

## Redes de Computadores

Prof. José Augusto Suruagy Monteiro  
suruagy@unifacs.br  
www.nuperc.unifacs.br/suruagy/redes

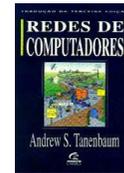
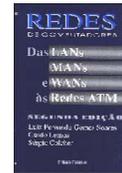
2a: Camada de Aplicação 1

## Livro Texto:



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

Bibliografia Complementar:



2a: Camada de Aplicação 2

## 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)

2a: Camada de Aplicação 3

## 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



2a: Camada de Aplicação 4

## 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

2a: Camada de Aplicação 5

## Exemplo de http

Supomos que usuário digita a URL [www.algumaUniv.br/algumDepartamento/inicial.index](http://www.algumaUniv.br/algumDepartamento/inicial.index) (contém texto, referências a 10 imagens jpeg)

- 1a. Cliente http inicia conexão TCP a servidor http (processo) a [www.algumaUniv.br](http://www.algumaUniv.br). Porta 80 é padrão para servidor http.
- 1b. servidor http no hospedeiro [www.algumaUniv.br](http://www.algumaUniv.br) espera por conexão TCP na porta 80. "aceita" conexão, avisando ao cliente
2. cliente http envia *mensagem de pedido* de http (contendo URL) através do socket da conexão TCP
3. servidor http recebe mensagem de pedido, formula *mensagem de resposta* contendo objeto solicitado ([algumDepartamento/inicial.index](http://algumDepartamento/inicial.index)), envia mensagem via socket

tempo ↓

2a: Camada de Aplicação 6

## Exemplo de http (cont.)

- cliente http recebe mensagem de resposta contendo arquivo html, visualiza html. Analisando arquivo html, encontra 10 objetos jpeg referenciados
- Passos 1 a 5 repetidos para cada um dos 10 objetos jpeg
- servidor http encerra conexão TCP.

tempo

2a: Camada de Aplicação 7

## 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

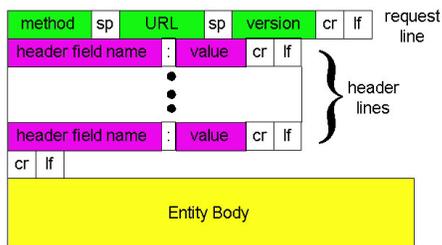
A maioria de browsers 1.0 usa conexões TCP paralelas.

### 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.

2a: Camada de Aplicação 8

## mensagem de pedido http: formato geral



2a: Camada de Aplicação 9

## formato de mensagem http: pedido

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

linhas do pedido (comandos GET, POST, HEAD) → GET /somedir/page.html HTTP/1.0  
linhas do cabeçalho → 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)

2a: Camada de Aplicação 10

## formato de mensagem http: resposta

linha de status (protocolo, código de status, frase de status) → HTTP/1.0 200 OK  
linhas de cabeçalho → 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, p.ex. arquivo html solicitado → dados dados dados dados ...

2a: Camada de Aplicação 11

## 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

2a: Camada de Aplicação 12

## 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!