

Redes de Computadores



Prof. Dr. Bruno Rodrigues



Protocolo HTTP



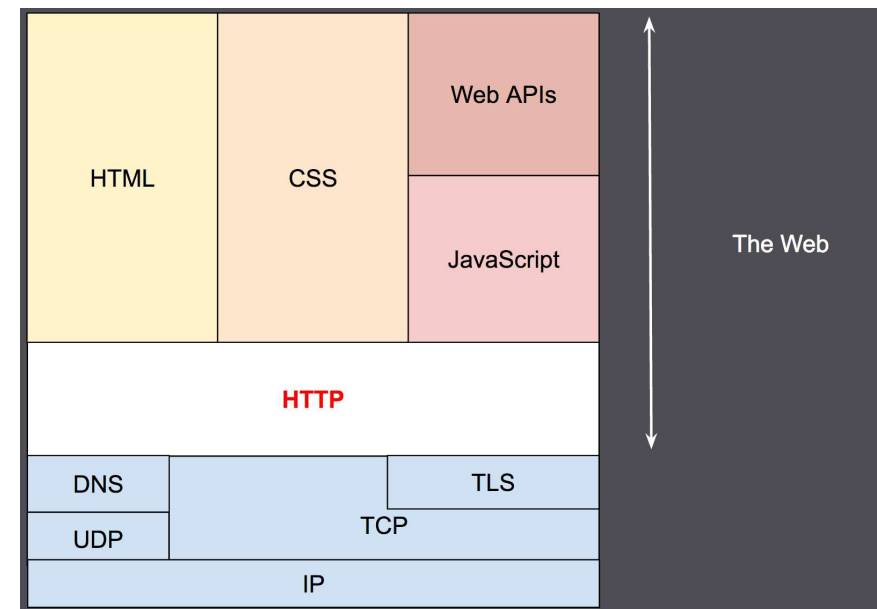
Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).

- ✓ **Página Web** consiste de **objetos**
- ✓ Objeto pode ser arquivo HTML, imagem JPEG, Java applet, arquivo de áudio, vídeos,...
- ✓ A página Web consiste de **arquivo-HTML base** que inclui vários objetos referenciados
- ✓ Cada objeto é endereçado por uma **URL (Uniform Resource Locator)**



Berners



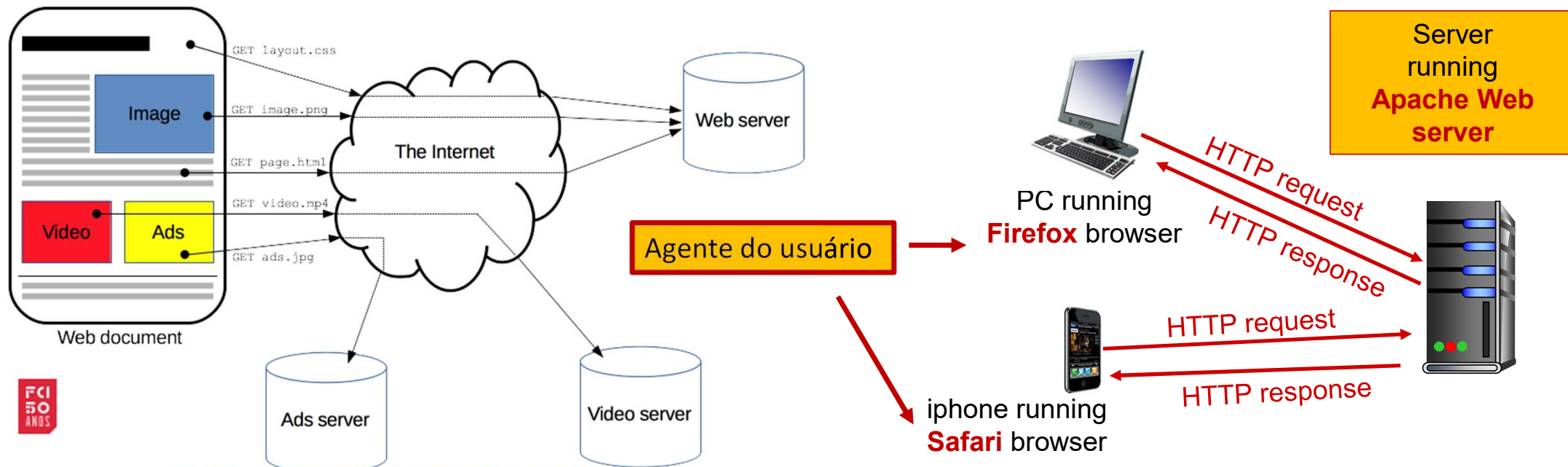
Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

Web e http

Neste modelo, Clientes e servidores se comunicam trocando mensagens individuais

Modelo cliente/servidor

- ✓ Define como clientes **requisitam** paginas aos servidores
- ✓ Define como o servidor **transfere** os dados para o cliente



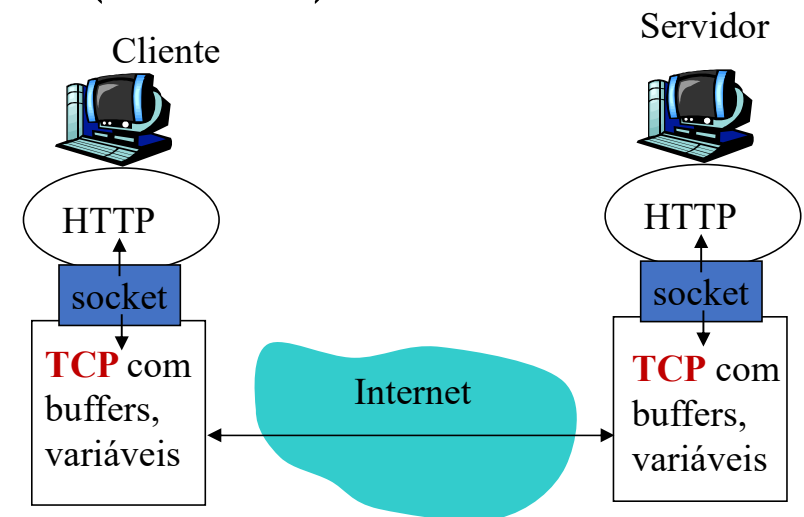
Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).

Usa TCP:

- ❖ Cliente inicia conexão TCP (cria socket) com servidor, **porta 80**
- ❖ servidor aceita conexão TCP do cliente
- ❖ mensagens HTTP (do protocolo da camada de aplicação) trocadas entre navegador (cliente HTTP) e servidor Web (servidor HTTP)
- ❖ conexão TCP fechada

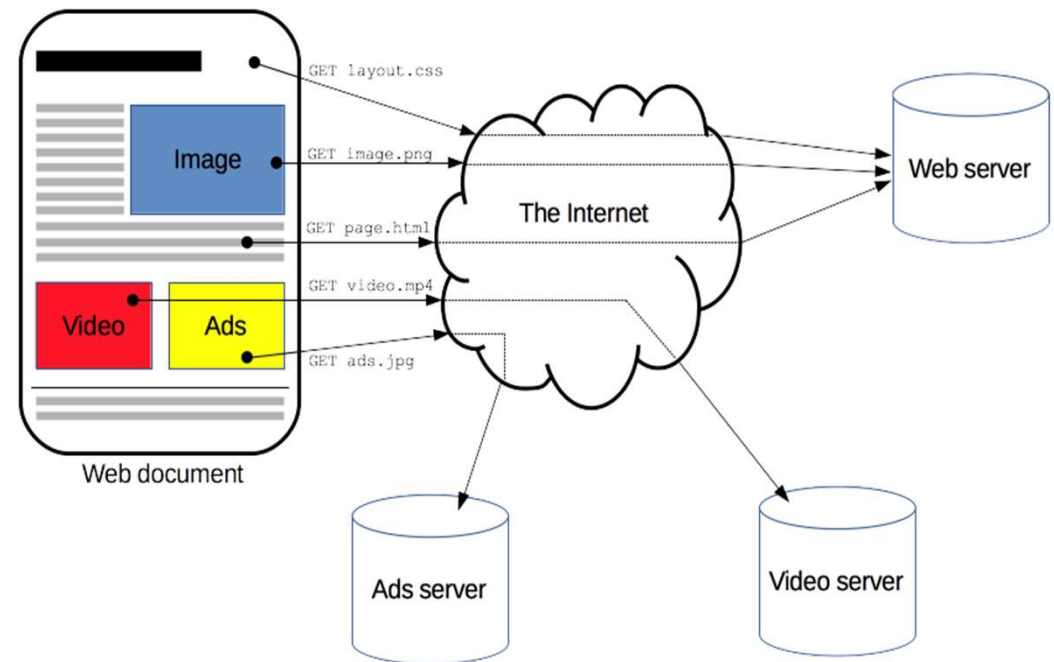
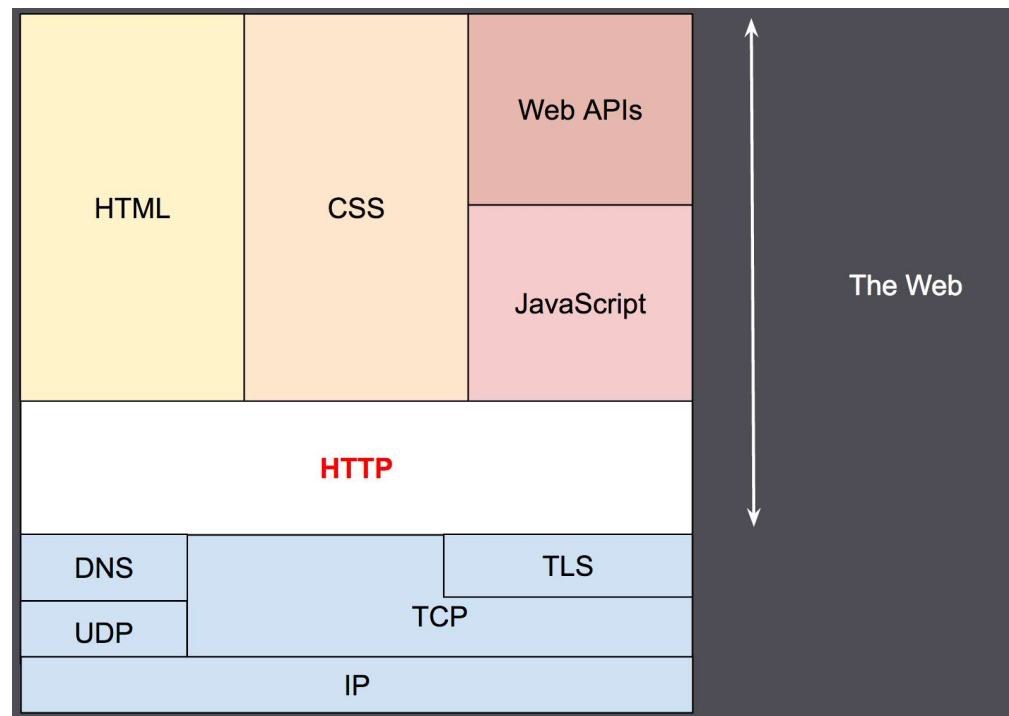


HTTP é “sem estado”

- ❖ Servidor não guarda informações sobre requisições passadas do cliente (complexo manter estado)

Web e http

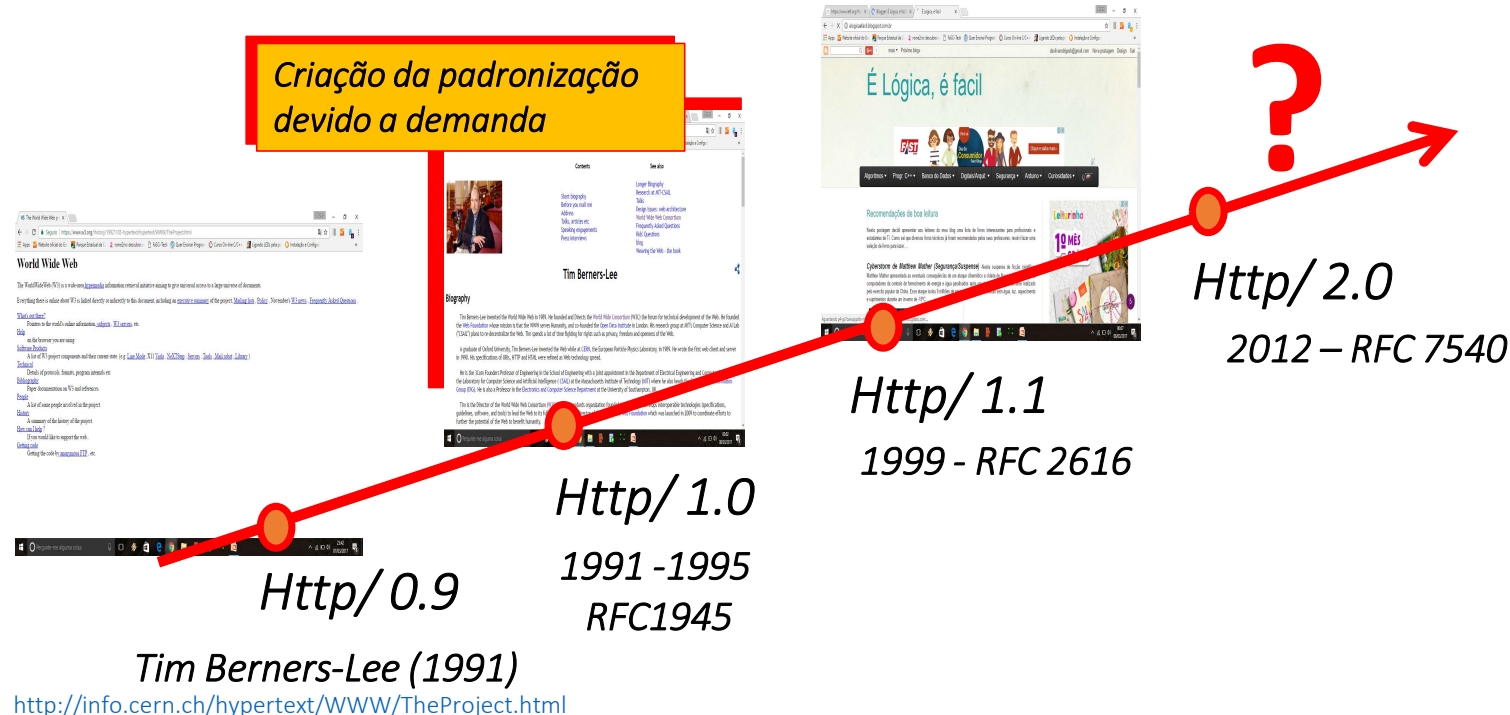
HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).



Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).



Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).

HTTP não persistente

- no máximo um objeto é enviado por uma conexão TCP.
- Utiliza **HTTP/1.0** [RFC 1945]

HTTP persistente

- Múltiplos objetos podem ser enviados por uma única conexão TCP entre cliente e servidor.
- Utiliza **HTTP/1.1** [RFC 2616]

Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).

HTTP não persistente

- no máximo um objeto é enviado por uma conexão TCP.
- Utiliza **HTTP/1.0** [RFC 1945]

HTTP persistente

- Múltiplos objetos podem ser enviados por uma única conexão TCP entre cliente e servidor.
- Utiliza **HTTP/1.1** [RFC 2616]

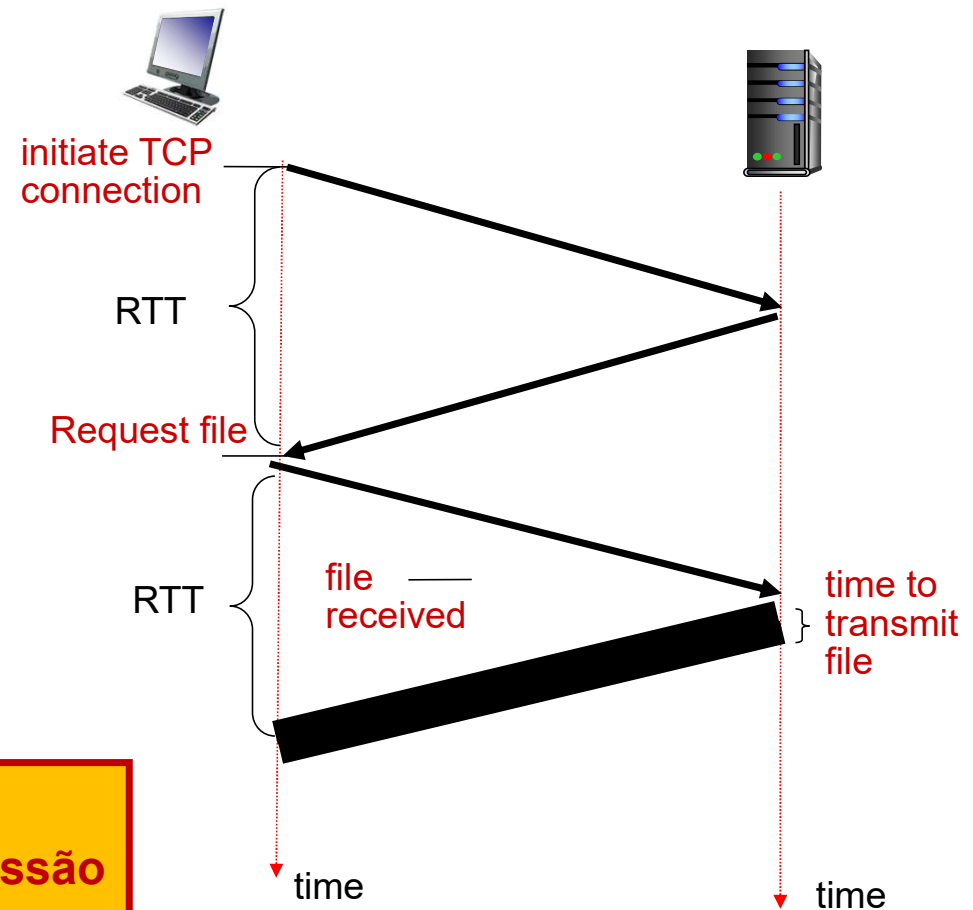
Web e http não persistente

Tempo de resposta de conexão não persistente

RTT (Round trip time): tempo para um pequeno pacote trafegar do cliente ao servidor e retornar.

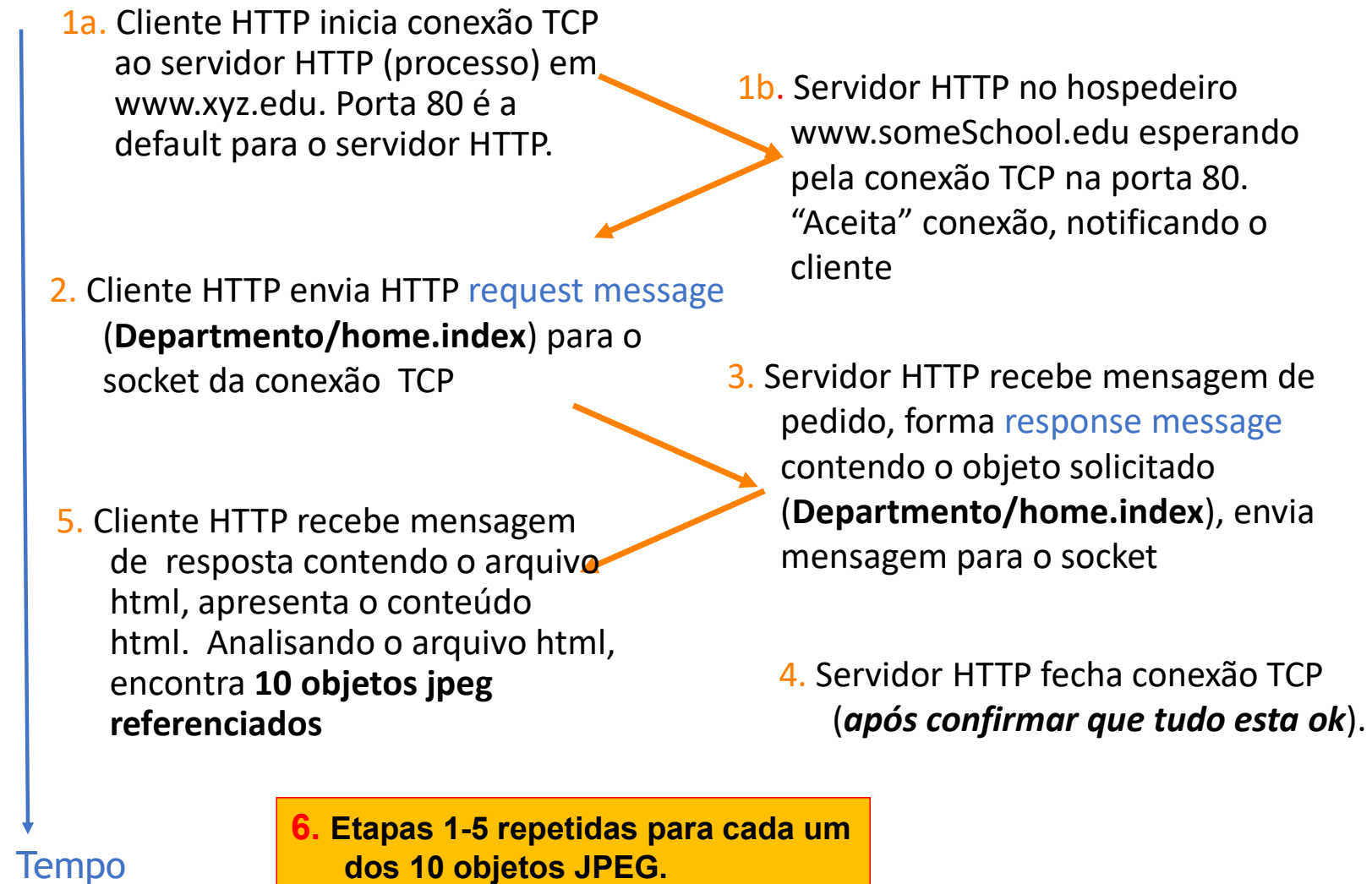
Tempo de resposta:

- um RTT para iniciar a conexão TCP
- um RTT para a requisição HTTP e primeiros bytes da resposta HTTP retornarem
- **Tempo de transmissão** de arquivo



$$T_{\text{total}} = 2\text{RTT} + \text{tempo de transmissão}$$

Web e http não persistente



Web e http não persistente

Tempo de resposta de conexão não persistente

Keith W. Ross

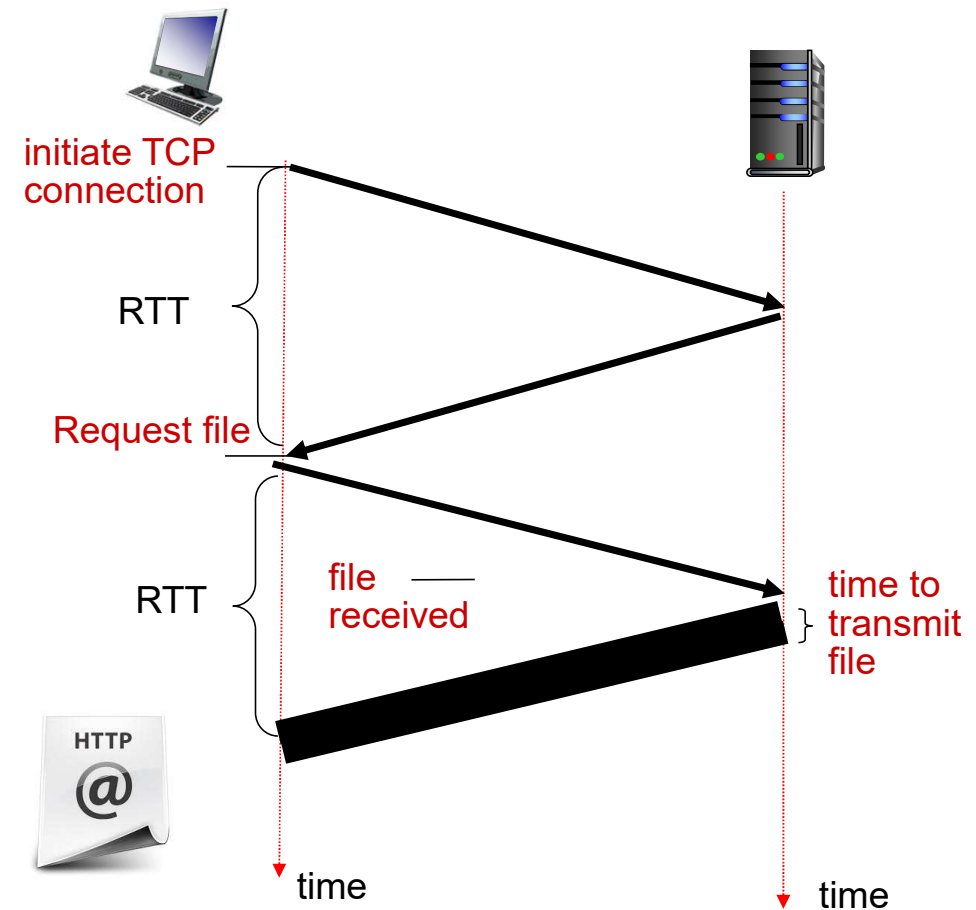


Dean of Engineering and Computer Science, NYU Shanghai
& Leonard J. Shustek Professor of Computer Science, CSE Dept, NYU

Keith Ross is the Dean of Engineering and Computer Science at NYU Shanghai and the Leonard J. Shustek Chair Professor in the Computer Science and Engineering Dept at NYU. Before joining NYU Poly in 2003, he was a professor at University of Pennsylvania (13 years) and a professor at Euronet Institute (5 years). He was the Department Head of the CSE Department at NYU from 2008 to 2013, and he joined NYU Shanghai in 2013. He received a B.S.E.E from Tufts University, a M.S.E.E. from Columbia University, and a Ph.D. in Computer and Control Engineering from The University of Michigan. He is an ACM Fellow and an IEEE Fellow.

His current research interests are in data-driven analysis of online social networks and privacy. He has also worked on peer-to-peer networking, Internet measurement, video streaming, applied probability and Markov decision processes. He is the recipient of several prestigious best paper awards and his work has been featured in the mainstream press, including New York Times, NPR, Bloomberg Television, Huffington Post, Fast Company, Ars Technica, and the New Scientist.

He is co-author (with James F. Kurose) of the popular textbook, Computer Networking: A Top-Down Approach Featuring the Internet, published by Pearson (first edition in 2000, seventh edition 2016). It is the most popular textbook on computer networking, both nationally and internationally, and has been translated into fourteen languages. Professor Ross is also the author of the research monograph, Multiservice Loss Models for Broadband Communication Networks, published by Springer in 1995.



Round trip time

Web e http não persistente

Tempo de resposta de conexão não persistente

Keith W. Ross



Dean of Engineering and Computer Science, NYU Shanghai
& Leonard J. Shustek Professor of Computer Science, CSE Dept, NYU

Keith Ross is the Dean of Engineering and Computer Science at NYU Shanghai and the Leonard J. Shustek Chair Professor in the Computer Science and Engineering Dept at NYU. Before joining NYU Poly in 2003, he was a professor at University of Pennsylvania (13 years) and a professor at Eurecom Institute (5 years). He was the Department Head of the CSE Department at NYU from 2008 to 2013, and he joined NYU Shanghai in 2013. He received a B.S.E.E from Tufts University, a M.S.E.E. from Columbia University, and a Ph.D. in Computer and Control Engineering from The University of Michigan. He is an ACM Fellow and an IEEE Fellow.

His current research interests are in data-driven analysis of online social networks and privacy. He has also worked on peer-to-peer networking, Internet measurement, video streaming, applied probability and Markov decision processes. He is the recipient of several prestigious best paper awards and his work has been featured in the mainstream press, including New York Times, NPR, Bloomberg Television, Huffington Post, Fast Company, Ars Technia, and the New Scientist.

He is co-author (with James F. Kurose) of the popular textbook, Computer Networking: A Top-Down Approach Featuring the Internet, published by Pearson (first edition in 2000, seventh edition 2016). It is the most popular textbook on computer networking, both nationally and internationally, and has been translated into fourteen languages. Professor Ross is also the author of the research monograph, Multiservice Loss Models for Broadband Communication Networks, published by Springer in 1995.

NAVIGATION

HOME

HONORS

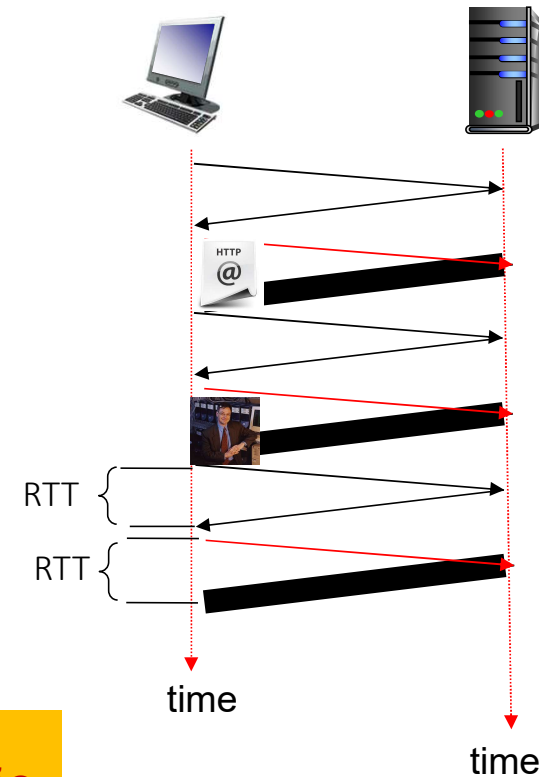
PUBLICATIONS

RESEARCH TEAM

SELECTED PRESS

RECENT COURSES

CONTACT



$$T_{\text{total}} = X \cdot (2RTT) + \sum \text{tempo de transmissão}$$

Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).

HTTP não persistente

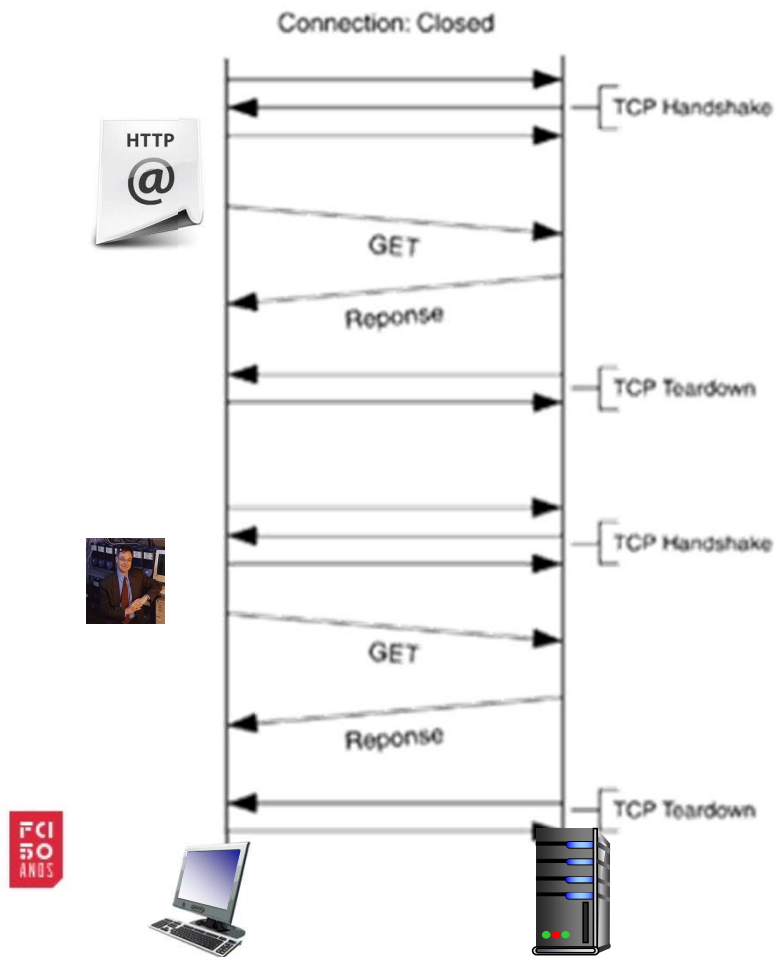
- no máximo um objeto é enviado por uma conexão TCP.
- Utiliza **HTTP/1.0 [RFC 1945]**

HTTP persistente

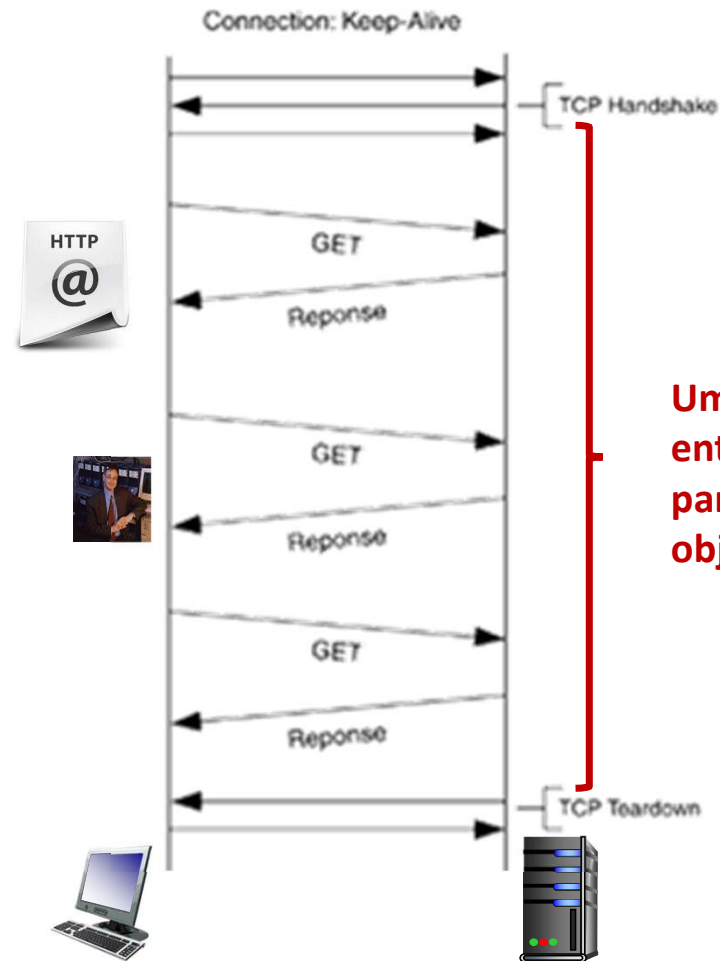
- Múltiplos objetos podem ser enviados por uma única conexão TCP entre cliente e servidor.
- Utiliza **HTTP/1.1 [RFC 2616]**

Web e http

HTTP não persistente (HTTP 1.0)



HTTP persistente (HTTP 1.1)



Uma única conexão TCP entre cliente e servidor para envio de múltiplos objetos

Web e http

Resumindo:

HTTP não persistente:

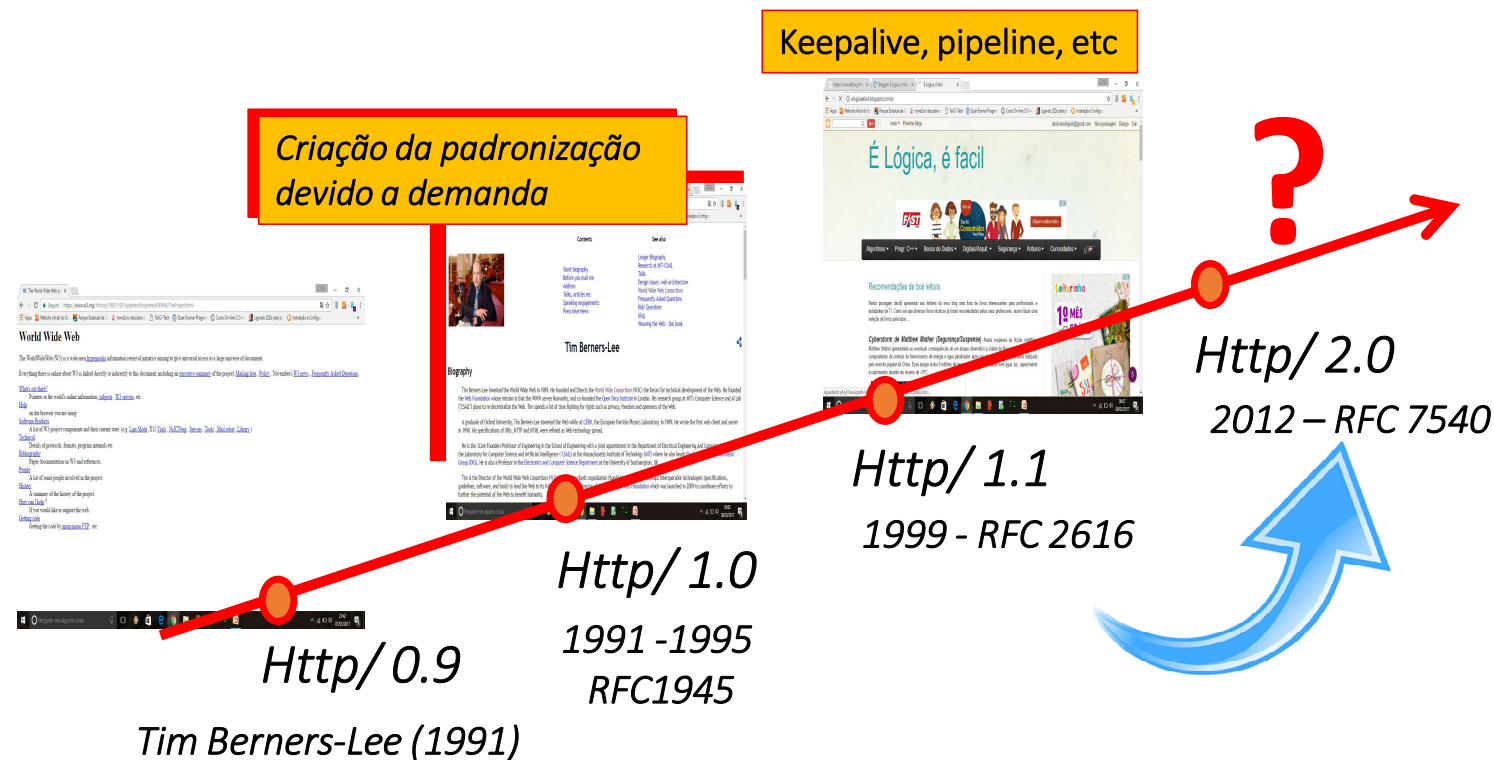
- Requer 2 RTTs por objeto
- Overhead do SO para *cada* conexão TCP
- Navegadores geralmente abrem conexões TCP paralelas para buscar objetos referenciados

HTTP persistente:

- Servidor deixa a conexão aberta depois de enviar a resposta
- Mensagens HTTP seguintes entre cliente/servidor enviadas pela conexão aberta
- Cliente envia requisições assim que encontra um objeto referenciado
- No mínimo um RTT para todos os objetos referenciados

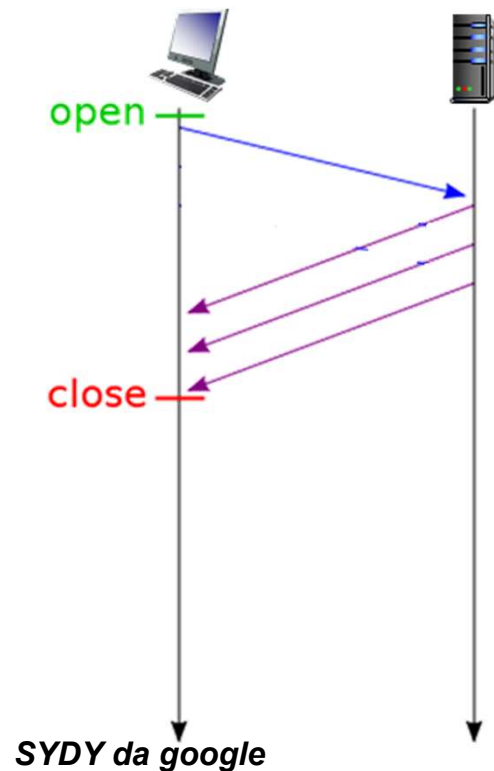
Web e http

HTTP (HyperText Transfer Protocol) - É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet).



Web e http

O **HTTP/2** permite um uso mais eficiente da rede Recursos e uma percepção reduzida de latência através da introdução compressão do cabeçalho e permitindo múltiplas comunicações usando a mesma conexão. Ele também introduz **Server Push** de representações de servidores para clientes



Mas isso é uma outra história.....

Formato das mensagens HTTP



Formato das msgs http

As especificações do HTTP (RFC 2616) definem o formato das mensagens (**Requisição e Resposta**)

Mensagem de requisição:

linha de requisição
(comandos GET,
POST, HEAD)

linhas de
cabeçalho

carriage return,
line feed
indica final
da mensagem

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

carriage return
line feed

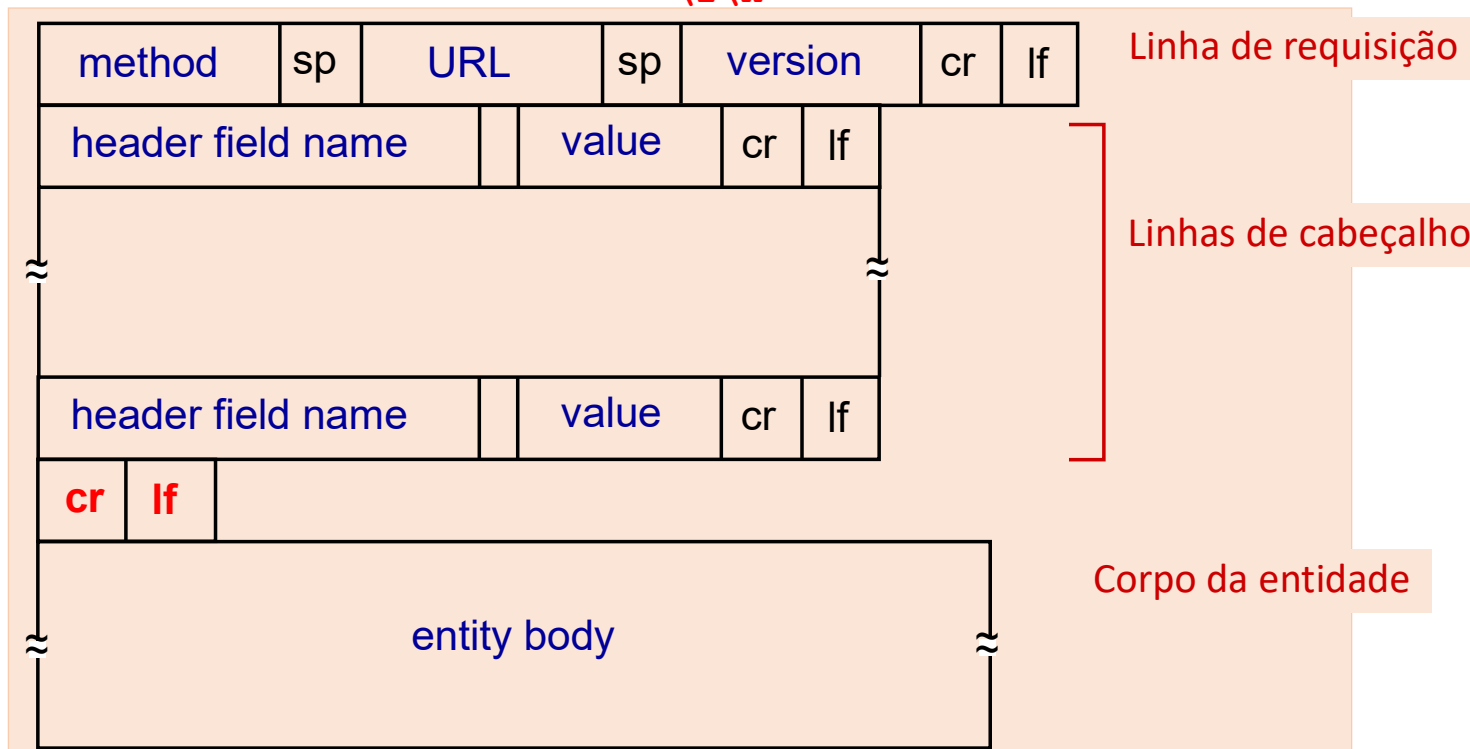
Formato das msgs http

Mensagem de requisição:

carriage return
line feed

**GET,
POST,
HEAD**

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
\r\n
```



Formato das msgs http

As especificações do HTTP (RFC 2616) definem o formato das mensagens (**Requisição e Resposta**)

Mensagem de requisição:

Sites são otimizados para visualização com navegadores web móveis.

linha de requisição
(comandos GET,
POST, HEAD)

linhas de
cabeçalho

carriage return,
line feed
indica final
da mensagem

```
GET /index.html HTTP/1.1\r\n
Host: www-newrelic.com\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9\r\n
Accept-Language: en-US,en;q=0.5\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
\r\n
```

Formato das msgs http

Mensagem de requisição: Métodos

HTTP/1.0

- GET
- POST
- HEAD
 - pede ao servidor para deixar objeto requisitado fora da resposta

HTTP/1.1

- GET, POST, HEAD
- PUT
- DELETE
 - exclui arquivo especificado no campo de URL

Formato das msgs http

Mensagem de requisição: Métodos

- ✓ **GET:** É o método mais comum: **solicita algum recurso** (qualquer dado que estiver identificado pelo URI) **por meio do protocolo HTTP**. O método GET é reconhecido por todos os servidores.

Formato das msgs http

Mensagem de requisição: Métodos

- ✓ **GET:** É o método mais comum: solicita algum recurso (qualquer dado que estiver identificado pelo URI) por meio do protocolo HTTP. O método GET é reconhecido por todos os servidores.
- ✓ **HEAD:** É o mesmo que GET, mas sem que o corpo da mensagem seja retornado. É usado para obter **meta-informações** por meio do cabeçalho da resposta, sem ter que recuperar todo o conteúdo.

Formato das msgs http

Mensagem de requisição: Métodos

- ✓ **GET:** É o método mais comum: solicita algum recurso (qualquer dado que estiver identificado pelo URI) por meio do protocolo HTTP. O método GET é reconhecido por todos os servidores.
- ✓ **HEAD:** É o mesmo que GET, mas sem que o corpo da mensagem seja retornado. É usado para obter **meta-informações** por meio do cabeçalho da resposta, sem ter que recuperar todo o conteúdo.
- ✓ **POST:** **Envia dados para serem processados** (por exemplo, dados de um formulário HTML) para o recurso especificado. Os dados são incluídos no corpo do comando

Formato das msgs http

Mensagem de resposta http

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

linhas de
cabeçalho

dados, p. e.,
arquivo HTML
requisitado

```
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 .....
Content-Length: 6821
Content-Type: text/html

dados dados dados dados dados ...
```

Formato das msgs http

Mensagem de resposta http

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

linhas de
cabeçalho

dados, p. e.,
arquivo HTML
requisitado

```
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 .....
Content-Length: 6821
Content-Type: text/html

dados dados dados dados dados ...
```

Formato das msgs http

Mensagem de resposta http

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Euroelec', displaying the website 'www.euroelec-smartenergy.com/pt/'. The page features the 'Euroelec SmartEnergy' logo and a navigation menu with links: Home, Expertise, Solutions, Réalisations, Actualités, and Contact. A message at the top indicates 'Erreur 404 - page introuvable' (404 Error - page not found). Below this, a search section states 'Aucune page ne correspond à votre recherche' (No page corresponds to your search) and provides a search bar with the text 'Rechercher'. To the right, a 'PAGES' sidebar lists various site sections: Actualités, Bâtiments intelligents, Contact, CSSI, Équipe, Expertise, Green Data Center, Home, Photovoltaïque, Réalisations, Réseau intelligent et communicant, Solutions, and Ville durable. At the bottom left, there is a small red logo for 'FCI 50 ANOS'.

Wiscite x Desafio x O futuro x PTC lan x Alexa x Conect x Alice In x Whats x Smart x cidade x Euroelec x Bruno

www.euroelec-smartenergy.com/pt/

Apps Website oficial do Go Parque Estadual da C rome2rio: descubra c FaSci-Tech Quer Ensinar Program PKP Tutorial sobre Redes Viagem à Europa: com Curso On-line C/C++

Euroelec
SmartEnergy

Home Expertise Solutions Réalisations Actualités Contact

Erreur 404 - page introuvable

Vous êtes ici : Accueil / Erreur 404

Aucune page ne correspond à votre recherche

Désolé, l'article que vous recherchez est introuvable. Veuillez utiliser la fonction Rechercher ci-dessous.

Rechercher

Suggestions :

- Vérifiez l'orthographe des termes de recherche.
- Utilisez des mots clés plus génériques.
- Essayez d'utiliser plusieurs termes de recherche.

Envie de parcourir plutôt nos actualités ?

PAGES

- Actualités
- Bâtiments intelligents
- Contact
- CSSI
- Équipe
- Expertise
- Green Data Center
- Home
- Photovoltaïque
- Réalisations
- Réseau intelligent et communicant
- Solutions
- Ville durable

FCI 50 ANOS

Formato das msgs http

Mensagem de resposta http

Códigos na primeira linha da mensagem de resposta :

- ❖ **1xx**: Informational (**Informação**) – utilizada para enviar informações para o cliente de que sua requisição foi recebida e está sendo processada;

Formato das msgs http

Mensagem de resposta http

Códigos na primeira linha da mensagem de resposta :

❖ **2xx**: Success (**Sucesso**) – Indica que a requisição do cliente foi bem sucedida;

200 - OK

✓ Requisição bem-sucedida

206 Conteúdo parcial

✓ O servidor cumpriu a solicitação GET parcial para o recurso, conforme condição de cabeçalho.

Formato das msgs http

Mensagem de resposta http

Códigos na primeira linha da mensagem de resposta :

- ❖ **3xx**: Redirection (**Redirecionamento**) – informa a ação adicional que deve ser tomada para completar a requisição;

301 Moved Permanently

- ✓ Objeto requisitado movido, novo local especificado mais adiante na mensagem (Location:)

304 Não modificado

- ✓ Se o cliente tiver executado uma solicitação **GET condicional** e o acesso for permitido, mas o documento não foi modificado, o servidor

307 Redirecionamento temporário

- ✓ O pedido deve ser redirecionado para outra URL

Formato das msgs http

Mensagem de resposta http

Códigos na primeira linha da mensagem de resposta :

❖ **4xx**: Client Error (**Erro no cliente**) – avisa que o cliente fez uma requisição que não pode ser atendida;

400 Bad Request

- ✓ Mensagem de requisição não entendida pelo servidor (syntaxe)

401 Não autorizado

- ✓ especificamente para o uso quando a autenticação é possível, mas não conseguiu ou ainda não foram fornecidos

404 Not Found

- ✓ Documento requisitado não localizado neste servidor

Formato das msgs http

Mensagem de resposta http

Códigos na primeira linha da mensagem de resposta :

Inspecionar elementos:

<http://www.nyu.edu/projects/keithwross/>

<http://up.mackenzie.br/graduacao/sao-paulo/ciencia-da-computacao/corpo-docente/>

Cookies



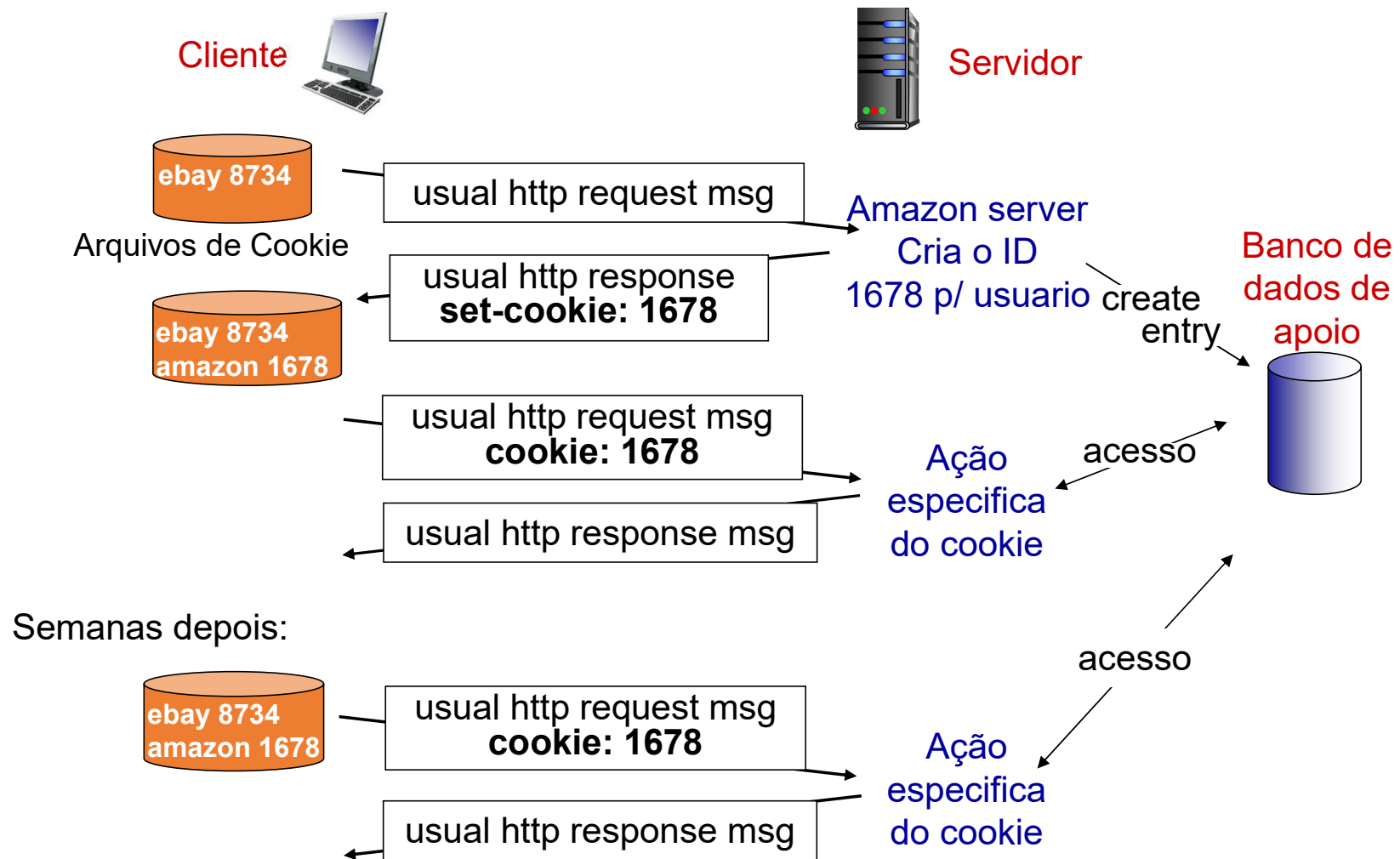
Estado usuário-servidor: cookies

Um **cookie** é um pedaço de texto que um servidor Web pode armazenar no disco rígido do usuário. São utilizados pelos sites principalmente para identificar e armazenar informações sobre os visitantes.

Quatro componentes:

- 1) Linha de cabeçalho do cookie na mensagem HTTP response
- 2) Linha de cabeçalho de cookie na mensagem HTTP request
- 3) Arquivo de cookie mantido no hospedeiro do usuário e manipulado pelo browser do usuário
- 4) Banco de dados **backend** no Web site

Estado usuário-servidor: cookies



Estado usuário-servidor: cookies

O que os cookies podem trazer:

- ✓ Autorização
- ✓ Cartões de compra
- ✓ Recomendações
- ✓ Estado de sessão do usuário (Web e-mail)

Cookies e privacidade:

- ☐ Cookies permitem que os sites descubram muito sobre você
- ☐ Você pode fornecer nome e e-mail aos sites

Como manter o “estado”:

- ✓ extremidades do protocolo: mantêm estado no emissor/receptor por múltiplas transações
- ✓ cookies: mensagens HTTP transportam estado

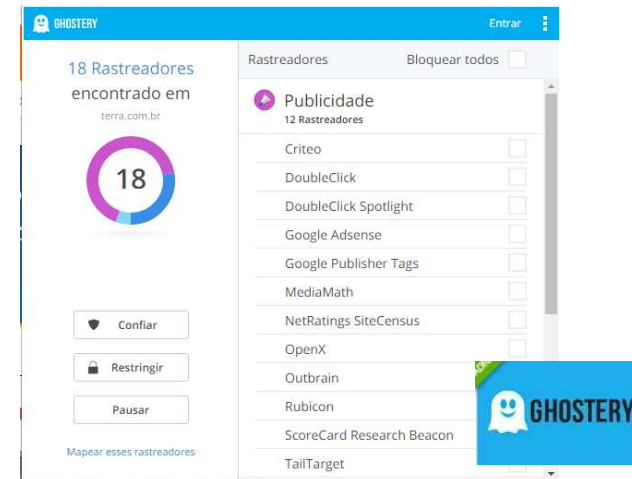
Estado usuário-servidor: cookies

Mensagem de requisição:

```
> Frame 4072: 780 bytes on wire (6240 bits), 780 bytes captured (6240 bits) on interface 0
> Ethernet II, Src: HonHaiPr_45:e7:95 (68:14:01:45:e7:95), Dst: Tp-LinkT_73:f8:1d (18:a6:f7:73:f8:1d)
> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 172.217.29.233
> Transmission Control Protocol, Src Port: 16102, Dst Port: 80, Seq: 1, Ack: 1, Len: 726
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.blogger.com\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate, sdch\r\n
    Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4\r\n
  > [truncated]Cookie: NID=88=cBYSQMBs19ucc4_PbQKJecT64b4bU1Qce5voj5cZiedJTqJ7aAvq6SVmTy58mZlNI1bF__40oBbf8TOyDrNNnCG1haJFN4gJy8AhzT3J\r\n
    \r\n
    [Full request URI: http://www.blogger.com/]
    [HTTP request 1/1]
    [Response in frame: 4082]
```

Cookies e privacidade!!!

<https://www.google.com/policies/technologies/types/>





Caches Web

Caches Web (Servidor proxy)

HTTP é tipicamente usado para sistemas de informação distribuídos, onde o desempenho pode ser melhorado pelo uso de caches de resposta.

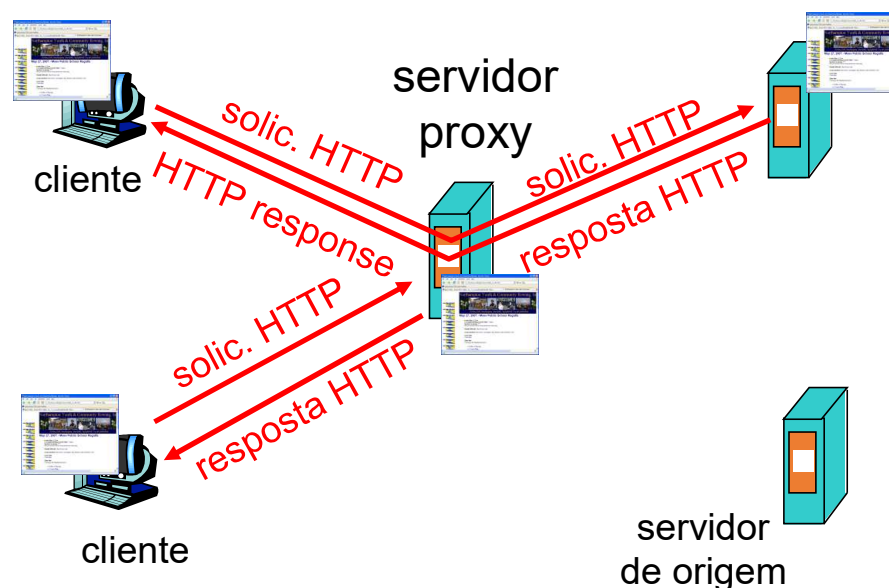
Objetivo: satisfazer a requisição do cliente sem envolver servidor de origem

Caches Web (Servidor proxy)

HTTP é tipicamente usado para sistemas de informação distribuídos, onde o desempenho pode ser melhorado pelo uso de caches de resposta.

Objetivo: satisfazer a requisição do cliente sem envolver servidor de origem

- Usuário configura o browser: acesso Web é feito por meio de um proxy
- Cliente envia todos os pedidos HTTP para o Web cache
 - Se o objeto existe no Web cache: Web cache retorna o objeto
 - Ou o Web cache solicita objeto do servidor original e então envia o objeto ao cliente

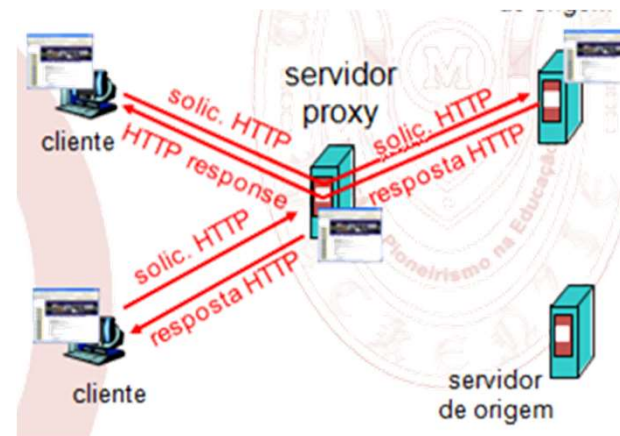


Caches Web (Servidor proxy)

HTTP é tipicamente usado para sistemas de informação distribuídos, onde o desempenho pode ser melhorado pelo uso de caches de resposta.

Objetivo: satisfazer a requisição do cliente sem envolver servidor de origem

- Usuário configura o browser: acesso Web é feito por meio de um proxy;
- Cliente envia todos os pedidos HTTP para o Web cache:
 - Se o objeto existe no Web cache: Web cache retorna o objeto
 - Ou o Web cache solicita objeto do servidor original e então envia o objeto ao cliente
- Cache atua como cliente e servidor;
- Normalmente, cache é instalado por ISP (da universidade, empresa, residencial).



Caches Web (Servidor proxy)

Por que caching Web?

- I. Reduz tempo de resposta à requisição do cliente;

Caches Web (Servidor proxy)

Por que caching Web?

Atrasos e rotas “reais”

traceroute: roteadores, atrasos de ida e volta no caminho da origem até o destino source-dest path
também: pingplotter, vários programas windows (tracert)

nºrot	nome rot	IP rot	RTT-pac1	RTT-pac2	RTT-pac3
1	cs-gw	(128.119.240.254)	1 ms	1 ms	2 ms
2	border1-rt-fa5-1-0.gw.umass.edu	(128.119.3.145)	1 ms	1 ms	2 ms
3	cht-vbns.gw.umass.edu	(128.119.3.130)	6 ms	5 ms	5 ms
4	jn1-at1-0-0-19.wor.vbns.net	(204.147.132.129)	16 ms	11 ms	13 ms
5	jn1-so7-0-0-0.wae.vbns.net	(204.147.136.136)	21 ms	18 ms	18 ms
6	abilene-vbns.abilene.ucaid.edu	(198.32.11.9)	22 ms	18 ms	22 ms
7	nycm-wash.abilene.ucaid.edu	(198.32.8.46)	22 ms	22 ms	22 ms
8	62.40.103.253	(62.40.103.253)	104 ms	109 ms	106 ms
9	de2-1.de1.de.geant.net	(62.40.96.129)	109 ms	102 ms	104 ms
10	de.fr1.fr.geant.net	(62.40.96.50)	113 ms	121 ms	114 ms
11	renater-gw.fr1.fr.geant.net	(62.40.103.54)	112 ms	114 ms	112 ms
12	nio-n2.cssi.renater.fr	(193.51.206.13)	111 ms	114 ms	116 ms
13	nice.cssi.renater.fr	(195.220.98.102)	123 ms	125 ms	124 ms
14	r3t2-nice.cssi.renater.fr	(195.220.98.110)	126 ms	126 ms	124 ms
15	eurecom-valbonne.r3t2.ft.net	(193.48.50.54)	135 ms	128 ms	133 ms
16	194.214.211.25	(194.214.211.25)	126 ms	128 ms	126 ms
17	***				
18	***				
19	fantasia.eurecom.fr	(193.55.113.142)	132 ms	128 ms	136 ms

link transoceânico

* sem resposta (perda de probe, roteador não responde)

Caches Web (Servidor proxy)

Por que caching Web?

Atrasos e rotas "reais"

traceroute: roteadores, atrasos de ida e volta no caminho da origem até o destino source-dest path
também: pingplotter, vários programas windows (tracert)

nº rot	nome rot	IP rot	RTT-pac1	RTT-pac2	RTT-pac3
1	cs-gw	(128.119.240.254)	1 ms	1 ms	2 ms
2	border1-rt-fa5-1-0.gw.umass.edu	(128.119.3.145)	1 ms	1 ms	2 ms
3	cht-vbns.gw.umass.edu	(128.119.3.130)	6 ms	5 ms	5 ms
4	jn1-at1-0-0-19.wor.vbns.net	(204.147.132.129)	16 ms	11 ms	13 ms
5	jn1-so7-0-0-0.wae.vbns.net	(204.147.136.136)	21 ms	18 ms	18 ms
6	abilene-vbns.abilene.ucaid.edu	(198.32.11.9)	22 ms	18 ms	22 ms
7	nycm-wash.abilene.ucaid.edu	(198.32.8.46)	22 ms	22 ms	22 ms
8	62.40.103.253	(62.40.103.253)	104 ms	109 ms	106 ms
9	de2-1.de1.de.geant.net	(62.40.96.129)	109 ms	102 ms	104 ms
10	de.fr1.fr.geant.net	(62.40.96.50)	113 ms	121 ms	114 ms
11	renater-gw.fr1.fr.geant.net	(62.40.103.54)	112 ms	114 ms	112 ms
12	nio-n2.cssi.renater.fr	(193.51.206.13)	111 ms	114 ms	116 ms
13	nice.cssi.renater.fr	(195.220.98.102)	123 ms	125 ms	124 ms
14	r3t2-nice.cssi.renater.fr	(195.220.98.110)	126 ms	126 ms	124 ms
15	eurecom-valbonne.r3t2.ft.net	(193.48.50.54)	135 ms	128 ms	133 ms
16	194.214.211.25	(194.214.211.25)	126 ms	128 ms	126 ms
17	***				
18	***				
19	fantasia.eurecom.fr	(193.55.113.142)	132 ms	128 ms	136 ms

link
transoceânico

* sem resposta (perda de probe, roteador não responde)

Um Web Cache neste ponto reduziria o tempo de comunicação

Caches Web (Servidor proxy)

Por que caching Web?

- I. Reduz tempo de resposta à requisição do cliente;
- II. Reduz tráfego no enlace de acesso de uma instituição;

Caches Web (Servidor proxy)

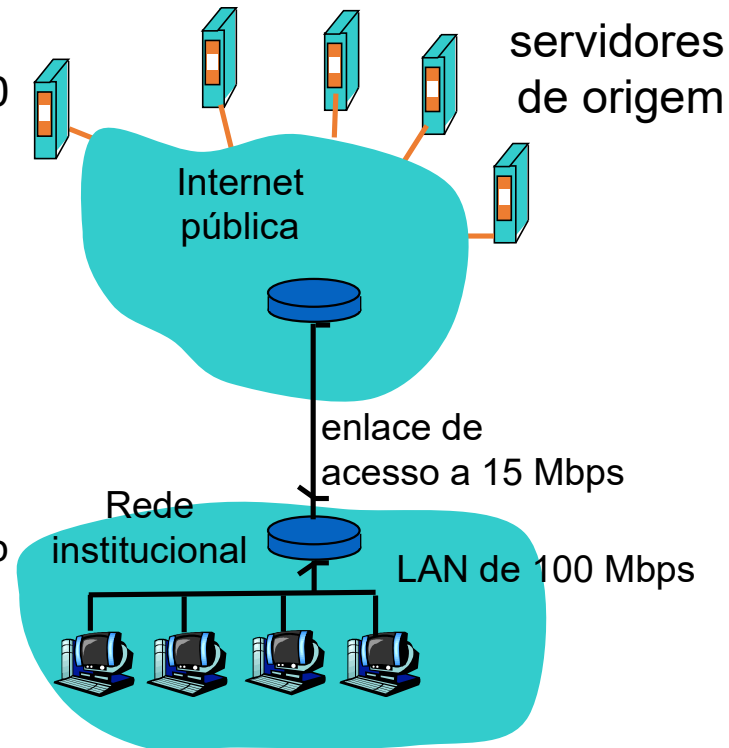
Exemplo de caching

Suposições:

- Tamanho médio do objeto = 1.000.000 bits
- Taxa de requisição média = 15 req./s
- Atraso RTT = 2 s

Conseqüências

- Utilização na LAN = 15%
- Utilização no enlace de acesso = **100%**
- Atraso total = atraso da Internet + atraso do acesso + atraso da LAN



Atraso = 2 s + x minutos + y milissegundos

Caches Web (Servidor proxy)

Exemplo de caching

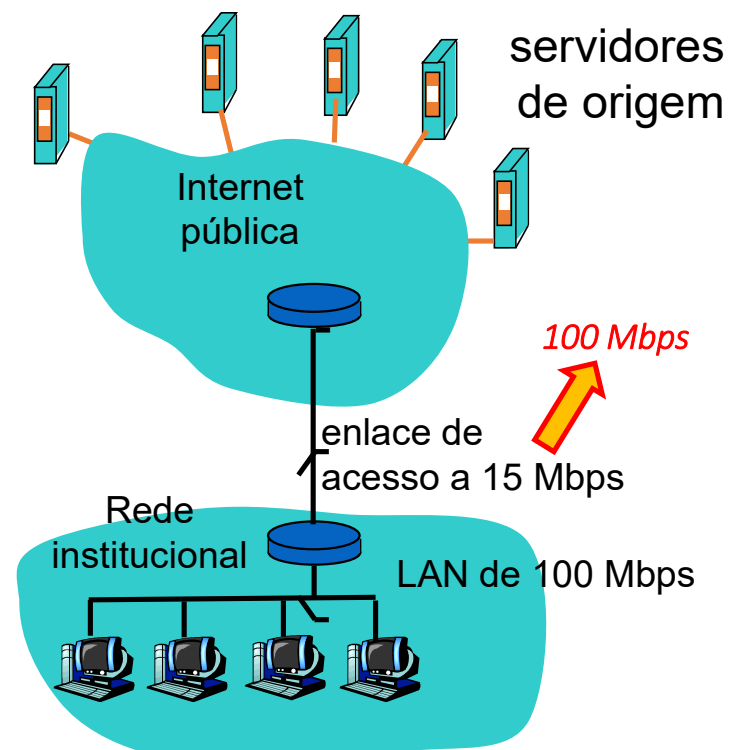
Solução possível :

- aumentar largura de banda do enlace de acesso para, digamos, 100 Mbps

Conseqüência:

- Utilização na LAN = 15%
- Utilização no enlace de acesso = 15%
- Atraso total = atraso da Internet + atraso do acesso + atraso da LAN = 2 s + x ms + y ms

- *normalmente, uma atualização dispendiosa*



Caches Web (Servidor proxy)

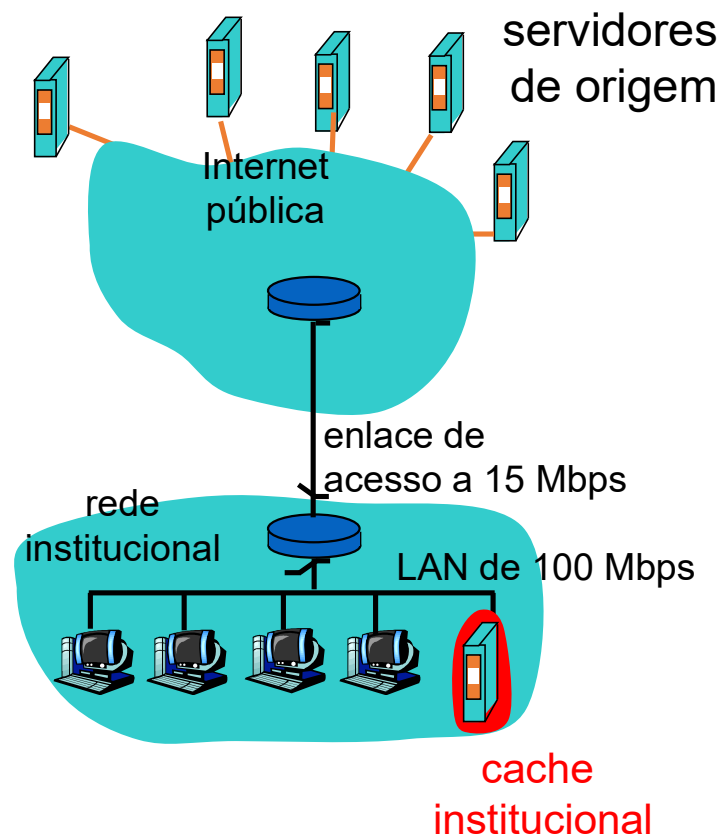
Possível solução: instalar cache:

- Suponha que índice de acerto é 0,4

Consequência:

- 40% de requisições serão satisfeitas imediatamente
- 60% de requisições satisfeitas pelo servidor de origem
- Utilização do enlace de acesso reduzida para 60%, resultando em atrasos insignificantes (digamos, 10 ms)
- Atraso médio total = atraso da Internet + atraso de acesso + atraso da LAN

Exemplo de caching



Caches Web (Servidor proxy)

- **OBJETIVO:** não enviar objeto se o cache tiver versão atualizada
- cache: especifica data da cópia em cache na requisição HTTP

If-modified-since: <data>

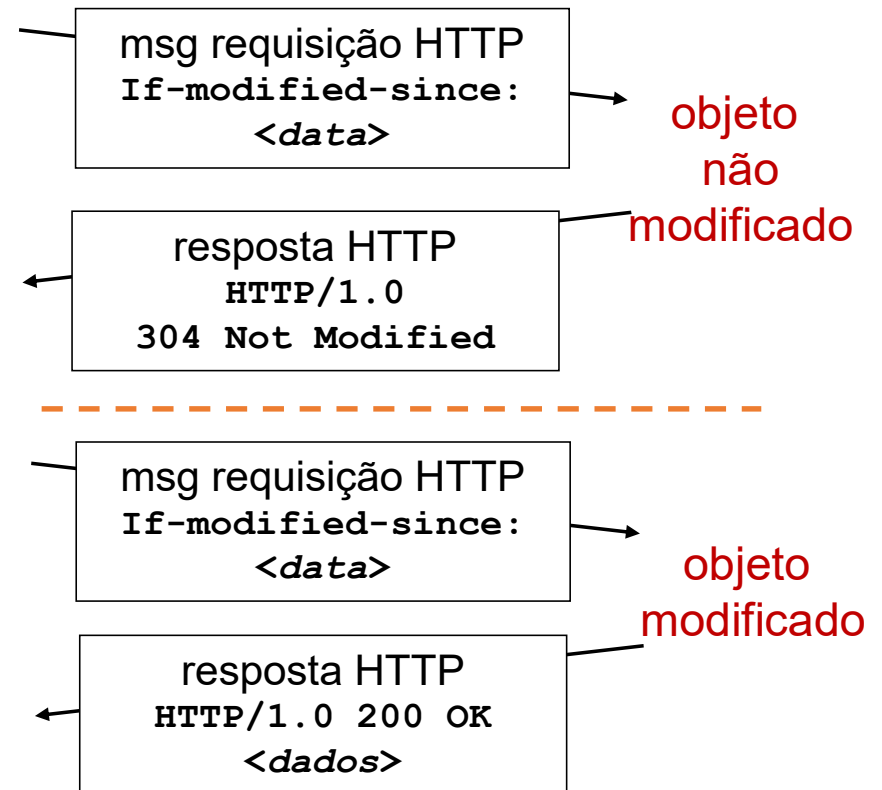
- Servidor: resposta não contém objeto se a cópia em cache estiver atualizada:

HTTP/1.0 304 Not Modified

Método GET condicional:

cache

servidor





Obrigado!

Referências :

KUROSE, J. F. e ROSS, K. W. **Redes de Computadores e a Internet – Uma Nova Abordagem**. Addison Wesley, 4ª Ed., 2010.



TANENBAUM, A. S. **Redes de Computadores** . 4ª Ed., Editora Campus (Elsevier), 2003.

Referências :

COMER, D. E. **Redes de computadores e internet**. 6. ed. Porto Alegre: Bookman, 2016.

The Internet Engineering Task Force (IETF®) - <https://www.ietf.org/>

Instituto de Engenheiros Eletricistas e Eletrônicos - <http://www.ieee802.org/>

Cisco Internet Business Solutions Group (IBSG) - <http://www.cisco.com/>



