

Arquitetura TCP/IP

Nível de Aplicação (HTTP, SMTP, FTP & DNS)

Prof. Helber Silva

Roteiro

- Introdução
- Protocolo HTTP
- Protocolo SMTP
- Protocolo FTP
- Protocolo DNS
- Conclusão

Introdução

- Desenvolvimento de aplicações é uma motivação para o uso de redes
- Aplicação de rede representa um par de processos que enviam mensagens entre si usando uma rede
 - Correio eletrônico, bate-papo, e a maior de todas: a *World Wide Web* (WWW), conhecida apenas por Web
- A Web permite o acesso a documentos de hipertexto (texto em formato digital) espalhados mundo afora
 - Compara-se, em importância, a invenções como o telefone, o rádio e a TV. Diferentemente dos dois últimos, porém, a Web funciona sob demanda

Arquiteturas de Aplicação de Rede

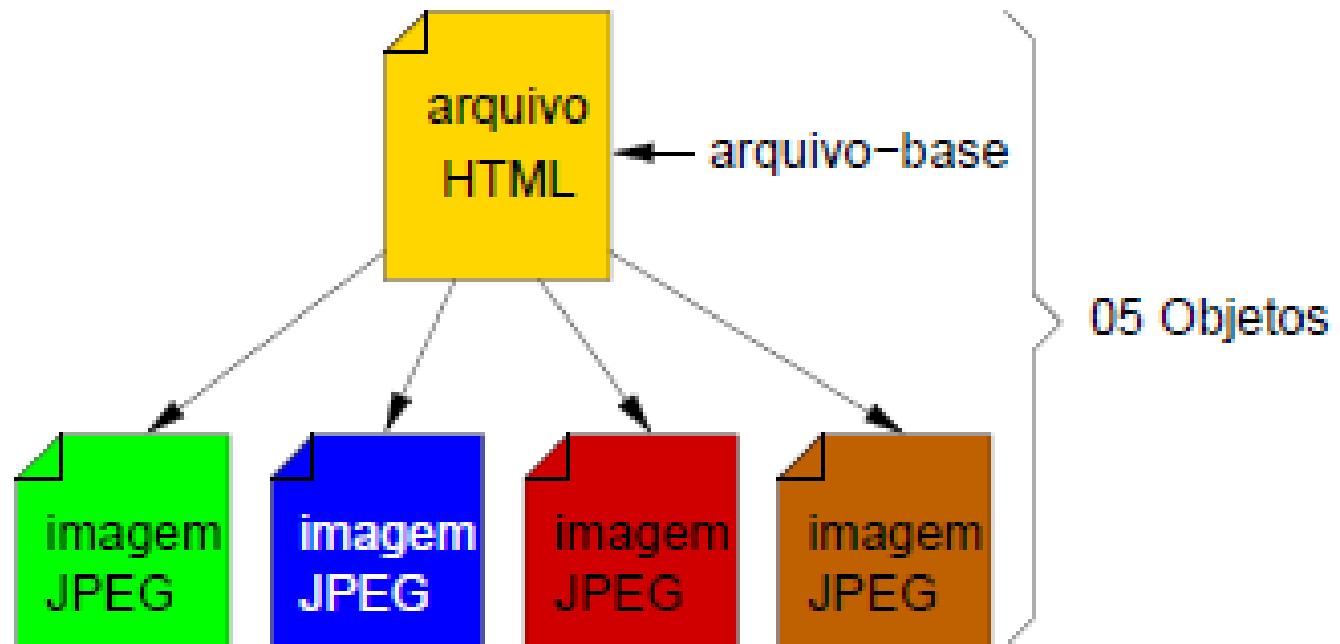
- Determina *como* a aplicação é organizada nos **sistemas finais** (máquinas que executam as aplicações)
 - Independe da arquitetura da rede, i.e, assume-se uma infraestrutura que provê serviços à aplicação
- Arquiteturas de aplicação de rede conhecidas
 - Cliente-servidor
 - Par a par (*peer to peer* ou P2P)

Arquiteturas de Aplicação de Rede

- Arquitetura Cliente-Servidor
 - Uma máquina, chamada *servidor*, atende a requisições de (possivelmente) muitas outras máquinas, chamadas *clientes*
 - Exemplo: um servidor Web (aplicação servidor) atende requisições de páginas de navegadores (aplicação cliente) instalados em outras máquinas
 - Servidores Web conhecidos: Apache e Microsoft Internet Information Server

Objetos Web

- Conteúdo da Web é estruturado em *objetos Web*



Objetos Web

- Acesso via URL (*Universal Resource Locator*)

- Exemplo:

`www.ifrn.edu.br/docente/alguem.jpg`

– `www.ifrn.edu.br` → nome do *host*
(hospedeiro)

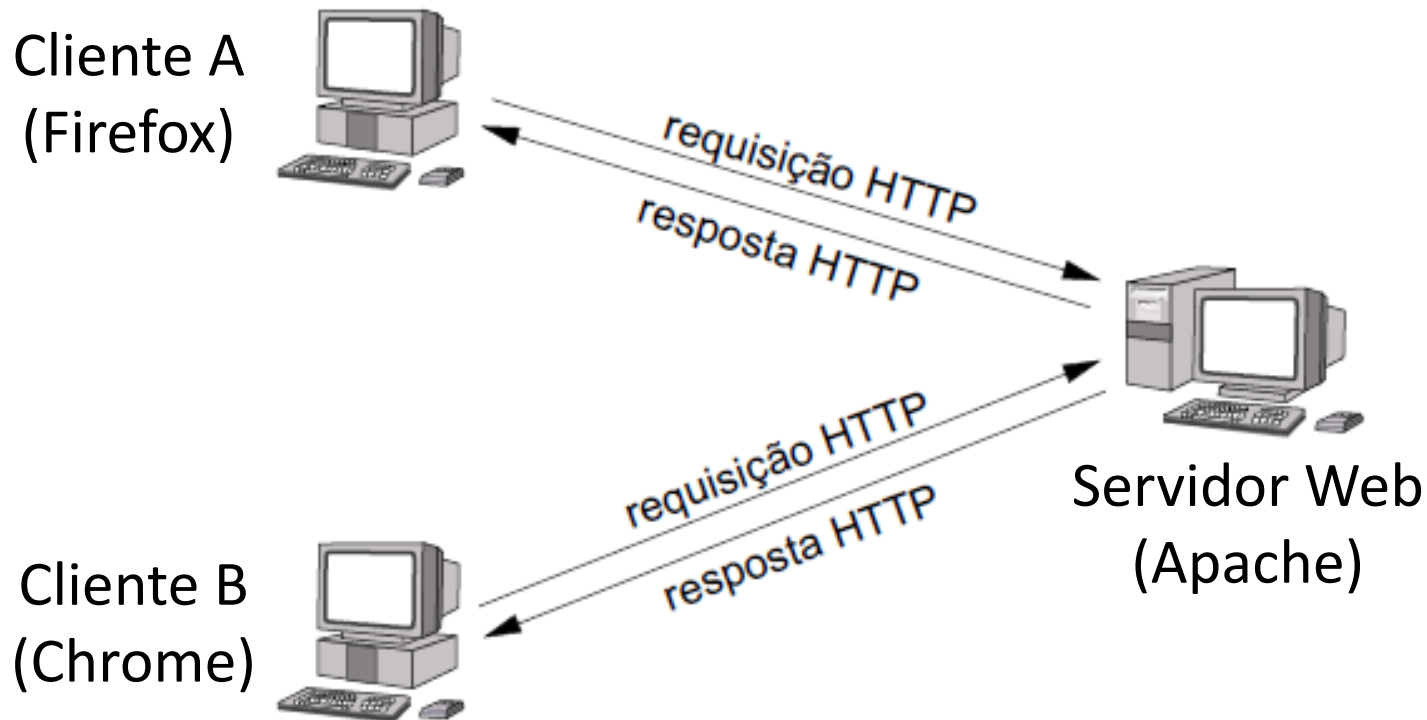
– `/docente/alguem.jpg` → caminho do objeto

Protocolos do nível de aplicação

- Responsáveis pela definição de:
 - Tipos de mensagens trocadas pelos processos
 - Sintaxe (estrutura) das mensagens
 - Semântica (significado) das mensagens
 - Regras de envio, recepção e resposta de mensagens

Protocolo HTTP

- HTTP é acrônimo de *HyperText Transfer Protocol*
- Usado na transferência de objetos Web



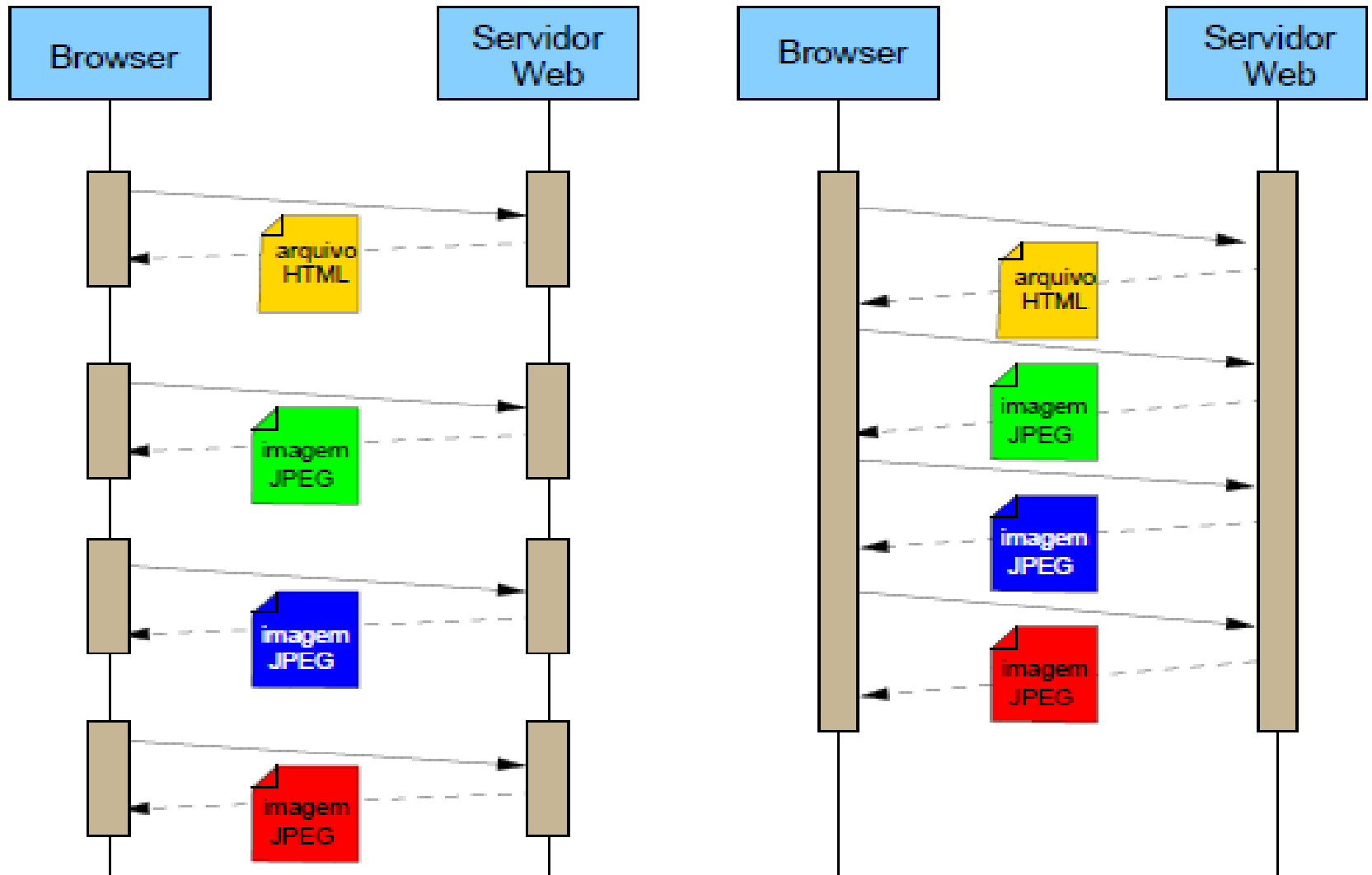
Protocolo HTTP

- Processo de requisição-resposta no modelo cliente servidor
 - Cliente (p. ex, navegador) envia mensagem de requisição de objetos web
 - Servidor envia mensagem de resposta, possivelmente com os objetos web requisitados pelo cliente
- Usa os serviços de entrega do protocolo TCP (*Transmission Control Protocol*) no nível de transporte
 - TCP garante a entrega dos dados entre as aplicações
 - Servidor HTTP usa a **porta 80** do TCP
 - Porta: identifica o ponto final de comunicação no sistema operacional de um computador

Protocolo HTTP

- Há atualmente duas versões: o HTTP/1.0 (RFC 1945) e o HTTP/1.1 (RFC 2616)
 - RFC (*Request For Comments*) descreve os padrões de cada protocolo da Internet
- A diferença entre o HTTP/1.0 e o HTTP/1.1 é que o primeiro usa **conexões TCP transientes**, enquanto o último faz uso de **conexões TCP persistentes**

Conexões Transientes x Persistentes



Requisição HTTP

- Exemplo típico

```
GET /someDir/page.html HTTP/1.1
```

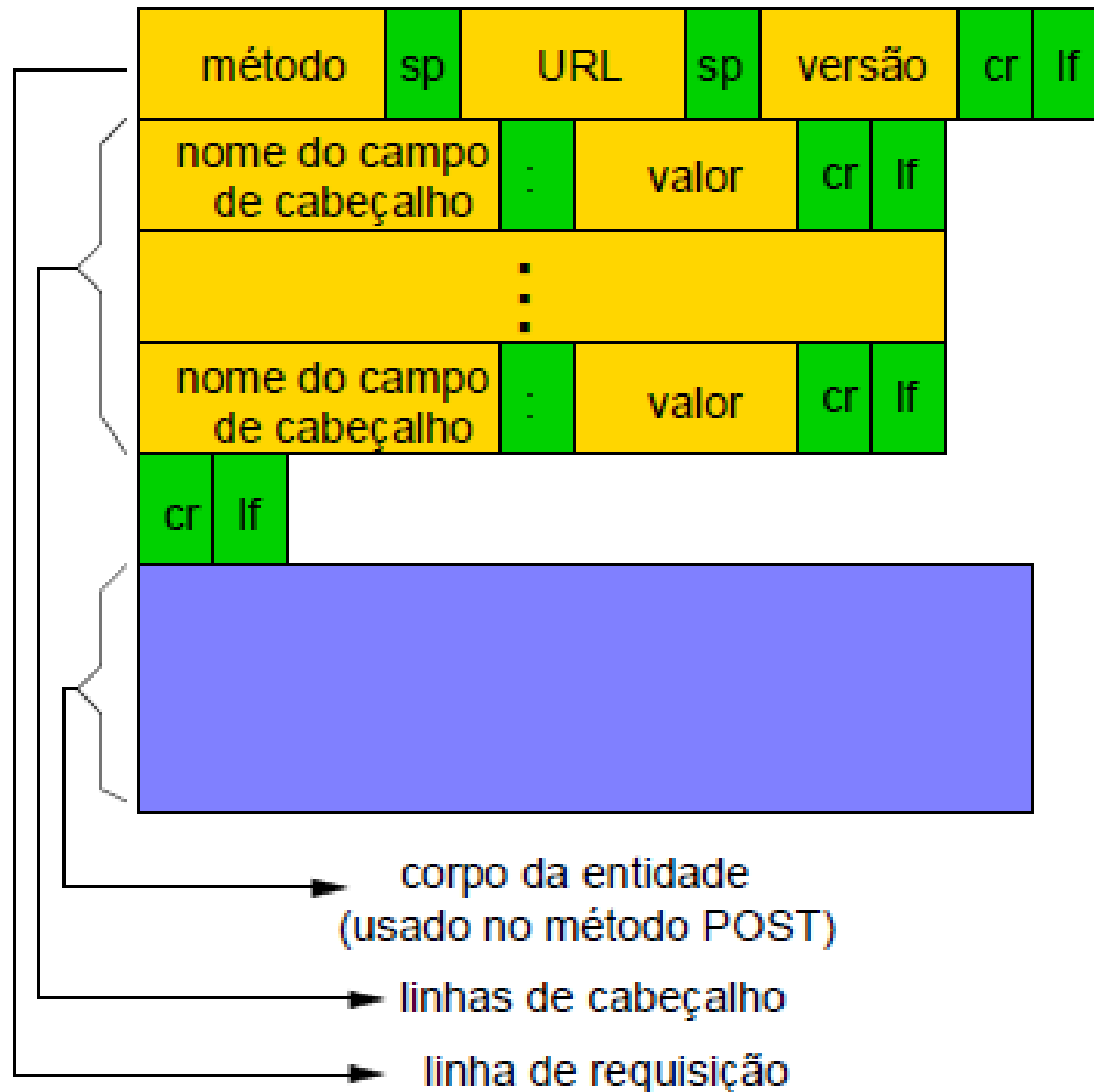
```
Host: www.someSchool.edu
```

```
Connection: close
```

```
User-agent: Mozilla/4.0
```

```
Accept-language: fr
```

Formato da Requisição HTTP



Resposta HTTP

- Exemplo típico

```
HTTP/1.1 200 OK
```

```
Connection: close
```

```
Date: Thu, 06 Aug 1998 12:00:15 GMT
```

```
Server: Apache/1.3.0 (Unix)
```

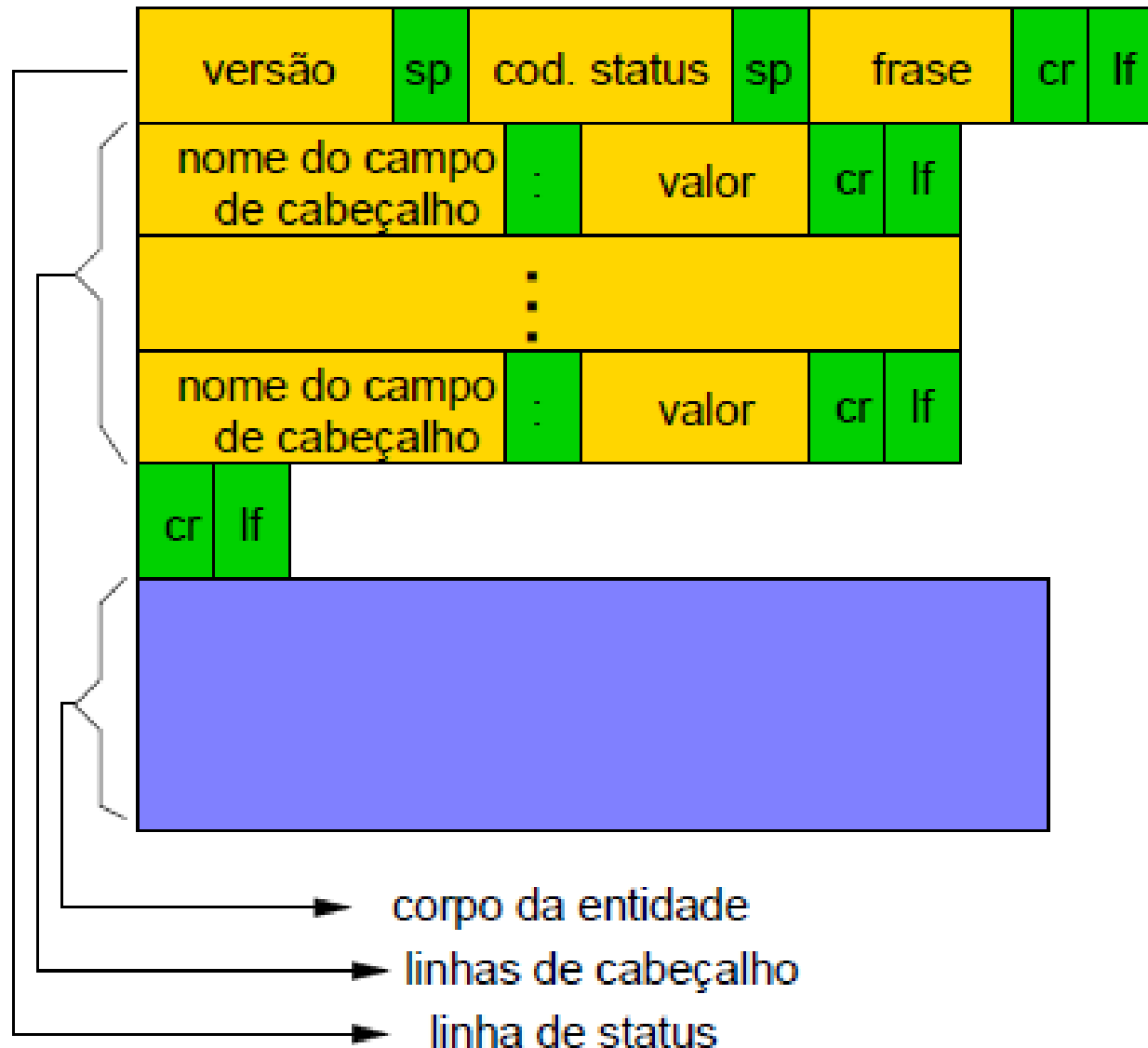
```
Last-Modified: Mon, 22 Jun 1998 09:23:24  
GMT
```

```
Content-Length: 6821
```

```
Content-Type: text/html
```

```
(data data data data data data ...)
```

Formato de Resposta HTTP



Códigos de Status de Respostas HTTP

- 1xx: informação. Ex.: 100 Continue → o servidor concorda em atender a requisição do cliente
- 2xx: sucesso. Ex.: 200 OK → requisição bem-sucedida; 204 No Content → sem conteúdo
- 3xx: redirecionamento. Ex.: 301 Moved Permanently → a página foi removida
- 4xx: erro do cliente. Ex.: 404 Not Found → página não encontrada
- 5xx: erro no servidor. Ex.: 500 Internal Server Error → erro interno no servidor; 503 Service Temporarily Unavailable → tente mais tarde

Protocolo HTTP

- É considerado um protocolo **sem estado**
 - Não mantém informações prévias de requisições HTTP
 - Respostas HTTP são enviadas tantas vezes forem recebidas requisições HTTP de um mesmo cliente

Cookies

- Os famosos “biscoitinhos”...
- Arquivos gravados no hospedeiro do cliente HTTP para armazenar a informação de clientes (persistência de comunicações usando o protocolo HTTP)
- Tamanho: 4KB (pequeno); quantidade de campos: até 5
- Exemplo

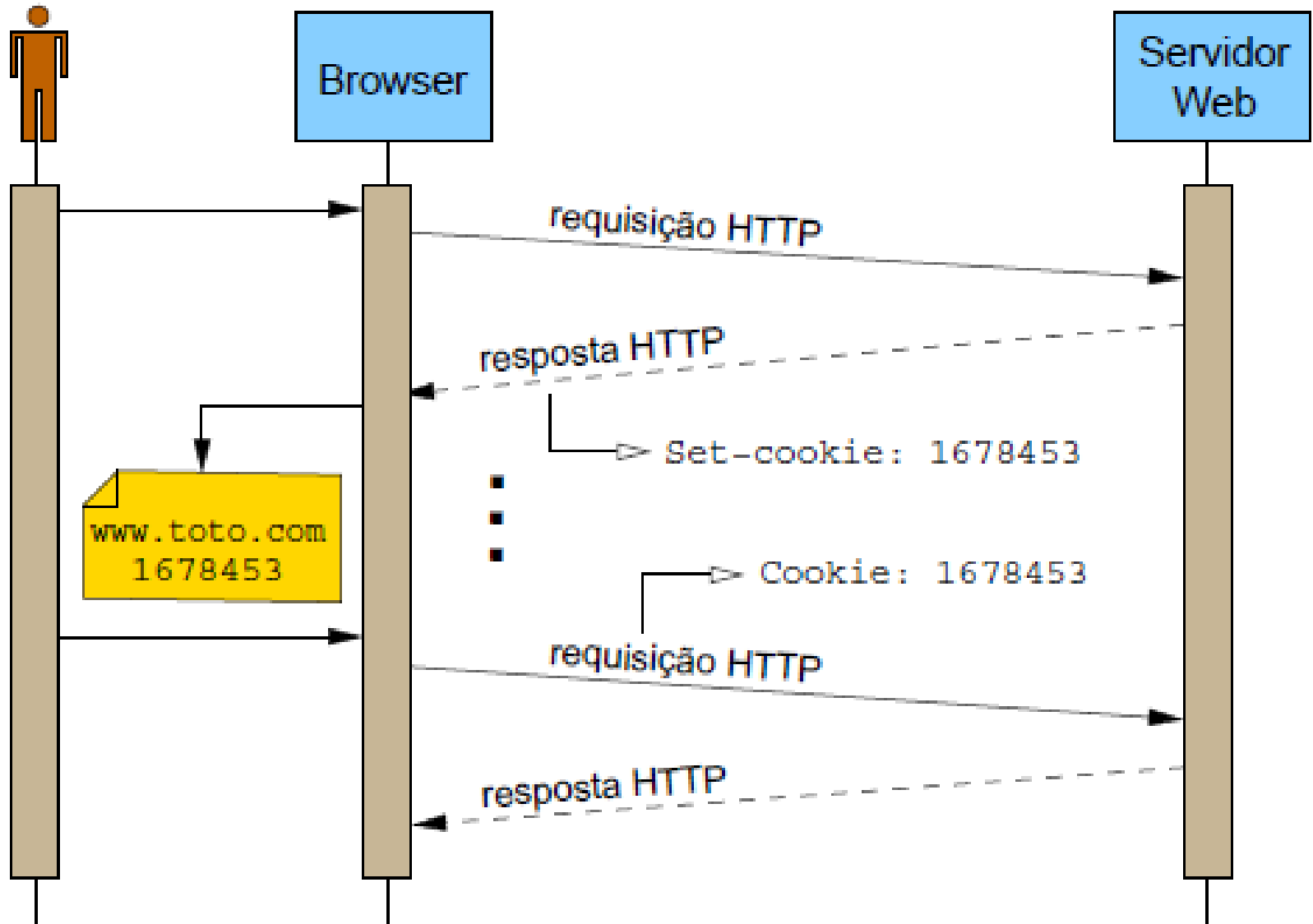
```
toms-casino.com / CustomerID=457793521  
15-10-02 17:10 Yes
```

Funcionamento dos *Cookies*

1. Cliente HTTP (navegador) envia uma mensagem de requisição HTTP a um servidor HTTP (servidor web)
2. Servidor HTTP envia uma mensagem de resposta HTTP (com o *cookie* atribuído) para o cliente HTTP
3. Cliente HTTP envia nova requisição HTTP juntamente com o *cookie* recebido

Funcionamento dos *Cookies*

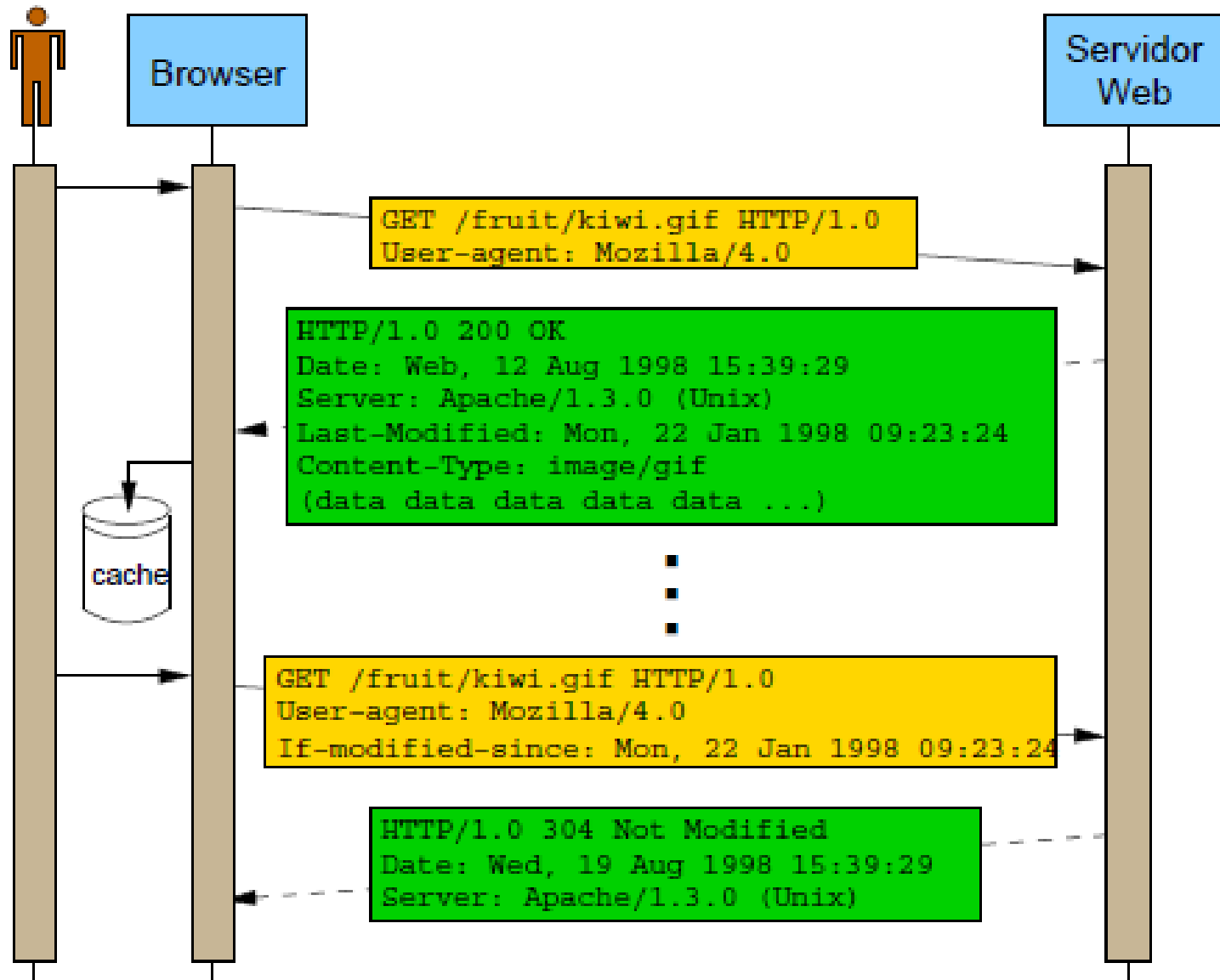
www.toto.com



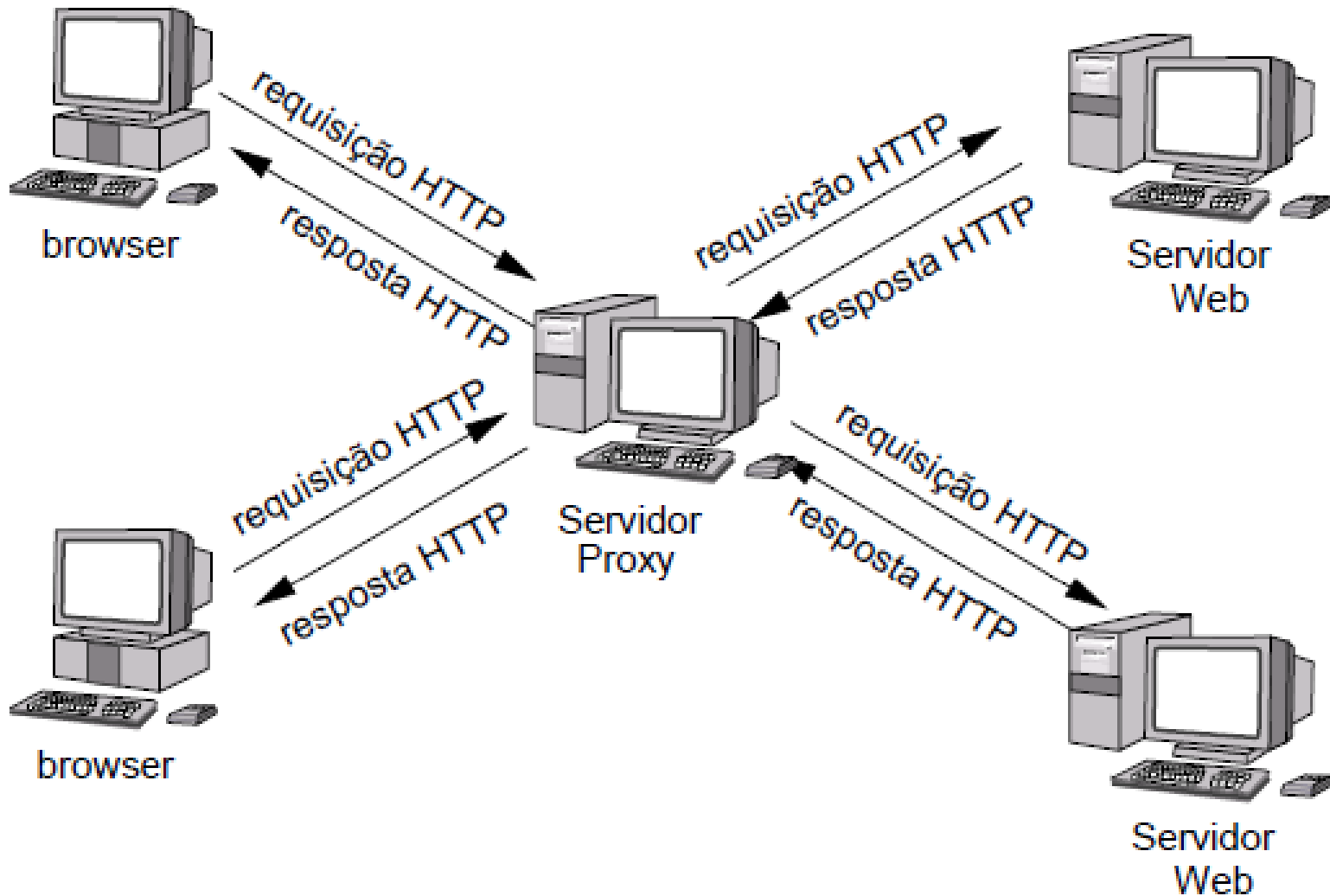
Cache Web

- Grande parte do tráfego de dados (pacotes) na Internet é proveniente da Web
- *Cache Web* é uma estratégia para reduzir o tráfego de dados (e o tempo de resposta para os clientes) devido acesso à Web
- Tipicamente instalado por um ISP (p. ex., em universidades ou ambientes residenciais)
- Pode ser implementada de duas formas:
 - No Cliente HTTP (navegador)
 - Usando GET Condicional
 - Na rede
 - Usando um servidor *proxy web*

GET Condicional (If-modified-since)

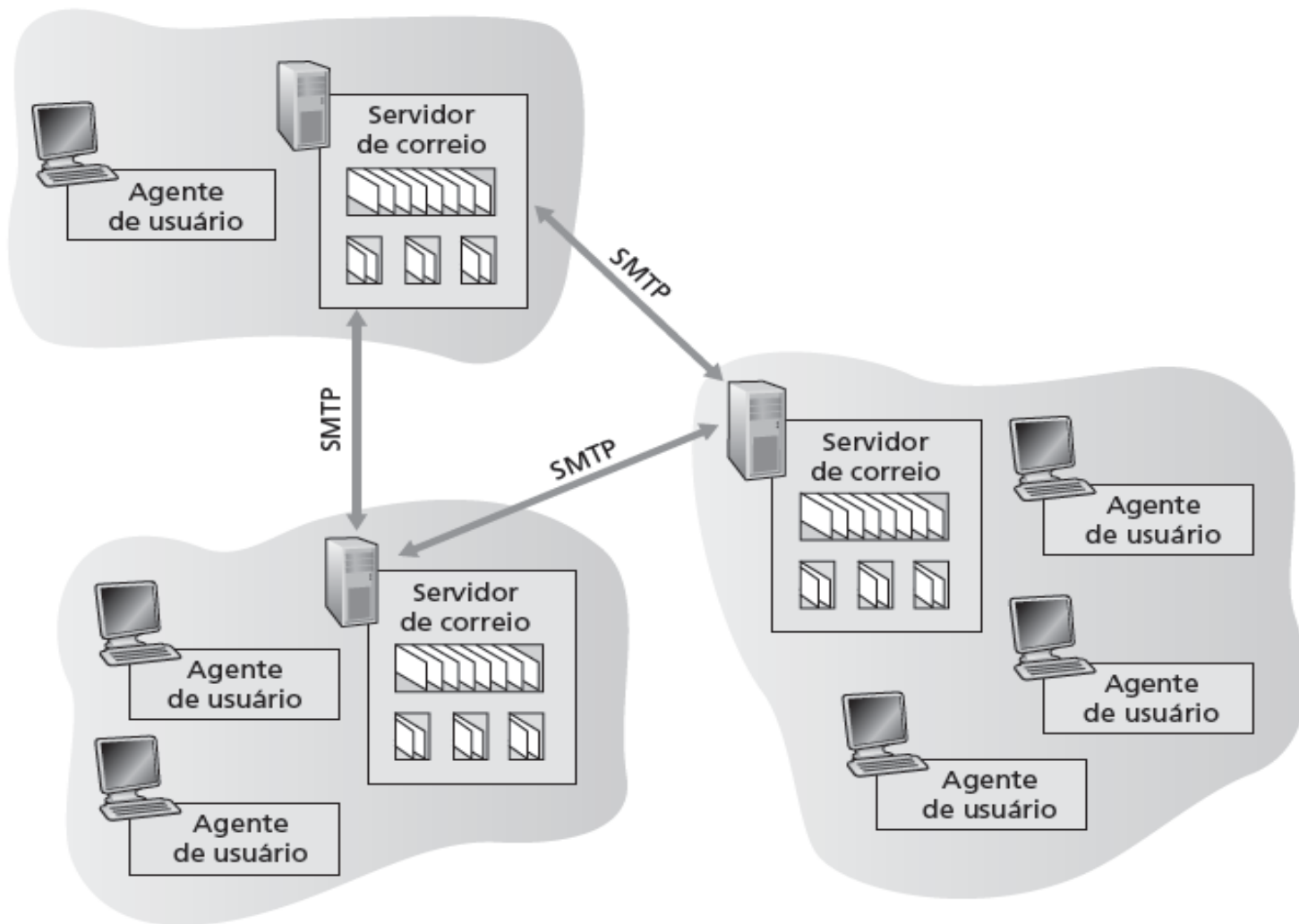


Servidor *Proxy* Web



Correio Eletrônico

Correio Eletrônico



Legenda:



Fila de mensagem de saída



Caixa de entrada do usuário

Correio Eletrônico

- Modo de comunicação assíncrono
- Componentes do sistema de correio eletrônico
 - UAs (*User Agents* – Agentes de Usuário)
 - MTAs (*Message Transfer Agents* – Agentes de Transferência de Mensagens)
- Exemplos de UAs
 - Interface gráfica (Outlook, Thunderbird)
 - Interface texto (mail, pipe)
- MTA é o Servidor de Correio Eletrônico

Protocolo SMTP

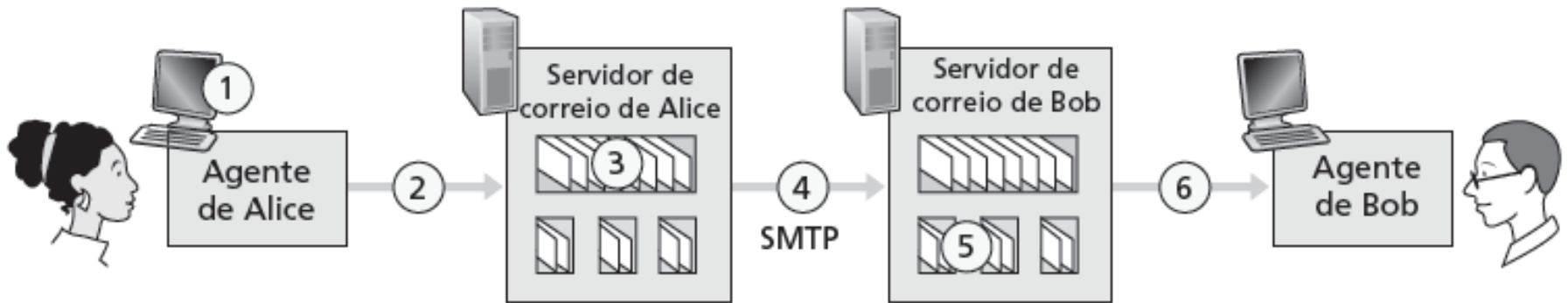
- *Simple Mail Transfer Protocol*
 - Especificado na RFC 821 (1982)
- Trata-se de um protocolo para **envio de informações**
- Utiliza o **TCP (porta 25)** como protocolo (do Nível) de Transporte
- Restringe as mensagens ao formato ASCII, de 7 bits
 - *American Standard Code for Information Interchange*
- No endereço `nome@dominio.com.br`:
 - `nome` representa uma caixa de correio
 - `dominio.com.br` representa o computador onde a caixa postal está localizada

Funcionamento do SMTP

- Uma mensagem de correio eletrônico de Alice para Bob:
 1. Alice usa seu UA para redigir a mensagem para Bob
 2. UA de Alice envia a mensagem para seu MTA (enfileiramento)
 3. *Cliente* SMTP (no MTA de Alice) abre uma conexão TCP (porta 25) para um *Servidor* SMTP (no MTA de Bob)
 4. Após alguma apresentação, *Cliente* SMTP envia a mensagem de Alice na conexão TCP
 5. No MTA de Bob, o *Servidor* SMTP recebe a mensagem e a coloca na caixa postal dele
 6. Bob chama seu UA para ler a mensagem (quando conveniente), usando um protocolo de acesso ao correio, como POP3, IMAP ou HTTP (nos navegadores)

Funcionamento do SMTP

- Uma mensagem de correio eletrônico de Alice para Bob (*graficamente*):



Legenda:



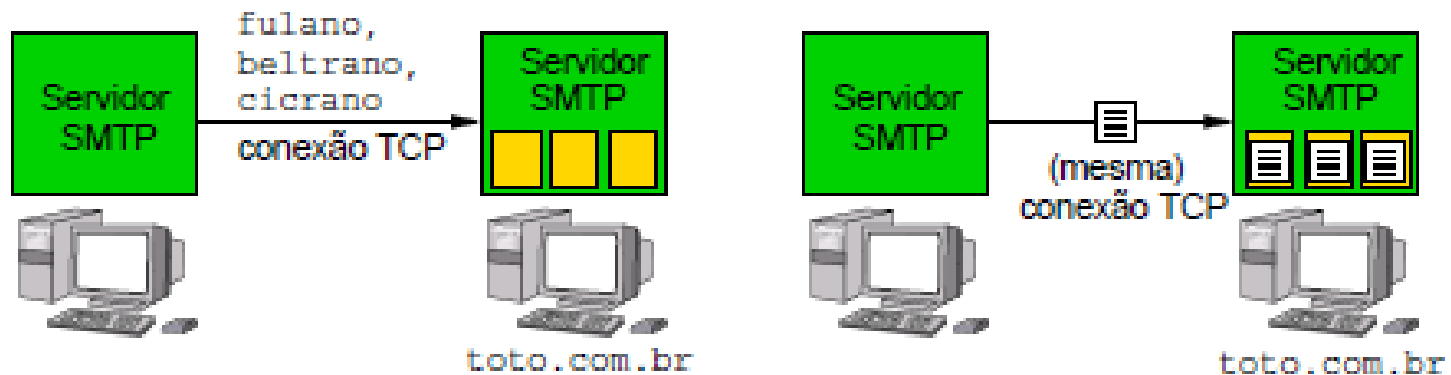
Fila de mensagem



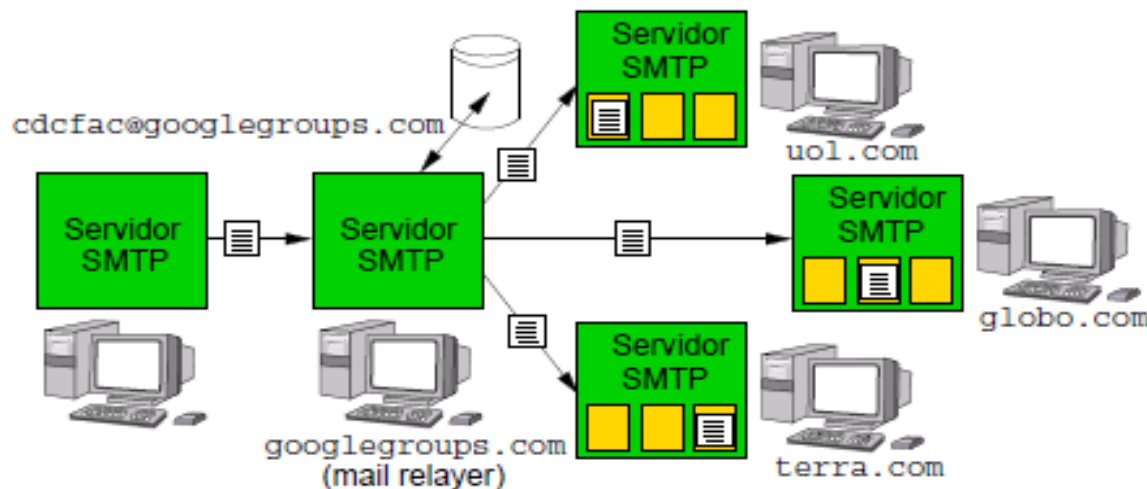
Caixa postal do usuário

Características do SMTP

- Múltiplos destinatários em um *mesmo* servidor



- “Explodidor” de mensagens de correio



Formato das Mensagens SMTP

- Formato básico das mensagens de correio é definido na RFC 822
- Conteúdo era restrito (inicialmente) a mensagens codificadas em ASCII
- Exemplo de mensagem de correio:

From: alice@alpha.fr

To: bob@bravo.edu

Subject: Quadro de Monet.

Exemplo de Troca de Mensagens

Cliente SMTP (C:) no MTA (alpha.fr) de Alice envia dados para Servidor SMTP (S:) no MTA (bravo.edu) de Bob

S: 220 bravo.edu

C: HELO alpha.fr

S: 250 Hello alpha.fr, pleased to meet you

C: MAIL FROM: <alice@alpha.fr>

S: 250 alice@alpha.fr ... Sender ok

C: RCPT TO: <bob@bravo.edu>

S: 250 bob@bravo.edu ... Recipient ok

C: DATA

S: 354 Enter mail, end with "." on a line by itself

C: Bom dia, Bob. Tudo bem contigo?

C: .

S: 250 Message accepted for delivery

C: QUIT

S: 221 bravo.edu closing connection

Cabeçalhos MIME

- MIME (*Multi purpose Internet Mail Extensions*)
- Cabeçalhos definidos nas RFCs 2045 e 2046
- Usados em mensagens multimídia ou mensagens não codificadas em ASCII
- Exemplo:

From: alice@alpha.fr

To: bob@bravo.edu

Subject: Quadro de Monet.

MIME-Version: 1.0

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

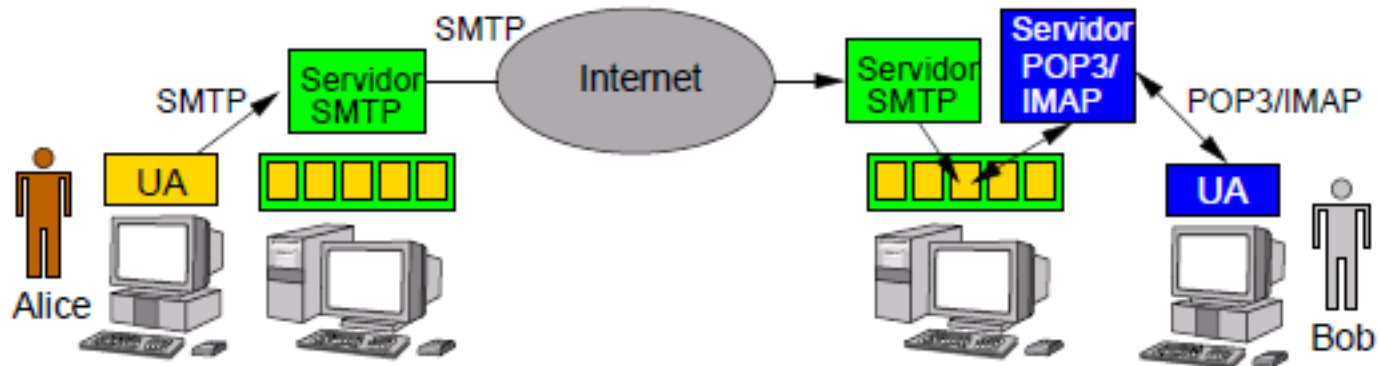
(dados codificados em base64....

.....

...(dados codificados em base64)

Entrega final da mensagem de correio

- Considera o modo de comunicação assíncrono (conectividade sob demanda dos usuários)



- Protocolos usados na entrega final
 - POP3 (*Post Office Protocol version 3*)
 - IMAP (*Internet Message Access Protocol*)

Protocolo POP3

- Definido na RFC 1939
- Usa o TCP como protocolo (do nível) de transporte
 - Porta 110 TCP
- UAs descarregam usualmente todas as mensagens e esvaziam a caixa de correio

POP3: Exemplo de Troca de Mensagens

```
telnet mailServer 110
+OK POP3 server ready
user alice
+OK
pass toto
+OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (bla bla .....
S: .....
S: ..... bla)
S: .
C: dele 1
S: (bla bla .....
S: .....
S: ..... bla)
S: .
C: retr 2
S: (bla bla .....
S: .....
S: ..... bla)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing| off
```

Protocolo IMAP

- Definido na RFC 2060
- Usa o TCP como protocolo (do nível) de transporte
 - Porta 143 TCP
- Assume que as mensagens permanecerão indefinidamente no servidor
- Permite criar, destruir e manipular várias caixas de correio no servidor

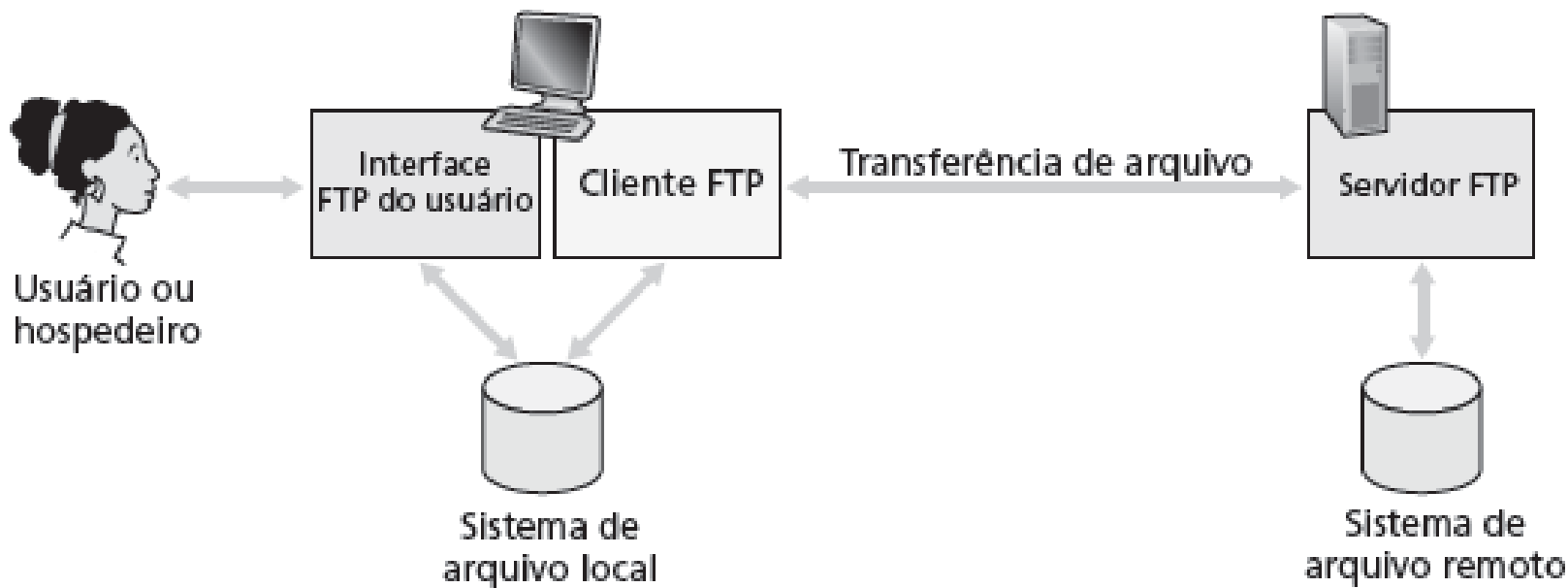
Webmail

- UA é o navegador do cliente
- Exemplos de serviços Webmail
 - Hotmail, Yahoo! Mail, Gmail, etc.
- Mensagens são **enviadas** para os servidores de email e **retiradas** deles usando o protocolo HTTP
- Os servidores (MTAs), contudo, continuam a se comunicar entre si usando o protocolo SMTP

Transferência de arquivos

Transferência de Arquivos

- FTP (*File Transfer Protocol*)
 - Especifica a transferência de arquivos de um hospedeiro local para um servidor remoto
 - Definido na RFC 959

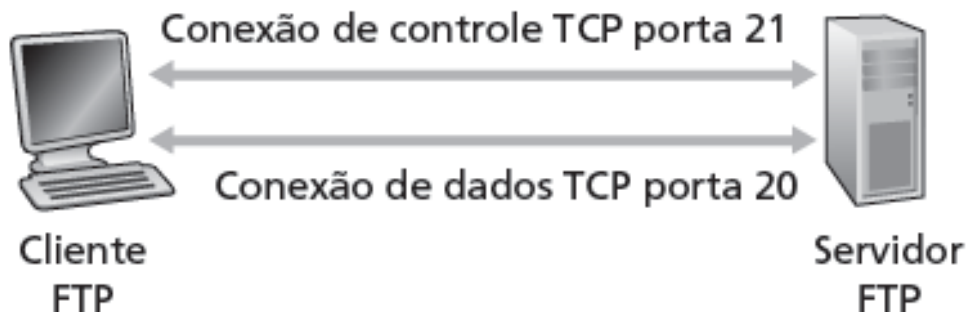


Transferência de Arquivos

- Alice quer transferir arquivo do seu hospedeiro local para um servidor remoto:
 1. Alice interage com o FTP usando agente de usuário FTP, fornecendo nome do hospedeiro remoto
 2. Cliente FTP do hospedeiro local estabelece conexão TCP com o servidor FTP (remoto)
 3. Alice fornece sua identificação e senha (via conexão TCP já aberta com o servidor)
 4. Após autorizado pelo servidor, Alice copia um ou mais arquivos armazenados no sistema de arquivo local para o sistema de arquivo remoto (ou vice-versa)

Transferência de Arquivos

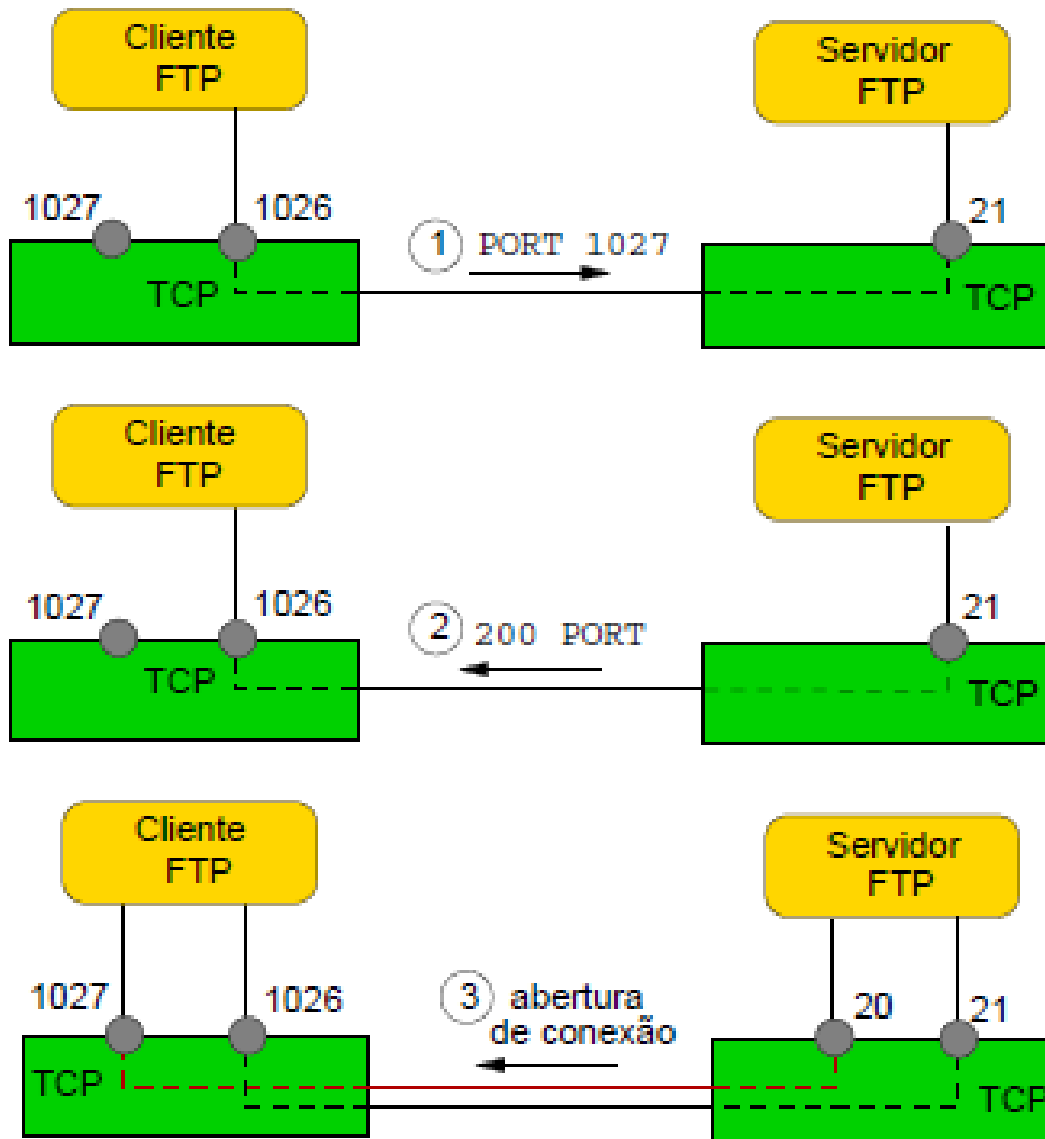
- Protocolo FTP usa **dois tipos de conexão TCP**:
 - Uma conexão de **controle** (TCP persistente, porta 21) durante toda a sessão FTP
 - Uma conexão de **dados** (TCP transiente, porta 20) para cada transferência de arquivo
- Conexão de controle separada da conexão de dados (mensagens de controle “fora da banda”)



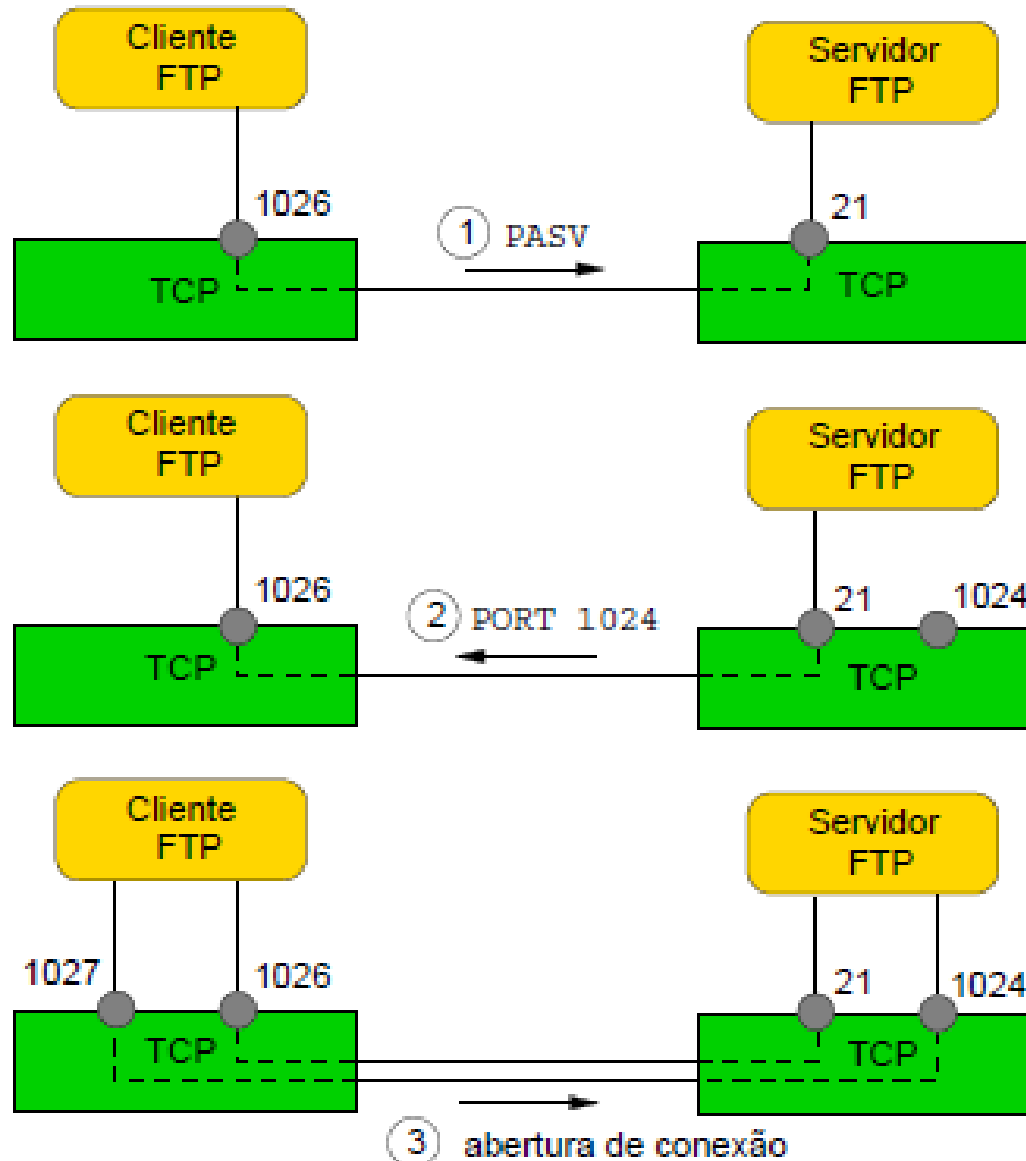
Modos de Operação FTP

- Modo de operação corresponde ao comportamento do **Servidor FTP** na abertura da **conexão de dados**
 - Pode ser ativo (*padrão*) ou passivo
 - No modo ativo, o Servidor FTP abre a conexão de dados TCP com o cliente FTP
 - No modo passivo, o Cliente FTP abre a conexão de dados com o servidor FTP
- O FTP **mantém o estado** do cliente
 - Monitora o diretório corrente do usuário no diretório remoto no servidor

FTP: Modo Ativo



FTP: Modo Passivo



Endereçamento de máquinas

Endereçamento de máquinas

- Endereços IP
 - Usados pelo protocolo IP (*Internet Protocol*) no nível de rede para realizar o serviço de roteamento
 - Possuem comprimento fixo e são estruturados hierarquicamente
- Humanos preferem nomes, enquanto equipamentos intermediários na rede (roteadores) preferem endereços IP
 - Como acomodar essas preferências?

Serviço DNS

- DNS (*Domain Name Service* – Sistema de Nomes de Domínio)
 - Serviço de diretório que mapeia *hostnames* para endereços IP na Internet
 - Especificado nas RFCs 1034 e 1035
 - Representa um banco de dados distribuído, seguindo uma hierarquia de servidores de nomes (servidores DNS), e um protocolo de nível de aplicação que permite que hospedeiros consultem o banco de dados distribuído
 - Protocolo DNS usa a **porta 53** do **protocolo UDP** (*User Datagram Protocol*) no nível de transporte

Funcionamento do DNS

- Usado por outros protocolos de nível de aplicação (p. ex., HTTP, SMTP, FTP)
- Exemplo: navegador requisita o URL `www.ifrn.edu.br`
 1. Hospedeiro do usuário executa lado cliente do DNS
 2. Navegador (cliente HTTP) extrai URL (*hostname*) da requisição HTTP e o repassa ao lado cliente da aplicação DNS
 3. Cliente DNS envia consulta contendo o *hostname* ao servidor DNS
 4. Servidor DNS envia resposta ao cliente DNS contendo o endereço IP associado ao *hostname*
 5. Cliente DNS repassa o endereço IP ao navegador, que pode então abrir uma conexão TCP com o servidor HTTP localizado naquele endereço IP

Funcionamento do DNS

- Note que o uso do DNS acrescenta atraso (devido tráfego de mais mensagens de controle) em uma sessão HTTP
 - Passos 3 e 4 do exemplo anterior

Conclusão

- Protocolo HTTP é usado na transferência de objetos Web
 - Mensagens: Requisição HTTP e Resposta HTTP
 - *Cache Web* busca reduzir o tráfego de dados
 - GET Condicional (no cliente HTTP)
 - Servidor Proxy (na rede)
- Protocolo SMTP: correio eletrônico
- Protocolo FTP: transferência de arquivos
- Protocolo DNS: tradução de endereços para nomes

Perguntas?

helber.silva@ifrn.edu.br