Curso de Redes de Computadores

Vinicius F. Caridá

UEMG
FRUTAL

Os principais protocolos de aplicação

- O HTTP (Protocolo de Transferência de Hipertexto) é um protocolo da camada de aplicação, ele é implementado em dois programas, um cliente e outro servidor. Os dois programas, executados em sistemas finais diferentes, conversam um com o outro por meio da troca de mensagens HTTP. O protocolo define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam.

- Uma **página Web** é constituída de objetos que são simplesmente arquivos que se podem acessar com um único URL. A maioria das páginas Web é constituída de um arquivo-base HTML e diversos objetos referenciados. Cada URL tem dois componentes, o nome do hospedeiro do servidor que abriga o objeto e o nome do caminho do objeto.

- Um **browser** é um agente de usuário para a Web, apresenta a página requisitada ao usuário e fornece numerosas características de navegação e de configuração. Um **servidor Web** abriga objetos Web, cada um endereçado por um URL.

- Quando um usuário requisita uma pagina Web, o browser envia ao servidor mensagens de requisição HTTP para os objetos da página, o servidor recebe as requisições e responde com mensagens de resposta HTTP que contêm os objetos.

- Até 1997, essencialmente todos os browser e servidores Web implementavam a versão HTTP/1.0, a partir de 1998 eles começaram a implementar a versão HTTP/1.1. O HTTP /1.1 é compatível com o HTTP /1.0, um servidor Web que executa a versão 1.1 pode se comunicar com um servidor que executa a 1.0.

- O HTTP usa o TCP como seu protocolo de transporte subjacente.
- O servidor HTTP não mantém nenhuma informação sobre clientes, por isso é denominado protocolo sem estado.

W W W

- Página WWW:
  - Consiste de “objetos”
  - Endereçada por um URL: *Universal Resource Locator*.
  - Quase todas as páginas WWW consistem de:
  - Página base HTML, e
  - Vários objetos referenciados.
- URL tem duas partes: nome do host, e nome de caminho.

- Agente de usuário para WWW = *browser*:
  - MS Internet Explorer.
  - Netscape Communicator.
- Servidor para WWW se chama “servidor WWW”:
  - Apache (*Open Software*).
  - *MS Internet Information Server* (IIS).

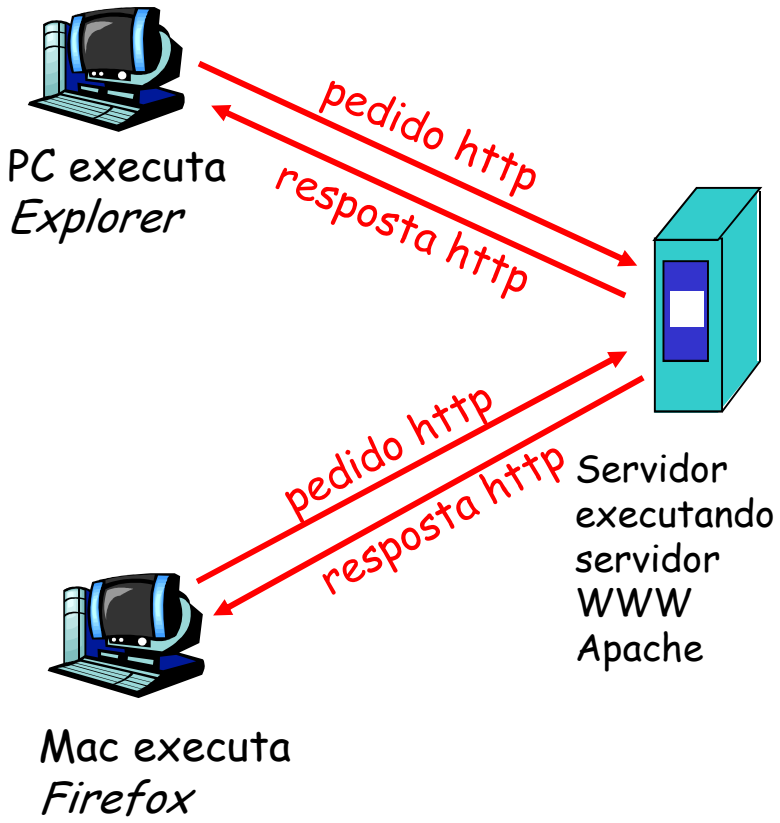
adriano.acmesecurity.org/themes/noprob/logo_cnpq.png

UEMG FRUTAL

WWW: o protocolo http

http: Hypertext Transfer Protocol

- Protocolo da **camada de aplicação** para WWW.
- Modelo cliente-servidor
  - *cliente:* *browser* que solicita, 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:

Usa serviço de transporte TCP:

1.) Cliente inicia conexão TCP (cria *socket*) ao servidor, na porta 80.

2.) Servidor aceita conexão TCP do cliente.

3.) Troca mensagens http (mensagens do protocolo da camada de aplicação) entre *browser* (cliente http) e servidor WWW (servidor http).

4.) **Encerra** conexão TCP.

http é "**sem estado**" ("***stateless***")

- Servidor não mantém informação sobre pedidos anteriores do cliente.

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

• O histórico passado (estado) tem que ser mantido.

• Demanda recursos maiores.

• São chamados de Protocolos "***Statefull***".

Exemplo de http (1)

Exemplo: Um usuário digita a URL www.unesp.br/index.html

(Ex: contém texto, referências a 10 imagens jpeg)

1a. Cliente http inicia conexão TCP ao servidor http (processo) em www.unesp.br.

Porta 80 é padrão para servidor http.

1b. servidor http no hospedeiro www.unesp.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, elabora *mensagem de resposta* contendo objeto solicitado (index.html), e envia mensagem via *socket*.

tempo

Continua...

Exemplo de http (2)

4. servidor http encerra conexão TCP .

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

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

tempo

E então, continua...

Conexões não-persistente e 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**.

Persistente:

- Default no HTTP/1.1
- **Na mesma conexão** TCP: servidor analisa pedido, responde, analisa novo pedido, etc... → não fecha a conexão.
- Cliente envia pedidos para todos objetos referenciados, assim que recebe o HTML base .
- **Menos RTTs, e menos partida lenta.**

A maioria de browsers usa conexões TCP paralelas.

UEMG FRUTAL

Conexão não-persistente (1)

- O cliente HTTP **inicia uma conexão TCP** com o servidor.
- O cliente **emite mensagem de requisição HTTP** usuário através do socket associado com a conexão do TCP que foi estabelecida.
- Servidor HTTP **recebe o *request* através do *socket*** associado com a conexão estabelecida, recupera o objeto `(index.html)` de seu armazenamento, encapsula o objeto em uma mensagem HTTP de resposta, e emite a mensagem de resposta ao cliente através do *socket*.
- O **servidor** diz ao cliente para **fechar a conexão TCP**.
- O cliente recebe a mensagem de resposta. **A conexão TCP termina**. A mensagem indica que o objeto encapsulado é um arquivo HTML. O cliente extrai o arquivo da mensagem de resposta, analisa, e encontra referências a 10 objetos do JPEG.
- **As primeiras quatro etapas são repetidas** então **para cada um dos objetos** JPEG referenciados.

UEMG FRUTAL

Conexão não persistente

- Cada conexão TCP é encerrada após o servidor enviar o objeto, ela não persiste para os outros objetos. Usuários podem configurar browsers modernos para controlar o grau de paralelismo(conexões paralelas podem ser abertas).
- O tempo de requisição que transcorre entre a requisição e o recebimento de um arquivo-base HTTP por um cliente é denominado tempo de viagem de ida e volta (RTT), já inclui atrasos.

Conexão não persistente

- Possui algumas desvantagens: uma nova conexão deve ser estabelecida e mantida para cada objeto solicitado. Para cada uma delas, devem ser alocados buffers TCP e conservadas variáveis TCP tanto no cliente quanto no servidor.

Conexão não-persistente (2)

- Quando o browser recebe uma página, mostra a página ao usuário. Dois browsers diferentes podem interpretar (isto é, mostrar ao usuário) um webpage de maneiras um diferentes.
  - **O HTTP não define como uma webpage é interpretada por um cliente.**
  - As especificações do HTTP (RFC1945 e RFC2616) definem somente o protocolo de comunicação entre o programa HTTP do cliente e o programa HTTP do servidor.
- Nas etapas das conexões não persistentes, **cada conexão do TCP é fechada depois que o servidor envia o objeto**: a conexão não persiste para outros objetos.
- **Cada conexão TCP transporta exatamente uma mensagem de pedido e uma mensagem de resposta**. No nosso exemplo, quando um usuário requisita uma página, 11 conexões TCP são geradas.

(discutir conexões paralelas)

UEMG FRUTAL

Round Trip Time (1)

- Quantidade de tempo gasta entre cliente **solicitar um arquivo HTML até que o arquivo esteja recebido**?
- Tempo ***round-trip-time*** (***RTT***), é o tempo gasto para um pacote viajar do cliente ao servidor e então voltar ao cliente. **É o tempo de ida-e-volta.**
- ***RTT*** inclui os atrasos de propagação, de enfileiramento e de processamento em routers e comutadores intermediários.
- Usuário clica num *hyperlink*. Isto faz com que o browser **inicie uma conexão do TCP entre o browser e o web server**.
- Envolve um "*three-way handshake*": o cliente emite uma mensagem TCP ao servidor, o servidor reconhece e responde com uma mensagem. Finalmente, o cliente confirma de volta ao servidor.

Round Trip Time (2)

- Mais um RTT decorre após as duas primeiras partes do *three-way handshake*.
  - Após ter terminado as primeiras duas partes do *handshake*, **o cliente emite a mensagem de requisição HTTP na conexão TCP**, e o TCP "agrega" a última confirmação (a terceira parte do *three-way handshake*) na mensagem do pedido. Uma vez a mensagem do pedido chega no usuário, o usuário emite o arquivo HTML na conexão do TCP.
- Esta interação HTTP "*request-response*" **gasta outro RTT**.
- Assim, **o tempo de resposta total é dois RTTs** mais o tempo da transmissão do arquivo HTML pelo servidor.

UEMG FRUTAL

Conexão persistente

- Em conexões persistentes, o servidor deixa a conexão TCP aberta após enviar a resposta, requisições e repostas subsequentes entre os mesmos cliente e servidor podem ser enviadas por meio da mesma conexão. Normalmente, o servidor HTTP fecha uma conexão quando ela não usada durante um certo tempo.

Conexão persistente (1)

- Cada objeto sofre dois RTTs:
  - um RTT para estabelecer a conexão TCP, e
  - um RTT para solicitar e receber um objeto.
- Cada objeto sofre do “partida lenta”(*slow start*) do TCP.
- **Com as conexões persistentes, o servidor deixa a conexão TCP aberta após ter emitido uma resposta.**
- Os pedidos e as respostas subseqüentes entre o mesmos cliente e servidor podem ser emitidos na mesma conexão.
- Uma *webpage* inteira (no exemplo, o arquivo HTML base e as 10 imagens) pode ser emitido sobre uma única conexão persistente do TCP.

Conexão persistente (2)

- Duas versões de conexões persistentes:
- **sem *pipelining*** (paralelismo) e **com *pipelining***.
  - **Sem *pipelining***: o cliente emite um pedido novo somente quando a resposta precedente foi recebida.
  - **Com *pipelining***: o cliente emite um pedido assim que encontrar uma referência.
    - **É o default para HTTP/1.1.**
    - **Somente um RTT para todos os objetos solicitados.**

UEMG FRUTAL

Mensagem http de **requisição** (1)

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

linha do pedido (comandos GET, POST, HEAD)

linhas do cabeçalho

```
GET /dir/page.html HTTP/1.1
User-agent: Mozilla/4.0
Accept: text/html, image/gif,image/jpeg
Accept-language:fr
```

(*carriage return* (CR), *linefeed* (LF) adicionais)

Carriage return, line feed indica fim de mensagem

- Escrita em ASCII, é constituída de cinco linhas, cada uma seguida de um ‘carriage return’ e ‘line feed’. Embora esta mensagem tenha 5 linhas, uma mensagem pode ter muito mais ou menos que isso. A primeira linha de uma mensagem é denominada linha de requisição, as subsequentes são denominadas linhas de cabeçalho.

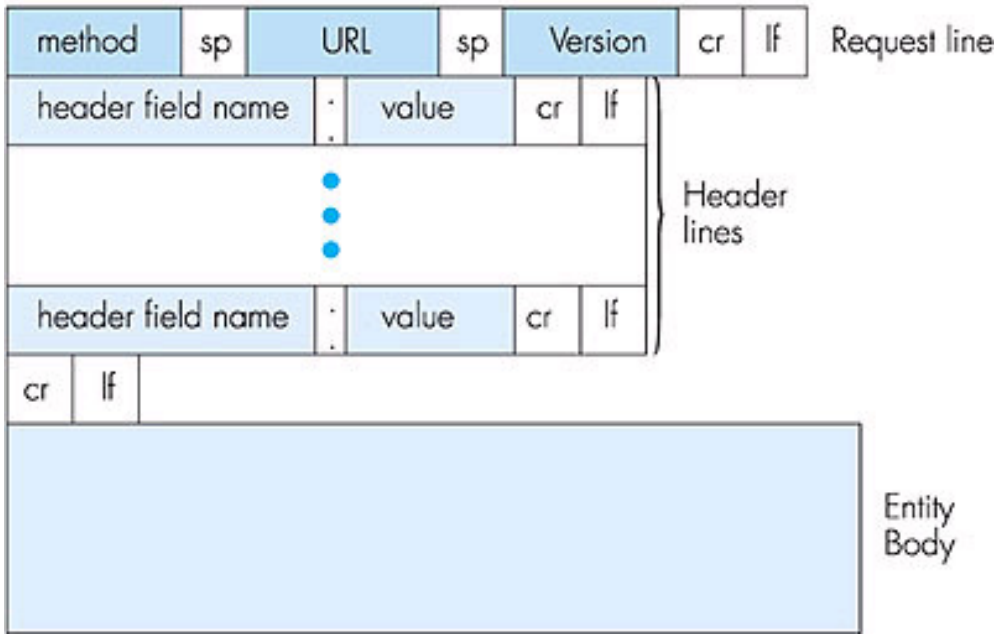
- O método GET é usado quando o browser requisita um objeto e este é identificado no campo do URL.
- Especifica o hospedeiro no qual o objeto se encontra.
- O browser está dizendo que não quer usar conexões persistentes.
- Especifica o agente de usuário.
- Mostra que o usuário prefere receber uma versão ,na língua especificada, do objeto se esse existir no servidor.

- Método POST: Utilizado quando o usuário preenche um formulário.

- Método HEAD: É semelhante ao GET, quando um servidor recebe uma requisição com ele, responde com uma mensagem HTTP, mas deixa de fora o objeto requisitado.

- HTTP/1.0 permite somente três tipos de métodos GET, POST e HEAD. Além desses três métodos, a especificação HTTP/1.1 permite vários métodos adicionais, entre eles PUT(permite que um usuário carregue um objeto para um caminho especifico) e DELETE(permite que o usuário elimine um objeto).

Mensagem http de **requisição** (2)



Formato geral de uma mensagem de **requisição**

Mensagem http de **resposta** (1)

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

linhas de cabeçalho

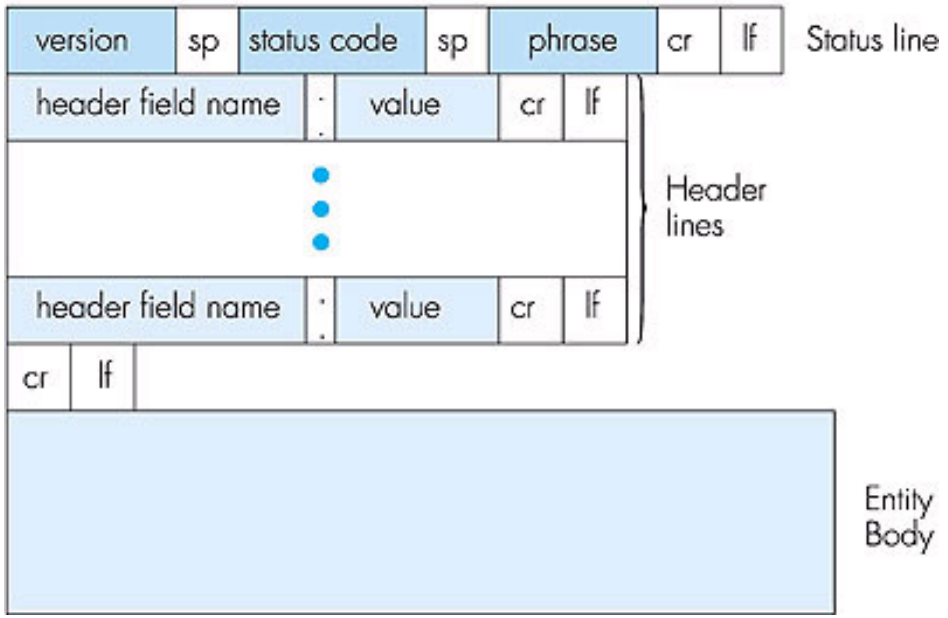
dados, p.ex., arquivo html solicitado

```
HTTP/1.1 200 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 ...
```

- Linha de estado.
- Conexão não persistente
- Indica a hora e a data em que a resposta foi criada e enviada pelo servidor.
- Mostra que a mensagem foi gerada por um servidor Web Apache.
- Indica a hora e data em que o objeto foi criado ou sofre a ultima modificação.
- Indica o número de bytes do objeto que está sendo enviado.
- Tipo do objeto.

Mensagem http de resposta (2)



Formato geral de uma mensagem de **RESPOSTA**

Códigos de *status* da resposta http

Aparecem na primeira linha da mensagem de resposta cliente-servidor. 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 aceita por este servidor.

UEMG FRUTAL

Vinicius Fernandes Caridá

vfcarida@gmail.com