intensidade do tráfego na LAN $\rightarrow (15 \text{ req/s}) \times (100\text{Kb/req}) \div (10 \text{ Mbps}) = 0,15$
- estimativa da intensidade do tráfego no acesso $\rightarrow (15 \text{ req/s}) \times (100\text{Kb/req}) \div (1,5 \text{ Mbps}) = 1,00$



HTTP

Exemplo

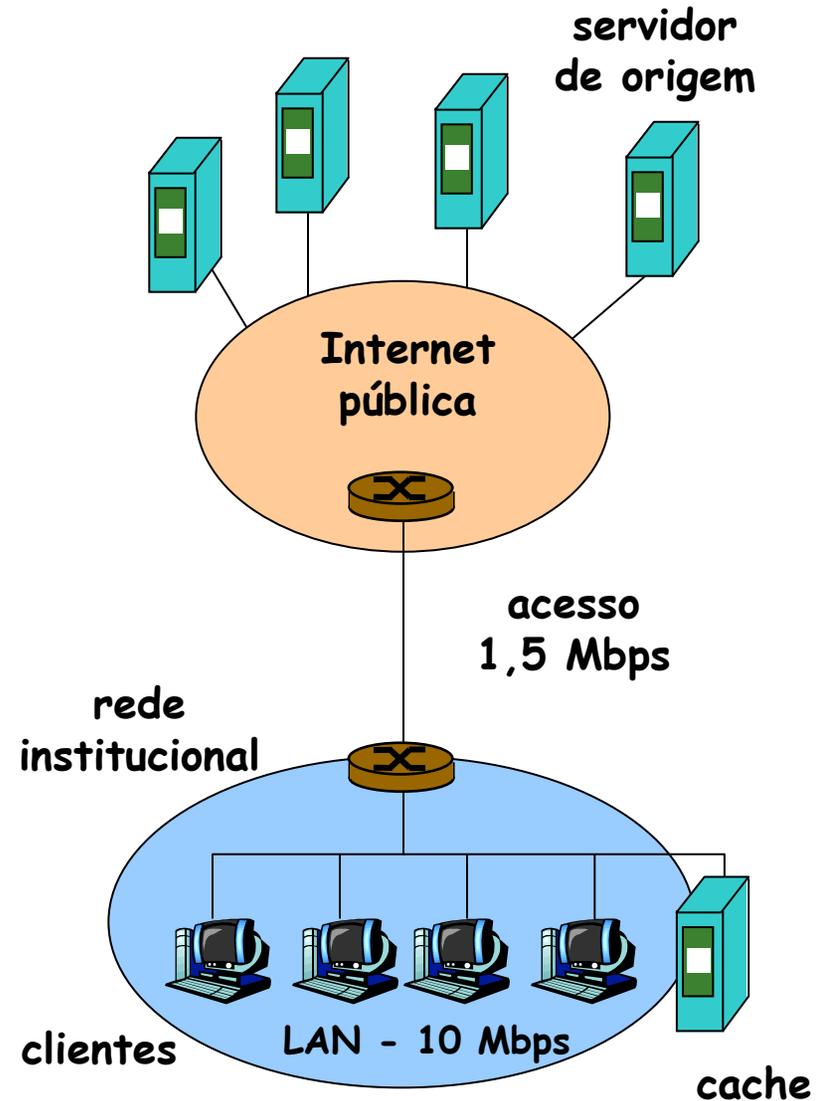
- uma intensidade de tráfego igual 0,15 em uma LAN resulta em atrasos de dezenas de milissegundos
- a medida que o tráfego do link se aproxima de 1 o atraso cresce indefinidamente



HTTP

Exemplo

- soluções possíveis – aumento do link de acesso ou uso de cache
- um cache atendendo 40% das requisições (*hit rate*) já reduz a utilização do link para 0,6
- um intensidade de tráfego no acesso menor que 0,8 corresponde a um atraso pequeno
- desprezível comparado aos 2 segundos da Internet



APLICATIVOS DE ACESSO E TRANSFERÊNCIA DE ARQUIVOS

- Muitos sistemas de redes fornecem computadores com capacidade para acessar arquivos em máquinas remotas
 - Os projetistas exploraram uma variedade de abordagens para a questão do acesso remoto, com cada tipo de abordagem otimizando um conjunto especial de objetivos
 - Arquivos podem ser compartilhados de duas maneiras: acesso *online* e transferência de arquivos
-

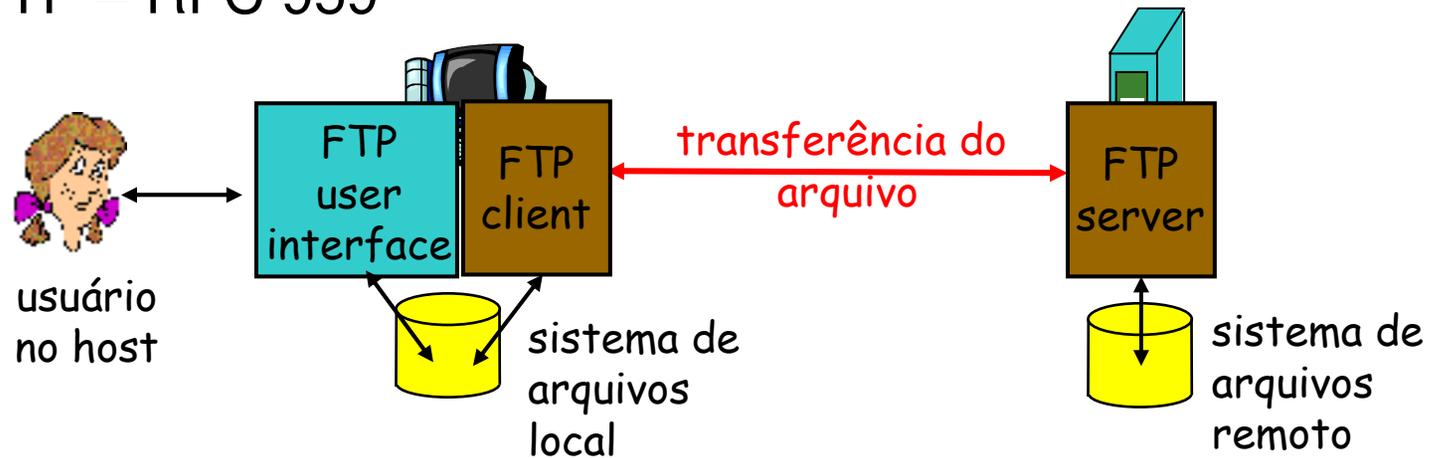
APLICATIVOS DE ACESSO E TRANSFERÊNCIA DE ARQUIVOS

- No método de acesso online vários programas podem acessar um arquivo único simultaneamente. As mudanças no arquivo ocorrem imediatamente e estão disponíveis em todos os programas
 - No método de acesso através de transferência, sempre que um programa deseja acessar um arquivo, ele obtém uma cópia local, executa as mudanças (se for o caso) e devolve o arquivo modificado para o local de origem
 - Principais protocolos TCP/IP para manipulação de arquivos – FTP, TFTP e NFS
-

FTP

■ File Transfer Protocol

- ❑ transfere arquivos de/para um host remoto
- ❑ modelo cliente/servidor
- ❑ cliente – inicia a transferência (de ou para o servidor)
- ❑ servidor – host remoto
- ❑ FTP – RFC 959

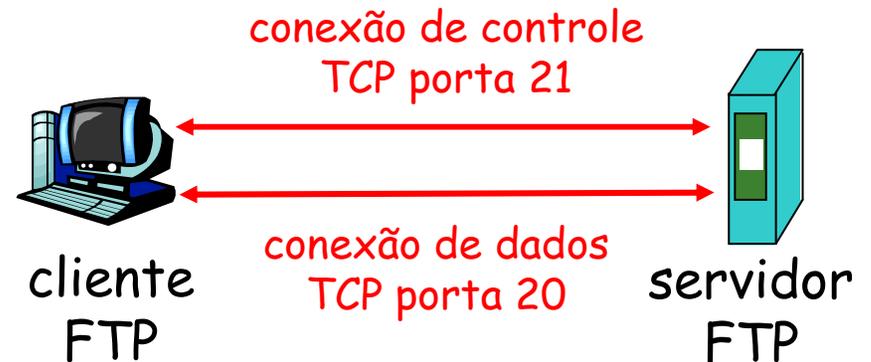


FTP

- A transferência de arquivos é um dos aplicativos usados com mais frequência e responde por grande parte do tráfego da rede
 - O FTP oferece algumas facilidades além da transferência propriamente dita
 - Acesso interativo – muitas implementações fornecem uma interface que permite interação com os servidores remotos
 - Especificação de formato – o FTP permite que o cliente especifique o tipo e o formato dos dados armazenados, como texto (ASCII ou EBCDIC) ou números inteiros binários
 - Controle de autenticação – o FTP requer que os clientes enviem um nome de login e um senha para o servidor antes de poder acessar
-

FTP

- O cliente FTP contata o servidor na porta 21 do TCP
- Duas conexões TCP paralelas são abertas:
 - **control** – troca comandos e respostas entre cliente e servidor (*out-of-band control*)
 - **data** – transfere dados de arquivos de ou para o servidor
- O servidor FTP mantém informações de estado da conexão – diretório corrente, autenticação



FTP

Exemplos de comandos

- enviados como texto ASCII no canal de controle
- **USER username**
- **PASS password**
- **LIST** retorna a lista dos arquivos no diretório corrente
- **RETR filename** recupera (traz) um arquivo
- **STOR filename** armazena (envia) um arquivo para o host remoto

Exemplos de códigos

- código de status e comentário (como no HTTP)
 - **331 Username OK, password required**
 - **125 data connection already open; transfer starting**
 - **425 Can't open data connection**
 - **452 Error writing file**
-

TFTP

■ Trivial File Transfer Protocol

- ❑ O FTP é muito complexo e de difícil implementação
 - ❑ Muitas aplicações não necessitam de todos os recursos oferecidos pelo FTP
 - ❑ Alguns sistemas que precisam transferir arquivos também não dispõem de toda a complexidade de hardware e sistemas operacionais sofisticados
 - ❑ O TFTP não oferece autenticação, sendo portanto muito mais simples
 - ❑ O tamanho do código é muito menor
-

TFTP

- Um aplicativo de tamanho reduzido como o TFTP permite aos fabricantes de dispositivos “sem disco” codificá-lo na ROM e utilizá-lo para obter a imagem do SO quando o sistema é ligado
 - O TFTP não requer um serviço de transporte confiável, e portanto utiliza o UDP na porta 69
 - A transmissão do arquivo é feita em blocos de tamanho fixo de 512 bytes
 - Aguarda um ACK para cada bloco enviado
 - Primeiro pacote enviado especifica nome do arquivo e direção de transferência
-

TFTP

- Formato das PDUs TFTP

2 octetos opcode	n octetos	1 octeto	n octetos	1 octeto
req. de leitura	nome do arquivo	0	modo	0

2 octetos opcode	n octetos	1 octeto	n octetos	1 octeto
req. de escrita	nome do arquivo	0	modo	0

2 octetos opcode	2 octetos	até 512 octetos
dados	# bloco	octetos de dados

2 octetos opcode	2 octetos
ACK	# bloco

2 octetos opcode	2 octetos	n octetos	1 octeto
erro	código de erro	mensagem de erro	0

NFS

■ Network File System

- ❑ Permite aos computadores acesso remoto transparente a arquivos na rede
 - ❑ Desenvolvido inicialmente pela Sun Microsystems
 - ❑ Portável para diferentes máquinas, sistemas operacionais, arquiteturas de redes e protocolos de transporte
 - ❑ Um computador (cliente) pode realizar operações sobre arquivos que residam em outros computadores (servidores)
-

NFS

■ Funcionamento

- ❑ o cliente realiza uma chamada de acesso a um arquivo remoto
 - ❑ a chamada de acesso é convertida em um *request* do protocolo NFS
 - ❑ enviada ao sistema servidor através da rede
 - ❑ o servidor recebe o pedido e realiza a operação no sistema de arquivos
 - ❑ o servidor envia uma resposta ao cliente
-

NFS

- Ao projetar o NFS os projetistas optaram por criar três mecanismos diferentes
 - O próprio protocolo NFS
 - Um mecanismo de RPC (Remote Procedure Call) para fins gerais
 - Um mecanismo denominado XDR (eXternal Data Representation)
-

NFS

- A idéia era permitir que o RPC e o XDR pudessem ser utilizados por outros programas
 - Da forma como foi concebido o NFS acessa arquivos remotos usando exatamente as mesmas operações utilizadas por arquivos locais
 - O RPC e o XDR fornecem mecanismos para construir programas distribuídos
-

NFS – RPC (Remote Procedure Call)

- O RPC é um mecanismo que permite a comunicação entre os lados cliente e servidor de um programa
 - Do lado cliente o programador atribui alguns procedimentos como remotos, obrigando o compilador a incorporar a estes procedimentos os códigos RPC
 - Do lado servidor o programador implementa os procedimentos desejados e usa outros recursos RPC para declará-los parte do servidor
 - Quando o programa cliente chama um dos programas remotos, o RPC coleta valores, monta uma mensagem, envia ao servidor, aguarda a resposta e armazena os valores devolvidos
-

NFS – XDR (eXternal Data Representation)

- O XDR é uma ferramenta que fornece meios para a passagem de dados entre máquinas heterogêneas
 - O XDR resolve o problema definindo uma representação dos dados independente da máquina
 - Em uma extremidade da comunicação, o programa chama o procedimento XDR para fazer a conversão do formato local para a representação independente do hardware
 - Após a transferência, o programa receptor solicita rotinas XDR para proceder à conversão para a representação local da máquina
-

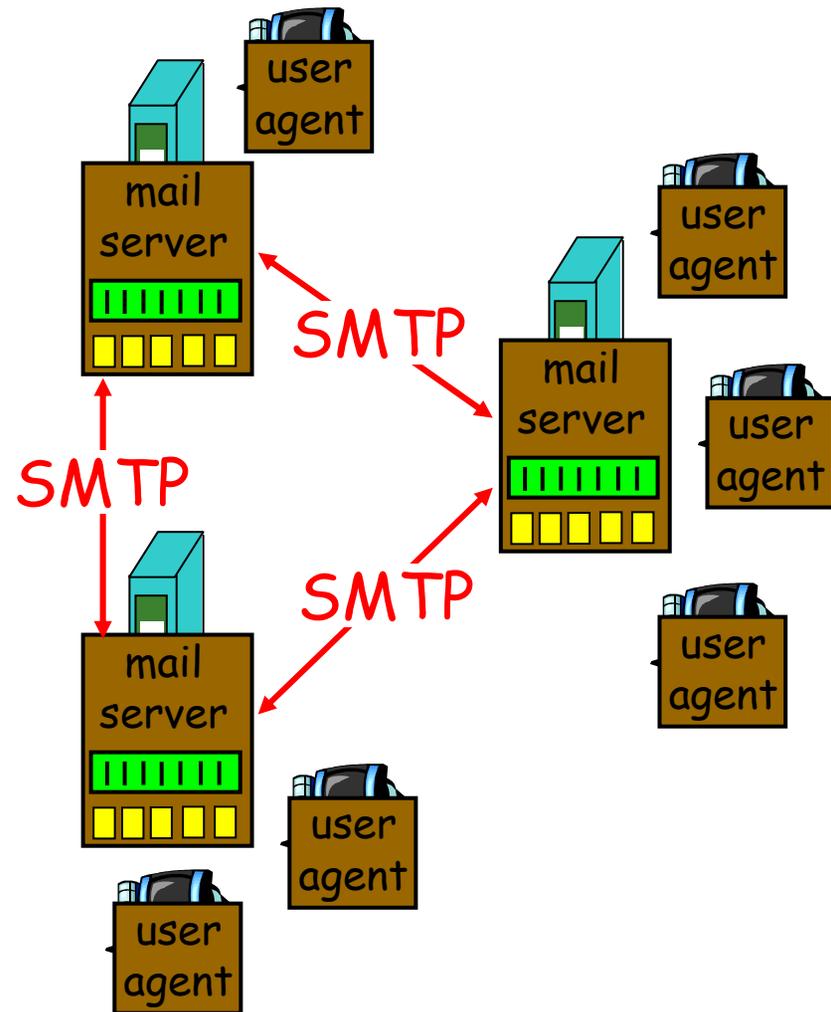
CORREIO ELETRÔNICO

Principais componentes

- user agents
- mail servers
- simple mail transfer protocol – SMTP

User Agent

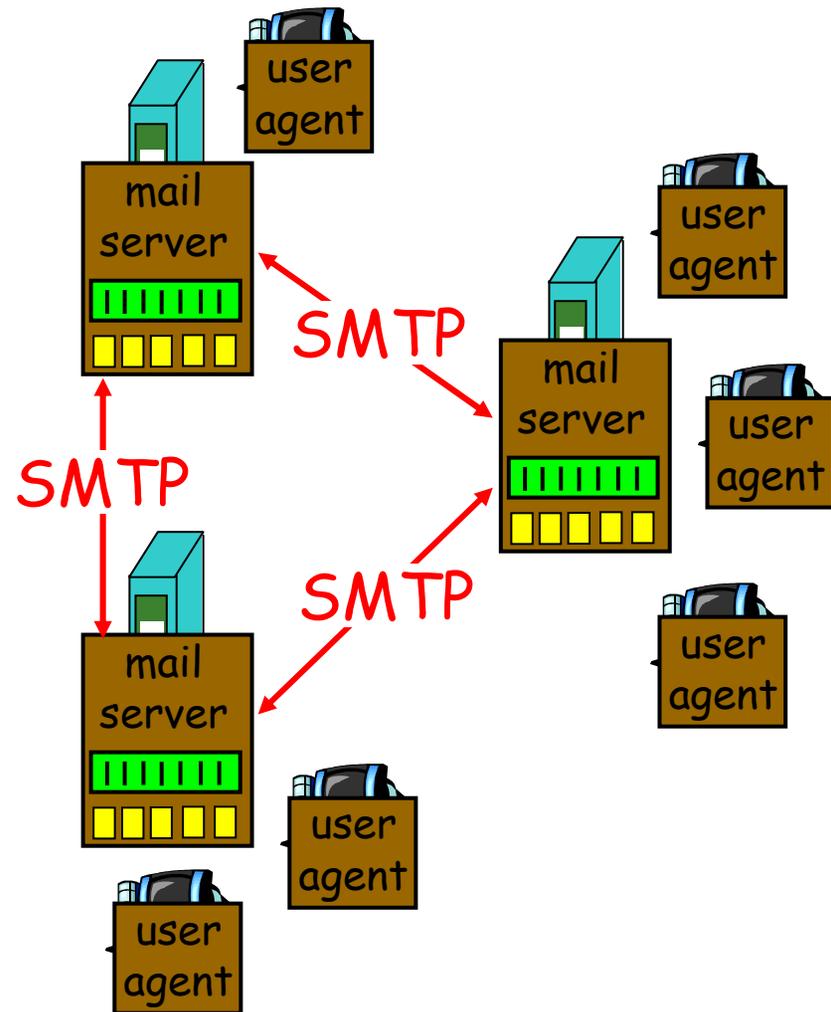
- leitor de correio
- compõe, edita, lê mensagens de correio (e-mail)
- exemplo: Eudora, Outlook, elm, Netscape Messenger
- mensagens de saída e de entrada são armazenadas no servidor



CORREIO ELETRÔNICO

Mail Servers

- **caixa de correio (mailbox)** contém as mensagens recebidas pelo usuário (não lidas)
- **fila de mensagens (message queue)** a serem enviadas
- **protocolo SMTP [RFC 821]** utilizado entre servidores de mail para enviar mensagens de correio eletrônico



SMTP

- **Simple Mail Transfer Protocol**
 - Utiliza o TCP para transferência confiável de mensagens de correio entre o cliente e o servidor na porta 25
 - Transferência direta de servidor para servidor
-

SMTP

- Transferência em três fases
 - handshaking (greeting)
 - transferência das mensagens
 - encerramento
 - Interação do tipo comando/resposta
 - comandos – texto ASCII
 - Resposta – frase e código de status (status code)
 - As mensagens devem ser ASCII de 7 bits
-

SMTP

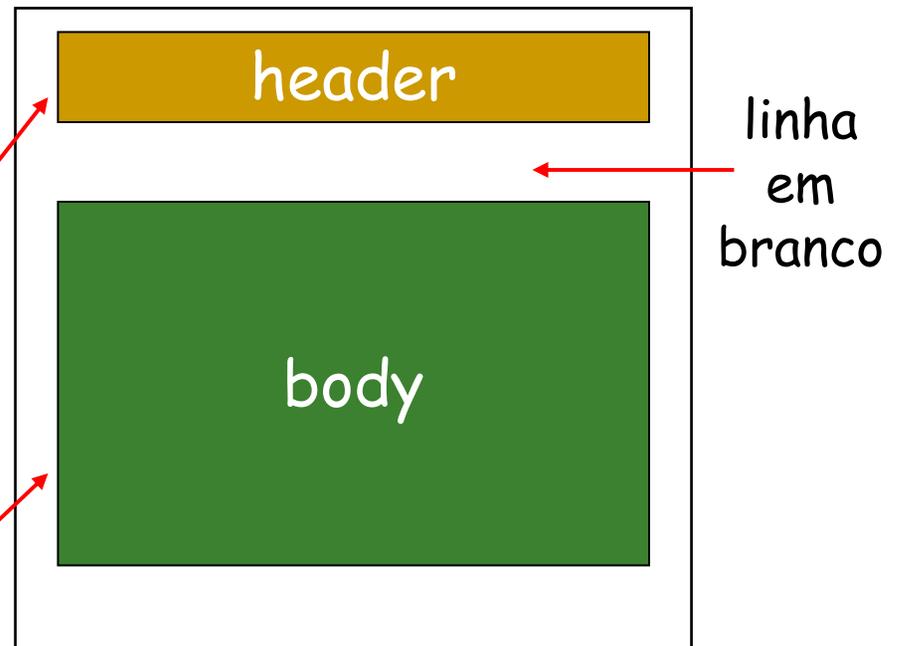
- Exemplo: Usuário Smith no host alpha.edu envia uma mensagem para os usuários Jones, Green e Brown no host beta.gov

```
R:220 beta.gov Simple Mail Transfer Service Ready
S:HELO alpha.edu
R:250 beta.gov
S:MAIL FROM:<smith@alpha.edu>
R:250 OK
S:RCPT TO:<jones@beta.gov>
R:250 OK
S:RCPT TO:<green@beta.gov>
R:No such user here
S:RCPT TO:<brown@beta.gov>
R:250 OK
S:DATA
R:354 Start Mail Input; end with <CR><LF>.<CR><LF>
S:Mensagem
S:<CR><LF>.<CR><LF>
R:250 OK
S:QUIT
R:221 beta.gov Service Closing Transmission Chanel
```

CORREIO ELETRÔNICO

Formato da mensagem

- RFC 822 – padrão para o formato das mensagens de texto
- cabeçalho
 - To:
 - From:
 - Subject:
- diferente dos comandos SMTP
- corpo
 - mensagem com somente caracteres ASCII



MIME

■ Multimedia Mail Extension

- ❑ RFC 2045, 2056 (extensões para multimídia)
- ❑ linhas adicionais no cabeçalho informam o tipo de conteúdo MIME

versão do MIME

método usado para
codificar os dados

dados multimídia, tipo,
subtipo, declaração
de parâmetro

dados codificados

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....base64 encoded data
```

MIME

■ Text

- subtipos exemplo: plain, html

■ Image

- subtipos exemplo: jpeg, gif

■ Audio

- subtipos exemplo: basic (codificação 8-bit □-law), 32kadpcm (codificação delta-pcm 32 kbps)

■ Video

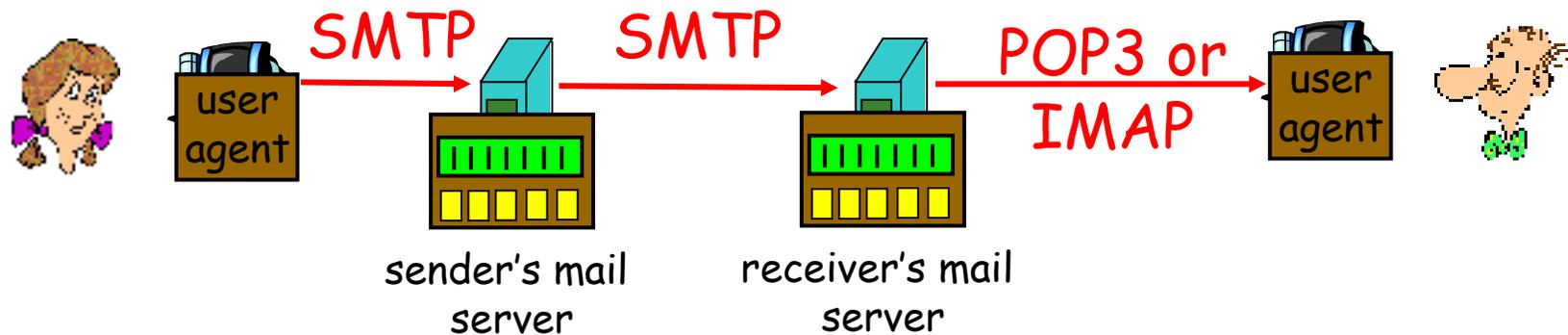
- subtipos exemplo: mpeg, quicktime

■ Application

- outro tipo de dado que precisa ser processado antes de ser exibido
 - subtipos exemplo: msword, octet-stream
-

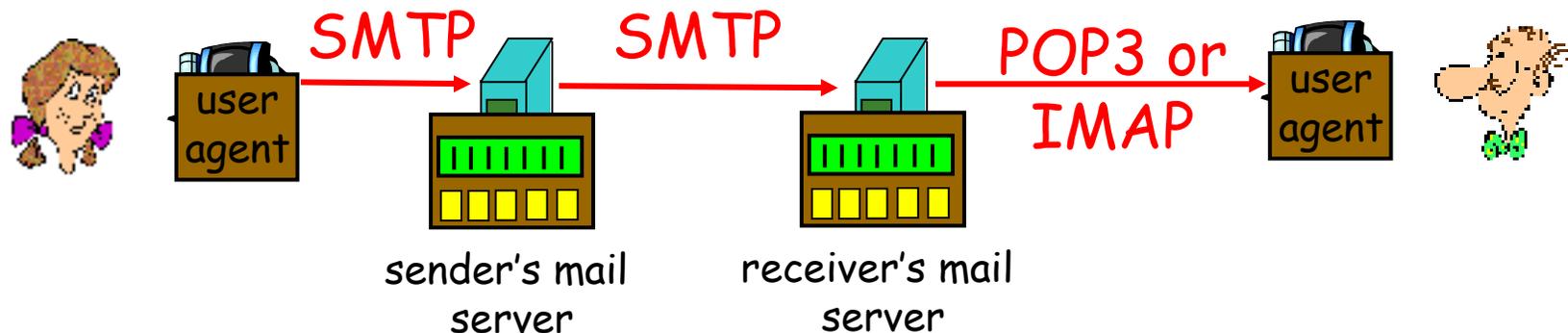
RECUPERAÇÃO DAS MENSAGENS

- SMTP – entrega e armazena mensagens no servidor
- Mail Access Protocol – recupera mensagens do servidor
 - POP
 - IMAP
 - HTTP



RECUPERAÇÃO DAS MENSAGENS

- POP – Post Office Protocol [RFC 1939]
 - autenticação (agent ↔ server) e download
- IMAP – Internet Mail Access Protocol [RFC 1730]
 - mais ferramentas (maior complexidade)
 - permite manipulação das mensagens armazenadas no servidor
- HTTP – Hotmail , Yahoo! Mail, IGWebMail, etc.



POP3

authorization phase

- comandos do cliente
 - **user: username**
 - **pass: senha**
- respostas do servidor
 - **+OK**
 - **-ERR**

transaction phase

- cliente
 - **list: lista as mensagens**
 - **retr: recupera as mensagens pelo número**
 - **dele: apaga**
 - **quit: sai do programa**

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

APLICATIVOS DE LOGIN REMOTO

- O modelo cliente/servidor permite o fornecimento de serviços computacionais específicos
 - Os protocolos de stream confiáveis, como o TCP, também possibilitam o uso interativo de máquinas remotas
 - O uso destes componentes de rede permite a implementação de serviços remotos, acessados através de conexões TCP
-

APLICATIVOS DE LOGIN REMOTO

- Dificuldades na implementação de serviços de rede
 - Necessidade de um servidor para cada tipo de serviço
 - As máquina podem se tornar sobrecarregadas de processos servidores



APLICATIVOS DE LOGIN REMOTO

- A complexidade dos servidores especializados pode ser eliminada permitindo que o usuário estabeleça um sessão de login em uma máquina remota
 - Através do login remoto o usuário tem acesso a todos os comandos disponíveis no sistema remoto
 - Os projetistas do sistema não precisam fornecer servidores especializados
-

APLICATIVOS DE LOGIN REMOTO

■ Problemas

- ❑ O sistema pode não ter sido projetado para interligação em rede
 - ❑ Neste caso só seria possível o login através de teclado e monitor diretamente conectados
 - ❑ O acréscimo do servidor exigiria modificações no SO da máquina
 - ❑ O software cliente tem que lidar com a interpretação de certas seqüências especiais do teclado local
-

TELNET

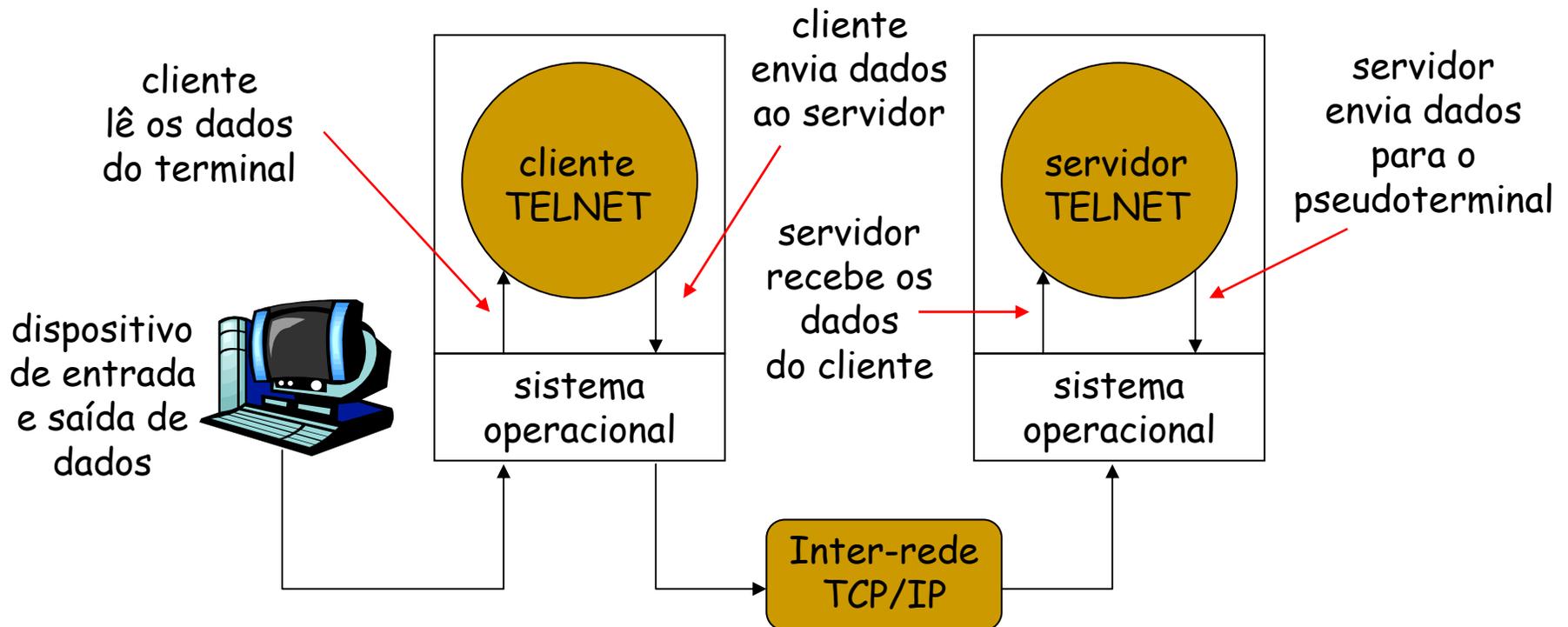
- A pilha TCP/IP inclui um protocolo simples de acesso remoto denominado TELNET
 - Permite que um usuário estabeleça uma conexão TCP com um servidor de login remoto
 - O TELNET transmite os toques no teclado do usuário diretamente para o computador remoto
 - Este terminal também retorna a saída da máquina remota até a tela do usuário
-

TELNET

- O software cliente permite que o usuário especifique a máquina remota fornecendo o nome e o domínio ou o endereço IP
 - Sintaxe: TELNET <nome.domínio> (ou <IP>)
 - Utiliza a porta 23 do TCP se nenhuma outra for especificada
-

TELNET

- Trajeto dos dados em uma sessão TELNET de terminal remoto



INICIALIZAÇÃO E AUTOCONFIGURAÇÃO

- Antes de enviar e receber datagramas IP, um computador conectado à uma rede TCP/IP precisa saber seu endereço IP
 - Outras informações também são necessárias, como o endereço do roteador, a máscara de sub-rede e o endereço do servidor de nomes
 - O protocolo projetado inicialmente para esta tarefa era o RARP (Reverse Address Resolution Protocol)
-

INICIALIZAÇÃO E AUTOCONFIGURAÇÃO

- Desvantagens do RARP
 - ❑ Somente retorna o endereço IP
 - ❑ Como trabalha em baixo nível, opera diretamente com o hardware de rede
 - ❑ A requisição RARP é enviada em um quadro MAC de difusão
 - ❑ O servidor RARP precisa estar na mesma rede física que a máquina cliente
-

BOOTP

- Protocolo para a obtenção de endereços IP como o RARP
 - O BOOTP utiliza UDP (porta 67) para obter informações através de broadcast IP
 - O endereço IP utilizado para o envio da solicitação BOOTP é o endereço especial de difusão limitada – 255.255.255.255
-

BOOTP

- O software IP pode receber e difundir datagramas que especificam o endereço de broadcast limitado, mesmo antes de saber o seu endereço IP local
 - Servidor BOOTP responde à requisição do cliente com broadcast (usual) ou altera diretamente sua tabela ARP
-

DHCP

- **Dynamic Host Configuration Protocol**
 - É compatível com BOOTP (também utiliza o UDP na porta 67) e provê mais funções que o antecessor
 - Pode passar mais informações como a máscara de sub-rede
-

DHCP

- Permite três tipos de configuração
 - **Manual** – atribui um endereço para uma máquina específica
 - **Automática** – atribui endereços permanentes na primeira vez que a máquina se conecta na rede
 - **Dinâmica** – “empresta” um endereço para o cliente por um período de tempo determinado
 - O cliente passa para o servidor um identificador (geralmente o endereço de hardware) através do qual o servidor determina o tipo de atribuição
-

DHCP

- Etapas da configuração de uma máquina cliente utilizando o protocolo DHCP
 - host envia mensagem de broadcast “**DHCP discover**” para 255.255.255.255
 - servidor DHCP responde com mensagem “**DHCP offer**”
 - host requisita endereço IP através da mensagem “**DHCP request**”
 - servidor DHCP confirma o endereço com mensagem “**DHCP ack**”
-

REFERÊNCIAS

- Redes de Computadores

Andrew S. Tanenbaum

- Interligação em redes com TCP/IP

Douglas E. Comer

- Redes de Computadores

Soares, Lemos & Colcher
