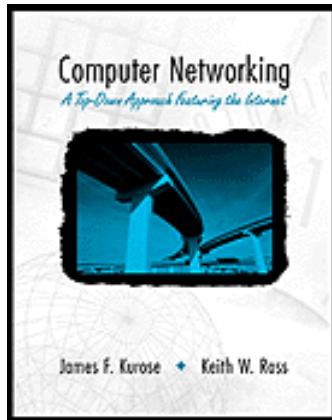


# Redes de Computadores

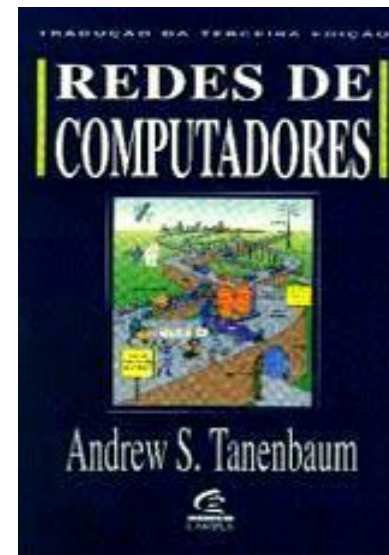
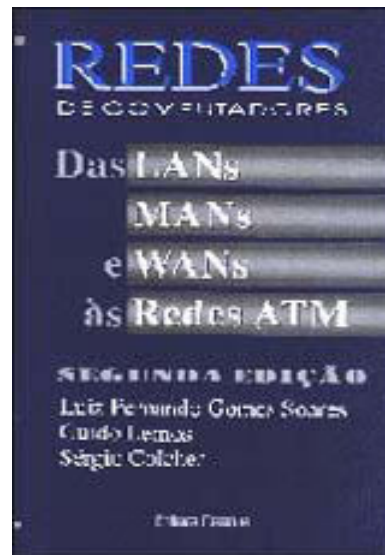
Prof. José Augusto Suruagy Monteiro  
suruagy@unifacs.br  
[www.nuperc.unifacs.br/suruagy/redes](http://www.nuperc.unifacs.br/suruagy/redes)

# Livro Texto:



- Kurose, J., Ross, K., Computer Networking: A Top-Down Approach - featuring the Internet, Addison-Wesley, 2001

## Bibliografia Complementar:



# WWW: algum jargão

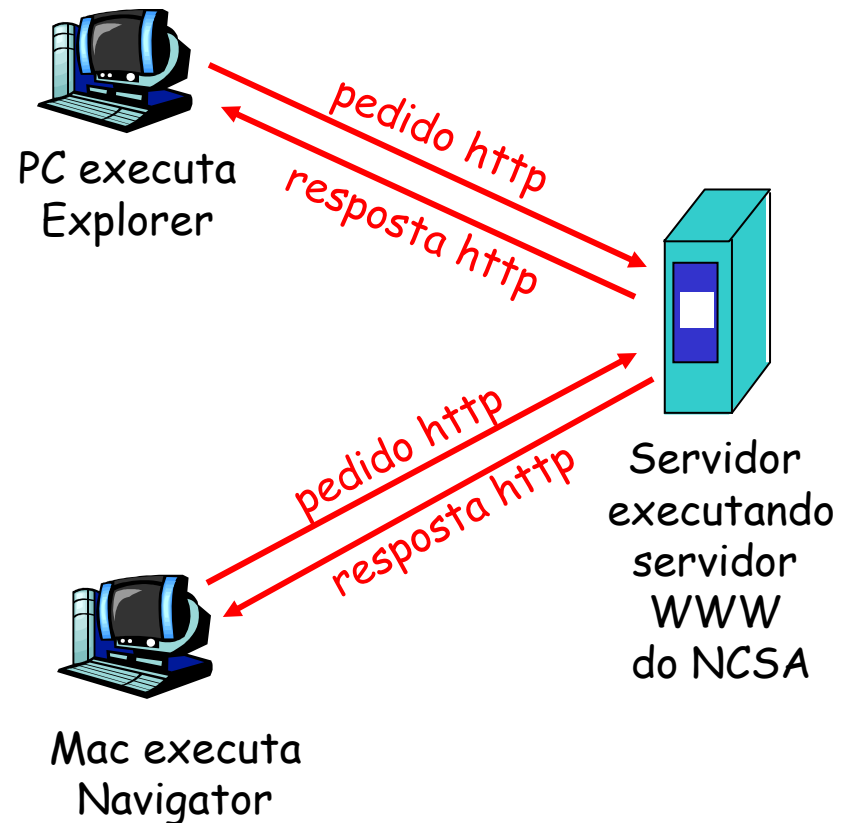
- ❑ Página WWW:
  - consiste de "objetos"
  - endereçada por uma URL
- ❑ Quase todas as páginas WWW consistem de:
  - página base HTML, e
  - vários objetos referenciados.
- ❑ URL tem duas partes: nome de hospedeiro, e nome de caminho:
- ❑ Agente de usuário para WWW se chama de browser:
  - MS Internet Explorer
  - Netscape Communicator
- ❑ Servidor para WWW se chama "servidor WWW":
  - Apache (domínio público)
  - MS Internet Information Server (IIS)

[www.univ.br/algum-depto/pic.gif](http://www.univ.br/algum-depto/pic.gif)

# WWW: o protocolo http

## http: hypertext transfer protocol

- ❑ protocolo da camada de aplicação para WWW
- ❑ modelo cliente/servidor
  - *cliente*: browser que pede, recebe, "visualiza" objetos WWW
  - *servidor*: servidor WWW envia objetos em resposta a pedidos
- ❑ http1.0: RFC 1945
- ❑ http1.1: RFC 2068



# Mais sobre o protocolo http

## http: serviço de transporte TCP:

- ❑ cliente inicia conexão TCP (cria socket) ao servidor, porta 80
- ❑ servidor aceita conexão TCP do cliente
- ❑ mensagens http (mensagens do protocolo da camada de apl) trocadas entre browser (cliente http) e servidor WWW (servidor http)
- ❑ encerra conexão TCP

## http é "sem estado"

- ❑ servidor não mantém informação sobre pedidos anteriores do cliente

### Nota

#### Protocolos que mantêm "estado" são complexos!

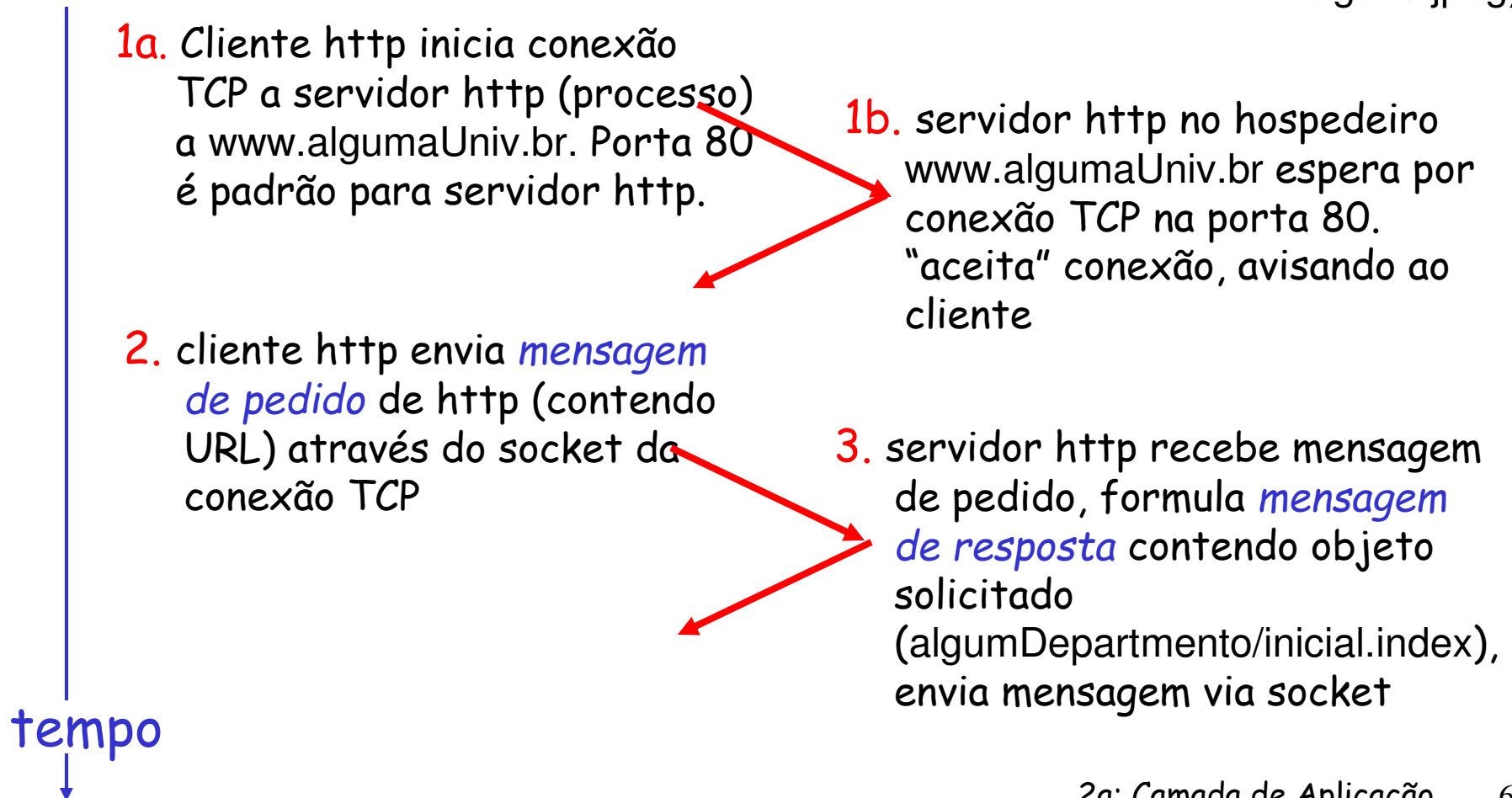
- ❑ história passada (estado) tem que ser guardada
- ❑ Caso caia servidor/cliente, suas visões do "estado" podem ser inconsistentes, devem ser reconciliadas

# Exemplo de http

Supomos que usuário digita a URL

www.algumaUniv.br/algumDepartamento/inicial.index

(contém texto,  
referências a 10  
imagens jpeg)



# Exemplo de http (cont.)

4. servidor http encerra conexão TCP .

5. cliente http recebe mensagem de resposta contendo arquivo html, visualiza html.  
Analisando arquivo html, encontra 10 objetos jpeg referenciados

6. Passos 1 a 5 repetidos para cada um dos 10 objetos jpeg

tempo

## Conexões não persistente and persistente

### Não persistente

- ❑ HTTP/1.0
- ❑ servidor analisa pedido, responde, e encerra conexão TCP
- ❑ 2 RTTs para trazer cada objeto (RTT=round trip time)
- ❑ transferência de cada objeto sofre de partida lenta

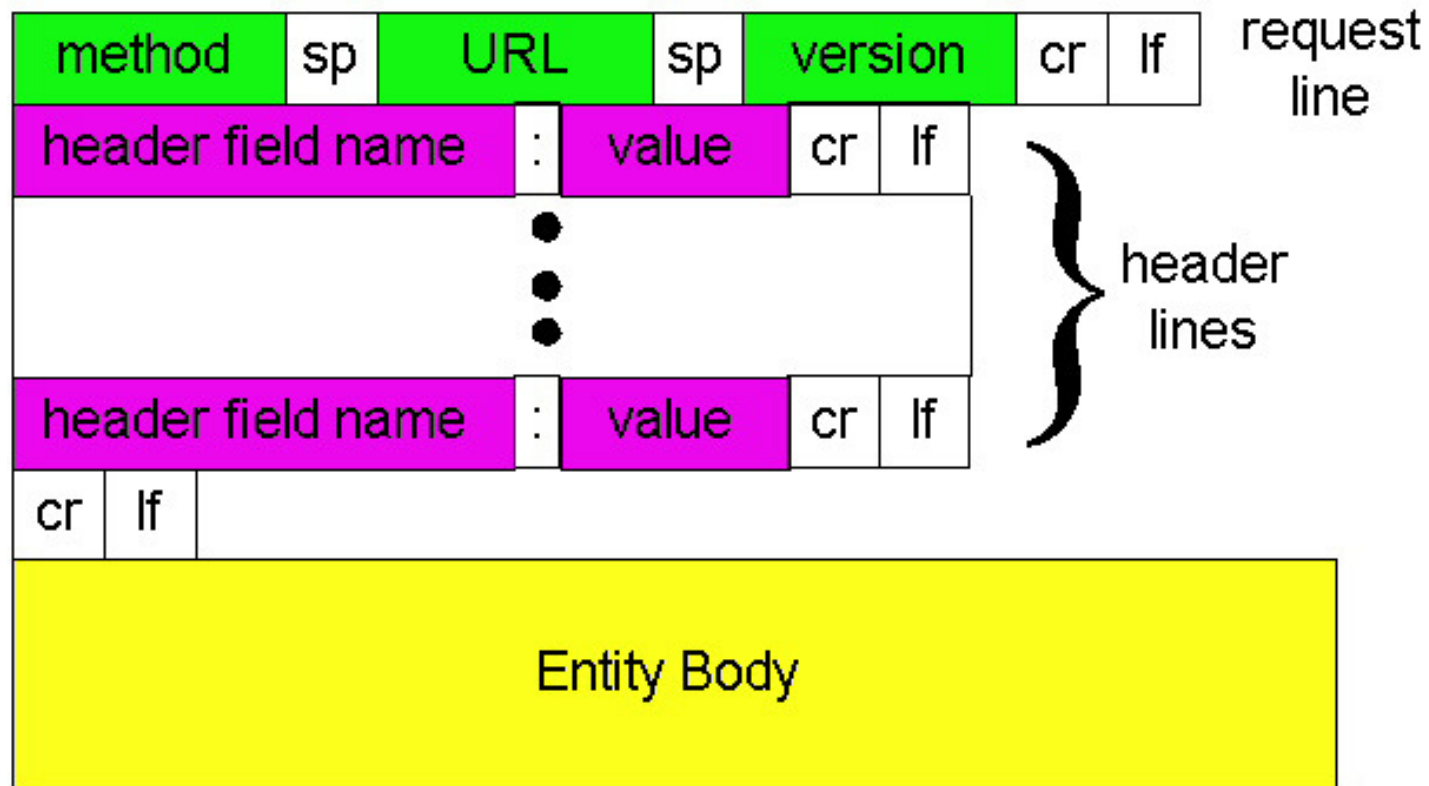
A maioria de browsers 1.0 usa conexões TCP paralelas.

### Persistente

- ❑ default for HTTP/1.1
- ❑ na mesma conexão TCP: servidor analisa pedido, responde, analisa novo pedido,...
- ❑ Cliente envia pedidos para todos objetos referenciados assim que recebe o HTML base .
- ❑ Menos RTTs e menos partida lenta.



## mensagem de pedido http: formato geral



# formato de mensagem http: pedido

- ❑ Dois tipos de mensagem http: *pedido, resposta*
- ❑ *mensagem de pedido http:*
  - ASCII (formato legível por pessoas)

linha do pedido  
(comandos GET,  
POST, HEAD)

linhas do  
cabeçalho

```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

Carriage return,  
line feed  
indicam fim  
de mensagem

(carriage return (CR), line feed(LF) adicionais)

# formato de mensagem http: resposta

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

linhas de  
cabeçalho

dados, p.ex.,  
arquivo html  
solicitado

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

dados dados dados dados ...

# códigos de status da resposta http

Na primeira linha da mensagem de resposta servidor->cliente. Alguns códigos típicos:

## **200 OK**

- sucesso, objeto pedido segue mais adiante nesta mensagem

## **301 Moved Permanently**

- objeto pedido mudou de lugar, nova localização especificado mais adiante nesta mensagem (Location:)

## **400 Bad Request**

- mensagem de pedido não entendida pelo servidor

## **404 Not Found**

- documento pedido não se encontra neste servidor

## **505 HTTP Version Not Supported**

- versão de http do pedido não usada por este servidor

# Experimente você com http (do lado cliente)

1. Use cliente telnet para seu servidor WWW favorito:

**telnet www.ic.uff.br 80**

Abre conexão TCP para a porta 80 (porta padrão do servidor http) a www.ic.uff.br. Qualquer coisa digitada é enviada para a porta 80 do www.ic.uff.br

2. Digite um pedido GET http:

**GET /~michael/index.html HTTP/1.0**

Digitando isto (deve teclar ENTER duas vezes), está enviando este pedido GET mínimo (porém completo) ao servidor http

3. Examine a mensagem de resposta enviada pelo servidor http !