

Redes de Computadores

Camada de Aplicação

HTTP – FTP – SMTP

Mensagem de Requisição HTTP

linha de pedido
(comandos GET
, POST, HEAD)

linhas de
cabeçalho

```
GET /~fabio.nelson/redes.html HTTP/1.1
Host: www.univasf.edu.br
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: en
```

*Carriage return,
line feed*
indica fim da
mensagem

(extra carriage return, line feed)

Mensagem de Resposta HTTP

linha de status
(protocolo,
código de status,
frase de status)

linhas de
cabeçalho

dados, e.x.,
arquivo html

```
HTTP/1.1 200 OK
Date: Fri, 04 Apr 2008 01:18:26 GMT
Server: Apache/1.3.37 (Unix) PHP/5.2.1
Last-Modified: Wed, 02 Apr 2008 14:52:58 GMT
ETag: "748051-12d7-47f39dca"
Accept-Ranges: bytes
Content-Length: 4823
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

dados, dados, dados, dados, dados, dados...

Código de Status das Mensagens HTTP

● 200 OK

- requisição bem-sucedida e a informação é entregue com a resposta

● 301 Moved Permanently

- objeto requisitado removido, nova localização informada adiante no cabeçalho Location:, neste ponto é informado a nova URL

● 400 Bad Request

- requisição não entendida pelo servidor

● 404 Not Found

- O objeto requisitado não encontrado no servidor

● 505 HTTP Version Not Supported

- Versão do protocolo HTTP não suportada pelo servidor

Rede de Computadores

Métodos HTTP

● GET

- Requisita um objeto do servidor;

● POST

- Enviar dados para um servidor (ex: formulário);

● HEAD

- Confirma a existência de um objeto no servidor;

● PUT

- Envia um objeto para o servidor;

● DELETE

- Apaga um objeto no servidor.

Solução para um Protocolo *Stateless*

● Cookie

- Utilizado para monitoramento e/ou acompanhamento do usuário numa conexão;
- Pode ser usado para criar uma camada de sessão de usuário;

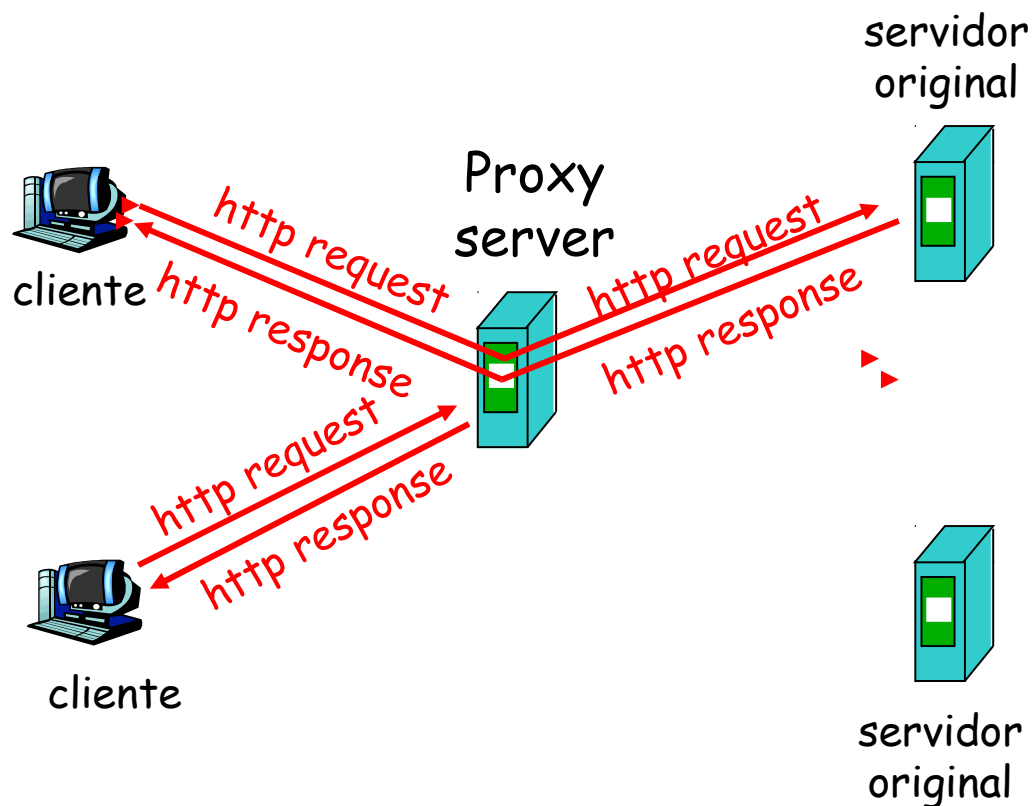
● Um cookie possui quatro componentes:

- Uma linha de cabeçalho de cookie na resposta HTTP (SET COOKIE: número);
- Uma linha de cabeçalho de cookie na mensagem de requisição HTTP (COOKIE: número);
- Um arquivo de cookies mantido pelo sistema e gerenciado pelo browser;
- Um banco de dados no site da web.

SERVIDOR PROXY

- Atende a requisições HTTP no lugar do servidor Web de origem;
- Funciona como cliente e servidor ao mesmo tempo;
- Diminui o tempo de resposta e o tráfego no enlace de acesso a internet;
- Utiliza um método chamado GET condicional
 - If-modifid-since: data do objeto armazenado
 - 302 Not Modified

Rede de Computadores



Rede de Computadores

● Sem Cache Web:

$$15 \text{ (req/seg)} * (100\text{k/req}) / (10 \text{ Mbps}) = 0,15$$

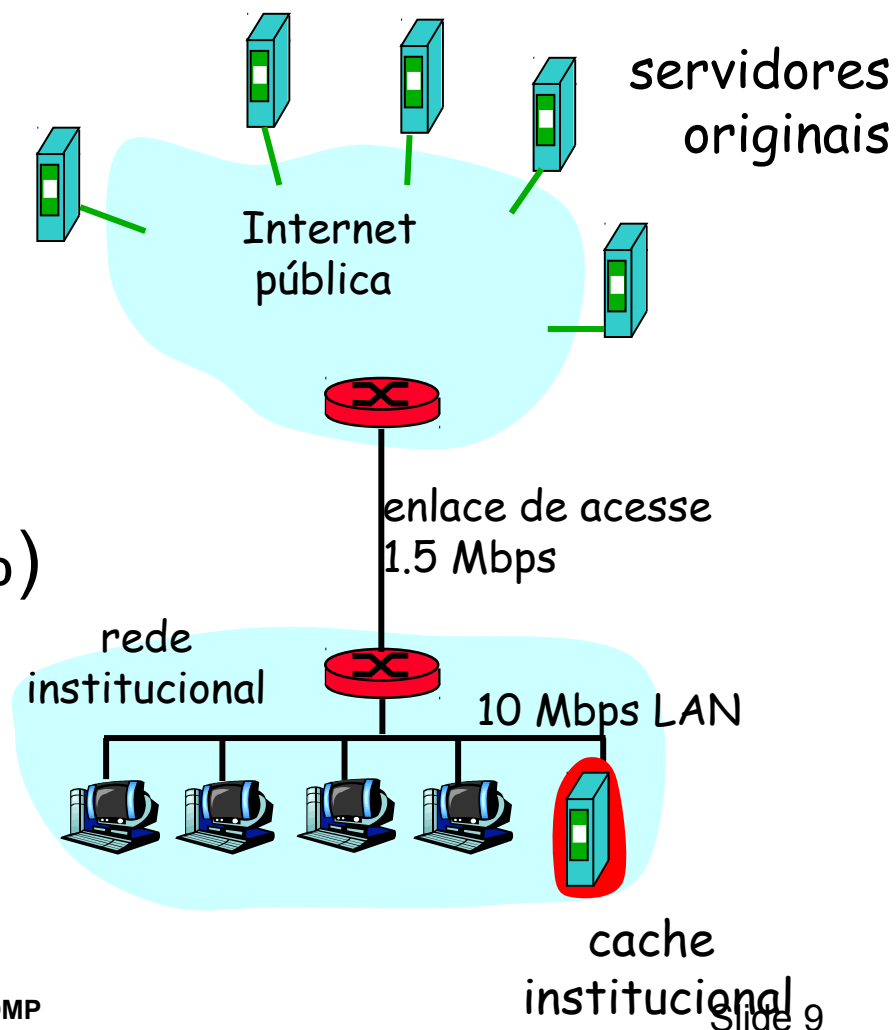
$$15 \text{ (req/seg)} * (100\text{k/req}) / (1,5 \text{ Mbps}) = 1$$

Atraso Internet aprox. 2s

● Com Cache Web (at. 40%)

$$\text{Tráfego: } 0,6 * 1 = 0,6$$

$$0,4 * 0,01\text{s} + 0,6 * 2,01\text{s} = \text{aprox } 1,2\text{s}$$



Protocolo FTP

- Protocolo de transferência de arquivos de e para o computador remoto
- Comunicação no modelo cliente/servidor:
 - Cliente: inicia a conexão
 - Servidor: host remoto
- RFC 959, porta 21 (controle) e porta 20 (dados)
- Trabalha com duas conexões:
 - Conexão de controle: permanente, utilizada para informações de controle; ex: id, senha, comandos...
 - Conexão de dados: temporária, utilizada para transferência de dados.

Protocolo FTP

- Envia as suas informações de controle fora da banda;
- Durante toda a sessão FTP o servidor mantém informações de estado sobre o usuário;
- A conexão de controle é associada a um usuário;
- Por manter o estado da sessão, o servidor tem uma limitação alta de usuários conectados simultaneamente;

Comandos Usuais FTP

● **USER username**

- Envia a identificação do usuário

● **PASS password**

- Envia a senha do usuário

● **LIST**

- Pede a listagem do diretório corrente, a lista é enviada por uma conexão de dados

● **RETR filename**

- Pede que o servidor envie um arquivo <filename>

● **STOR filename**

- Envia um arquivo <filename> para o servidor

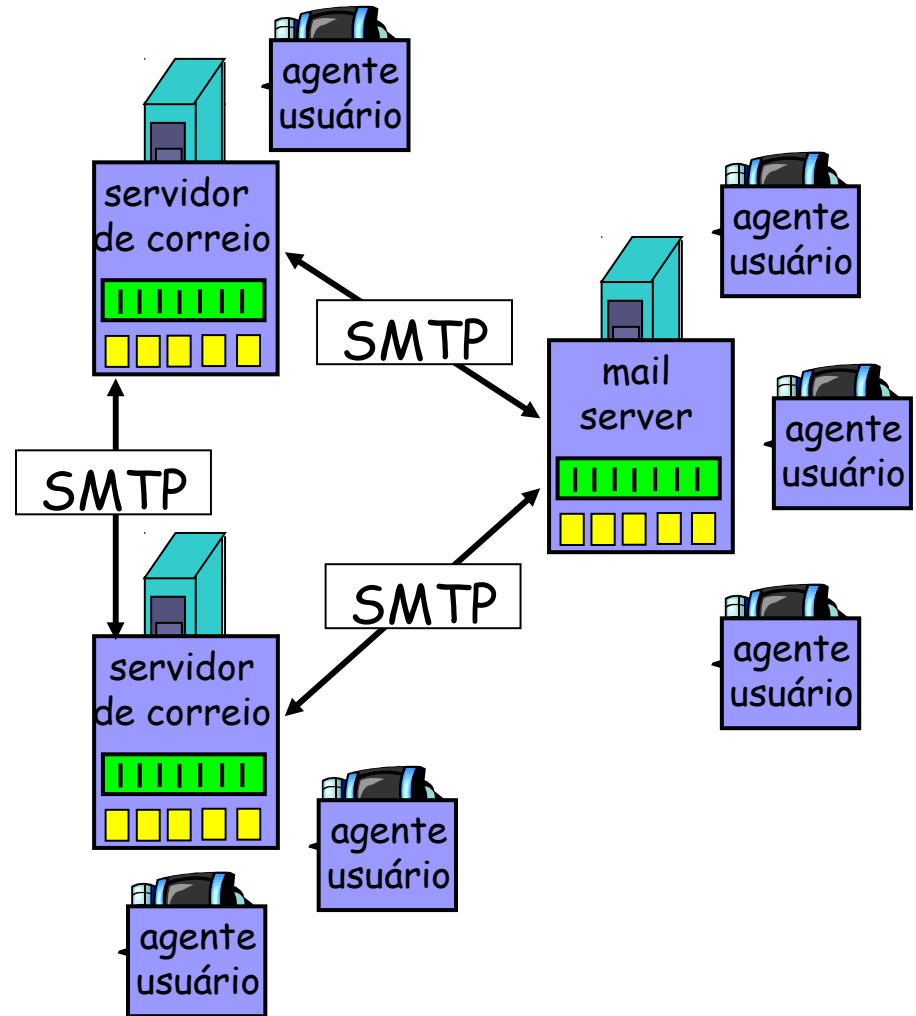
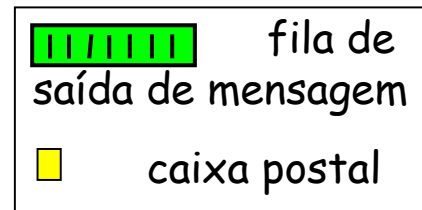
Respostas Usuais FTP

- **331 Username OK, password required**
 - Nome do usuário OK, senha requisitada
- **125 Data connection already open; starting transfer**
 - Conexão de dados aberta, iniciando a transferencia
- **425 Can't open data connection**
 - Não é possível abrir a conexão de dados
- **452 Error writing file**
 - Erro ao escrever o arquivo

Rede de Computadores

Correio Eletrônico

- Servidor de Correio
- Agente Usuário
- Protocolo SMTP
- Caixa Postal



Protocolo SMTP

- Definido no RFC 2821, usa a porta 25;
- As mensagens são em código ASCII de 7 bits;
- Utiliza o protocolo TCP para transporte;
- Usualmente não são utilizados servidores intermediários para entregar a correspondência;

Protocolo SMTP

● Etapas para envio de um e-mail:

- O cliente SMTP faz uma conexão TCP com um servidor SMTP;
- Feita a conexão é feito o *handshaking* (apresentação, identificação, endereço de entrega e origem);
- É enviada a mensagem;
- O processo se repete para cada e-mail.

Rede de Computadores

Comunicação SMTP

```
220 mail.univasf.edu.br ESMTP
HELO smtp.bol.com.br
250 mail.univasf.edu.br
MAIL FROM: <fabio.nelson@bol.com.br>
250 ok
RCPT TO <fabio.nelson@univasf.edu.br>
502 unimplemented (#5.5.1)
RCPT TO <suporte@univasf.edu.br>
250 ok
DATA
354 go ahead
Este eh um teste da turam de redes
.
250 ok 1207874477 qp 13407
QUIT
221 mail.univasf.edu.br
```

Conexão ao host perdida.

Comunicação SMTP

- São definidas no RFC 822;
- A cada comando enviado é recebida uma resposta;
- É utilizada conexões persistentes na comunicação;
- Comandos usuais:
 - HELO <endereço servidor>
 - MAIL FROM: <endereço de origem>
 - RCPT TO: <endereço de destino>
 - DATA
 - .
 - QUIT

SMTP versus HTTP

● Quanto a comunicação:

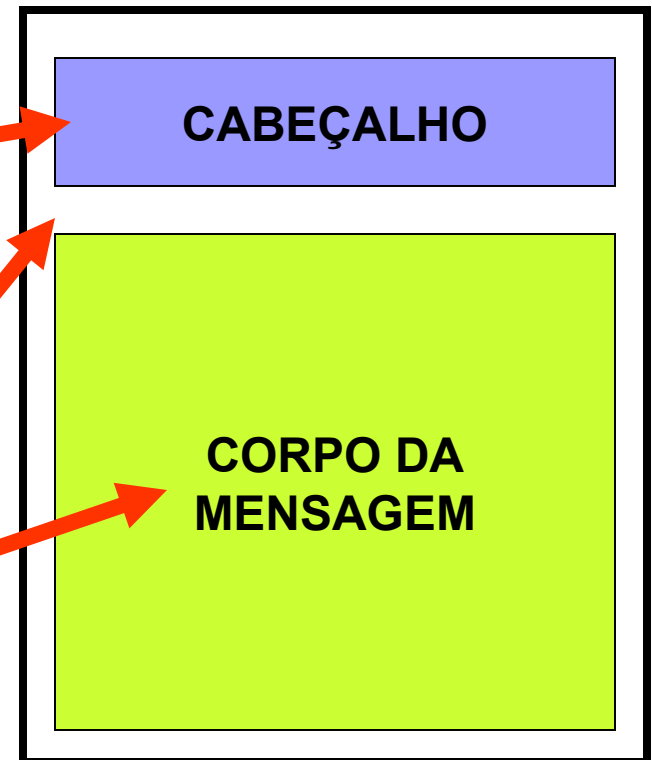
- HTTP é um protocolo de recuperação de informações (*pull protocol*);
 - A conexão é feita por quem quer receber o arquivo.
- SMTP é um protocolo de envio de informações (*push protocol*);
 - A conexão é feita por quem quer enviar o arquivo.

● Quanto ao envio de dados:

- SMTP exige que a mensagem e os dados enviados estejam no formato ASCII de 7 bits, todos os objetos componentes da mensagem são enviados de uma só vez;
- HTTP encapsula cada objeto em sua própria mensagem.

Formato das Mensagens

- São definidas no RFC 822;
- SMTP: protocolo para envio de mensagens de e-mail;
- Cabeçalho:
 - From
 - To
 - Subject
 - Diferente dos comandos SMTP
- Uma linha em branco separa o cabeçalho do corpo da mensagem.
- Corpo da Mensagem
 - Somente caracteres ASCII de 7 bits



Extensão MIME

- MIME – *Multipurpose Internet Mail Extensions* (extensões multiuso do correio da internet), RFC 2045 e RFC 2046 (extensões do RFC 822);
- Utilizado para enviar conteúdo que não seja no formato ASCII de 7bits;
- São cabeçalhos extras adicionados ao já existentes do SMTP;

Extensão MIME

- Permite o envio de arquivos diversos e informa a ação relacionada aos mesmos;
- Comandos usuais:
 - **Content-Type:** permite a realização de uma ação específica pelo agente usuário destinatário com a mensagem;
 - Podem ser: text, video, application, audio, image.
 - **Content-Transfer-Encoding:** informa o tipo de codificação utilizada

Comunicação Utilizando MIME

From: fabio.nelson@bol.com.br

To: fabio.nelson@univasf.edu.br

Subject: Foto da turma de redes

MIME-Version: 1.0

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

base64 encoded data

.....

.....

..... base64 encoded data

Cabeçalho *Received*

Received: from bol.com.br by univasf.edu.br; 10 Mar 2010 15:27:50 GMT

From: fabio.nelson@bol.com.br

To: fabio.nelson@univasf.edu.br

Subject: Foto da turma de redes

MIME-Version: 1.0

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

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