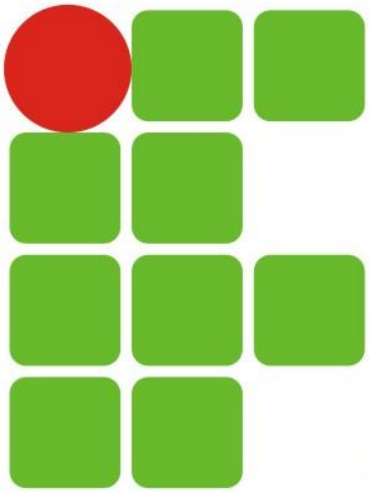


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Curso Superior em Redes de Computadores

HTTP

Objetivo

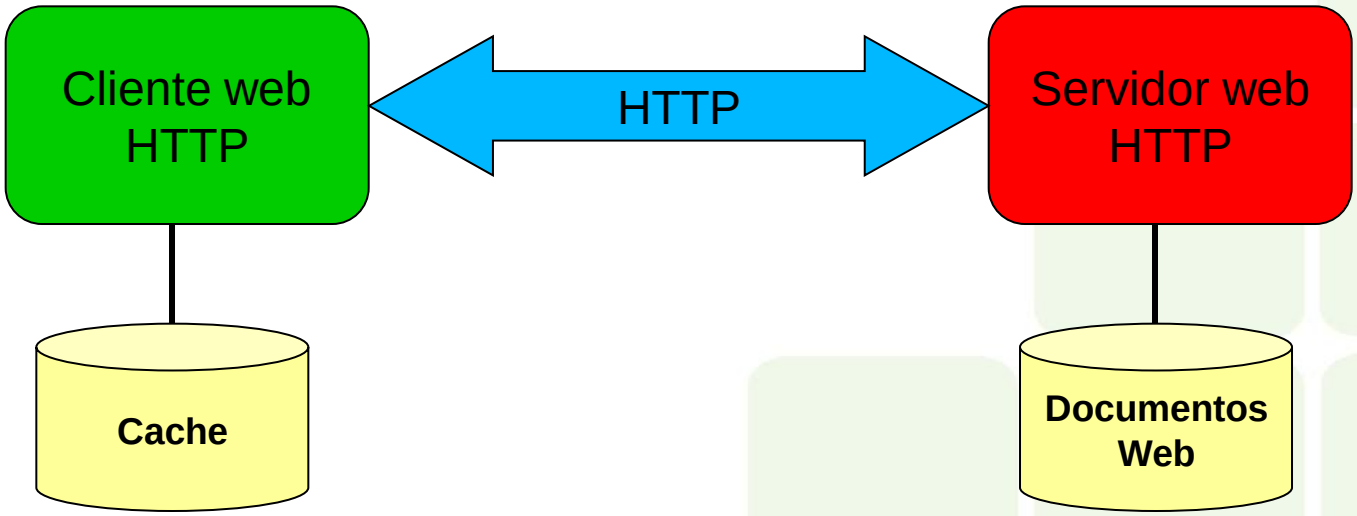
- Conhecer as características, funcionalidades e componentes do serviço de transferência de Hiper textos (***H****yper****T****ext* ***T****ransfer* ***P****rotocol* - **HTTP**)

HTTP
(*HyperText Transfer Protocol*)

- Características:
  - Implementa o serviço WWW – World Wide Web
  - Baseado no modelo Cliente-Servidor
  - Utiliza os serviços de transporte
    - Com conexão
      - Envio e recebimento de mensagens

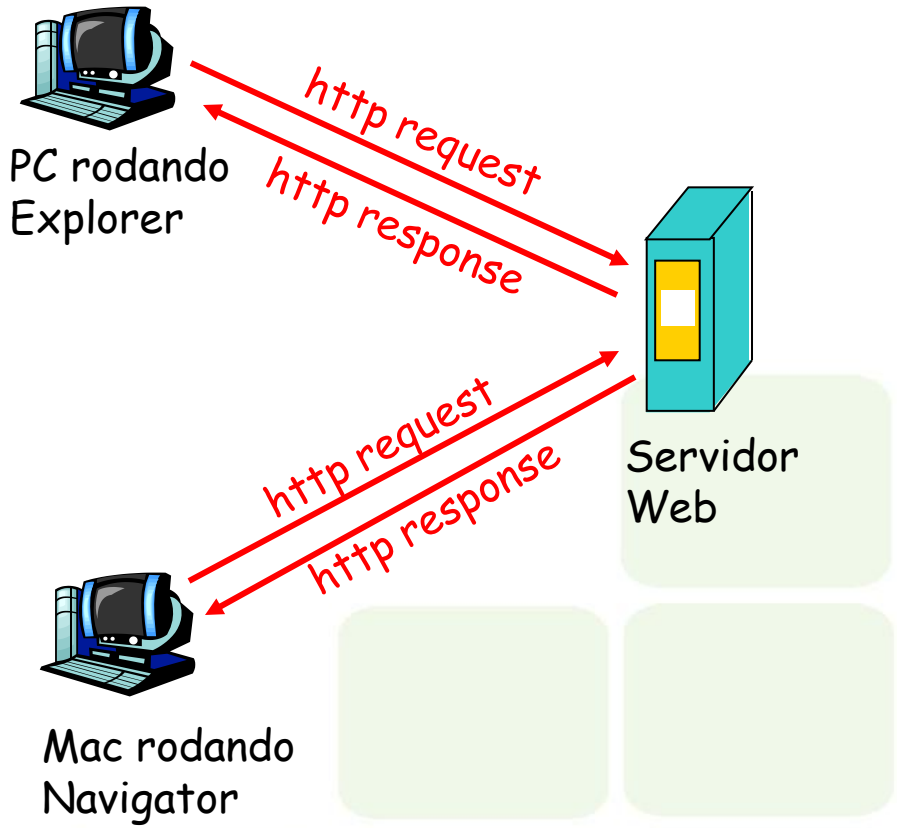
HTTP (*HyperText Transfer Protocol*)

- Características
  - Permite aos provedores de conteúdo a publicação de documentos
  - Permite aos usuários recuperar, visualizar e navegar nos documentos



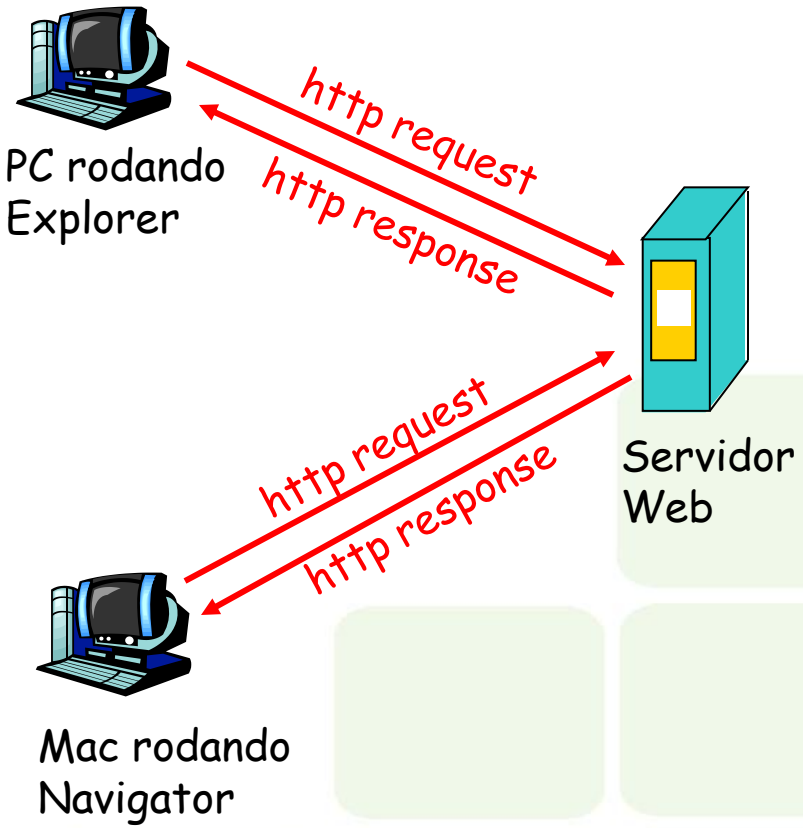
Protocolo HTTP
HyperText Transfer Protocol

- Protocolo que presta o serviço WEB.
- Uma página WEB geralmente contém um arquivo-base HTML que relaciona os outros objetos: arquivo HTML, imagem JPEG ou GIF, applet Java e Clipe de Áudio com seus URLs
- Cada URL (Uniform Resource Locator) indica o Server e o caminho para o objeto.



Protocolo HTTP
HyperText Transfer Protocol

- **Cliente:** browser que solicita, recebe e apresenta objetos da Web
- **Server:** abriga objetos, endereçados por URLs e os envia em resposta aos pedidos
  - Apache, MS Internet Information Server, Netscape Enterprise Server
- O Server HTTP não armazena informação sobre os Clientes. Assim, responderá ao Cliente mesmo que esse repita a solicitação em curto espaço de tempo (é um protocolo sem estado)



Versões: http1.0: (RFC 1945)
http1.1: (RFC 2616)

Protocolo HTTP
HyperText Transfer Protocol

- Utiliza o protocolo de transporte TCP:
  - Cliente inicia conexão TCP (cria socket) para o servidor na porta 80
  - Servidor aceita uma conexão TCP do cliente
  - Mensagens http (mensagens do protocolo de camada de aplicação) são trocadas, por demanda, entre o browser (Cliente http) e o Servidor Web (Servidor http)
  - O HTTP não se preocupa com detecção e correção de erros, reordenação, contenção/congestionamento
  - A conexão TCP é fechada

Conexões persistentes e não-persistentes

Não-persistente

- http/1.0, só utiliza esse modo
  1. O Cliente inicia a conexão TCP à Porta 80;
  2. O Cliente envia a requisição;
  3. O Servidor analisa pedido, envia resposta, aguarda o reconhecimento e fecha a conexão TCP;
  4. O Cliente recebe a mensagem;
  5. Os 4 primeiros passos são repetidos para cada objeto referenciado na página HTML
     - 2 RTTs para obter cada objeto (estabelecimento e transferência)
     - Uma Conexão TCP é gerada e fechada para cada objeto (operação em série)
     - Cada transferência sofre por causa do mecanismo de *partida-lenta* do TCP

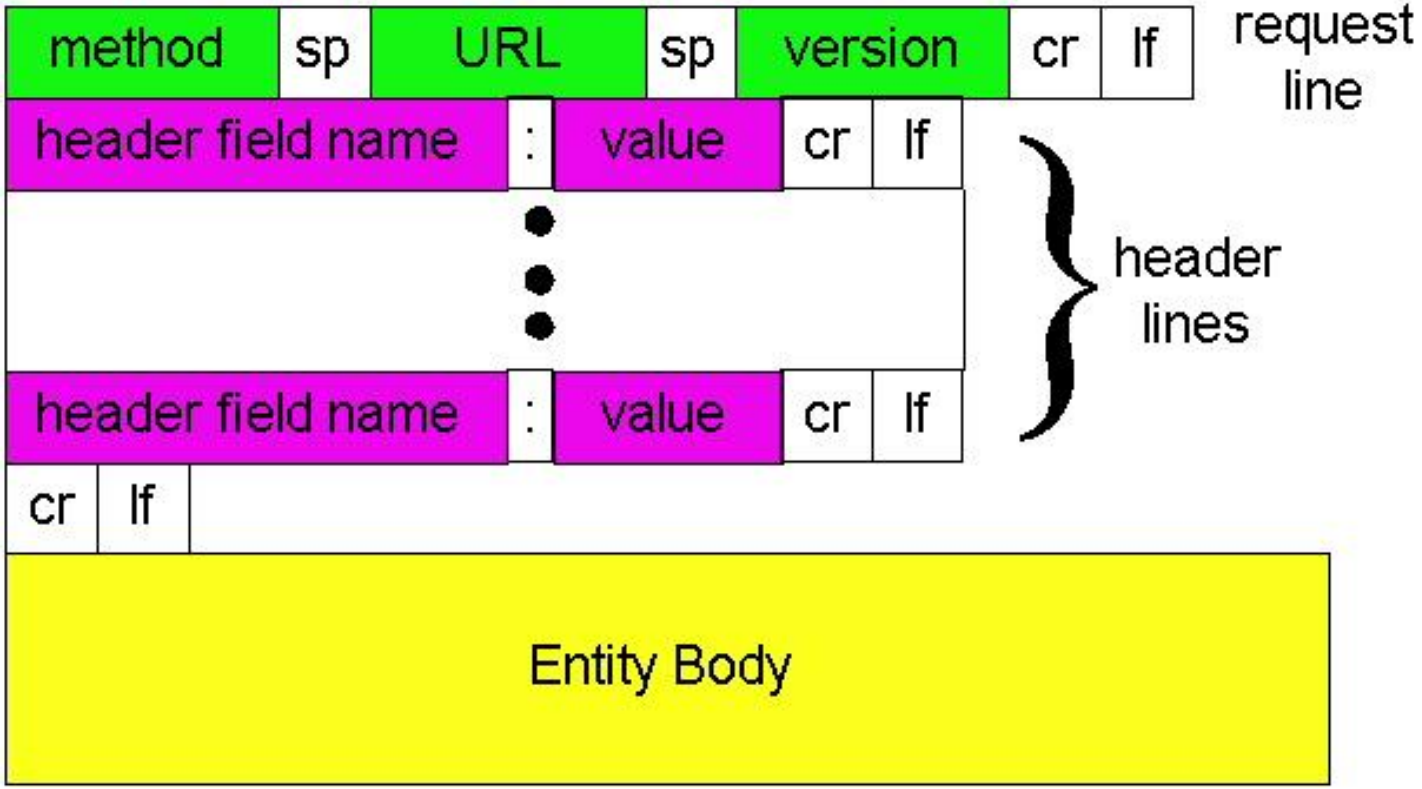
- Os browsers podem abrir várias conexões paralelas (default 05 a 10 - Configurável). Se 01, a operação será serial.

Conexões persistentes e não-persistentes

Persistente

- Para versão http/1.1, é o modo default
- Na mesma conexão TCP são recebidos vários (todos) objetos
- O cliente envia pedido para todos os objetos referenciados tão logo ele recebe a página HTML básica .
- O Server HTTP fecha a conexão após tempo de inatividade (configurável)
- Permite conexão com e sem Paralelismo
  - Sem Paralelismo, Cliente requisita e aguarda resposta (Half-duplex). Um (01) RTT (Round Trip Time) para cada objeto transferido
  - Com Paralelismo, poucos RTTs ou um RTT caso todos os Objetos sejam transferidos de uma só vez. Otimização dos recursos do Server.
- Poucos RTTs, menos slow-start devido ao TCP.

Formato das Mensagens
Formato Geral



Existem dois tipos de mensagens HTTP: **request e response**

Formato das Mensagens HTTP Request

- Escrita em texto ASCII (formato legível para humanos)

```
GET /somedir/page.html HTTP/1.1<CR LF>   {linha de requesição:Método=GET/POST/HEAD
                                          Objeto=URL e Versão do HTTP}
Host: www.someschool.edu<CR LF>          {indica o server onde o objeto reside}
Connection: close<CR LF>                 {o browser indica o não uso de conexão persistente}
User-agent: Mozilla/4.0<CR LF>           {especifica o tipo de browser. SERVER pode enviar
                                          versões diferentes de um mesmo objeto}
Accept: text/html, image/gif,image/jpeg<CR LF>
Accept-language:fr<CR LF>                {solicita a versão em francês do objeto}
<CR LF>                                  { extra <CR><LF>, indica fim da mensagem}
```

Cabeçalho

- O método GET não usa o campo CORPO DA ETIDADE.
- Já o método POST, possibilita através desse campo, o preenchimento de "formulário" que especifica o conteúdo desejado da página WEB. Por exemplo, uma palavra de busca.
- O método HEAD é semelhante ao GET, apenas requer uma resposta do Server. Isso é, não considera o CORPO DA ENTIDADE. É usado para fins de depuração pelos desenvolvedores de Servidores HTTP.

Formato das Mensagens HTTP Response

```
HTTP/1.0 200 OK <CR LF>          {Linha de Status:Contém a Versão, Cód.e Mensagem do Status.
                                  OK indica que o Server foi encontrado e enviou o OBJETO}
Connection: close<CR LF>         {indica que o Server fechará a conexão após envio do OBJ.}
Date: Thu, 06 Aug 1998 12:00:15 GMT <CR LF>   {data e hora do envio}
Server: Apache/1.3.0 (Unix) <CR LF>           {especifica o Server}
Last-Modified: Mon, 22 Jun 1998 …...<CR LF>   {data/hora que o OBJETO foi criado
ou sofreu a última atualização. Fundamental no processo de Cache do OBJETO no
Cliente/Servidores de Cache de Rede - Proxy}
Content-Length: 6821 <CR LF>     {indica tamanho em bytes do OBJETO}
Content-Type: text/html <CR LF>  {indica o tipo do OBJETO}

data data data data data ... <CR LF> <CR LF>  {Corpo da Entidade.Contém o OBJ}
```

Cabeçalho

Códigos de status das respostas e frases associadas

200 OK

- requisição bem-sucedida e objeto enviado ao Cliente

301 Moved Permanently

- objeto removido permanentemente. Um novo URL está especificado no cabeçalho Location: da resposta. O software do Cliente irá recuperar automaticamente a nova URL

400 Bad Request

- requisição não pode ser entendida pelo Server (erro)

404 Not Found

- documento requisitado não encontrado no Server

505 HTTP Version Not Supported

- Versão do HTTP requisitada não suportada pelo Server

Identificação de usuários Autenticação

- Restrição ao acesso através de username e senha

- O HTTP fornece codificação de status e cabeçalhos especiais para provê a autenticação:
  - O Cliente envia requisição comum, sem linhas de cabeçalhos especiais.
  - O Servidor responde com a codificação de status 401 Authorization Required e linha de cabeçalho WWW-Authorization: que detalha como proceder à autenticação (normalmente username e senha).
  - O Cliente solicita username e senha ao usuário e reenvia a requisição ao Servidor incluindo uma linha de cabeçalho Authorization:
  - Após recebimento do primeiro OBJETO, o Cliente continua enviando username e senha nas requisições subsequentes. Para essas requisições, o browser mantém o username e senha na memória cache e assim, não faz solicitação ao usuário.

Identificação de usuários Cookies

- Mecanismo alternativo que alguns sites podem usar para obter informações dos usuários (RFC 2109)
- Quando um Cliente acessa um site que usa COOKIES pela primeira vez, recebe um número de identificação através da linha de cabeçalho:

  Set-cookie: 1678453
- O Cliente anexa uma linha com o nome do Server e o ID associado ao usuário a um arquivo especial de COOKIES armazenado em seu HD.
- Nas requisições subsequentes a esse Servidor, o Cliente inclui a linha de comando:

  Cookie: 1678453

Identificação de usuários Cookies

- Os Servidores usam COOKIES para:
  - Liberar os usuários da digitação de username e senha a cada acesso;
  - Lembrar as preferências dos usuários, de forma a oferecer propaganda dirigida;
  - Para o caso de compras, anotar itens que o usuário está escolhendo (carrinho de compras)
- O usuário nômade, que acessa o mesmo site de várias máquinas, não goza de tal facilidade.
- Segurança é outro item que deve ser considerado pelo usuário quando do uso de COOKIES

GET Condicional
Cache no Cliente ou em Servidor

- Web Caches armazenam OBJETOS extraídos anteriormente;
- Possibilitam a redução do tempo para novas extrações;
- Reduzem o tráfego na Rede;
- Podem residir em uma estação Cliente ou em um Servidor de Cache intermediário da rede - Proxy.

GET Condicional
Cache no Cliente

Para a ocorrência de uma nova requisição, há necessidade de verificação da atualização do OBJETO.

Isso é realizado por uma requisição HTTP, GET Condicional, que contém a linha de cabeçalho If-Modified-Since: < data = Last-Modified>

O Server responde a requisição com ou sem o OBJETO, para os casos de ter ou não havido modificação.

```
GET /somedir/page.html HTTP/1.1<CR LF>
User-agent: Mozilla/4.0<CR LF>
If-Modified-since: Mon, 22 Jun 1998 09:23:24 <CR LF>
-----------------------------------------
HTTP/1.0 304 Not Modified<CR LF>
Date: Thu, 12 Aug 1998 15:39:25 GMT<CR LF>
Server: Apache/1.3.0 (Unix) <CR LF>
```

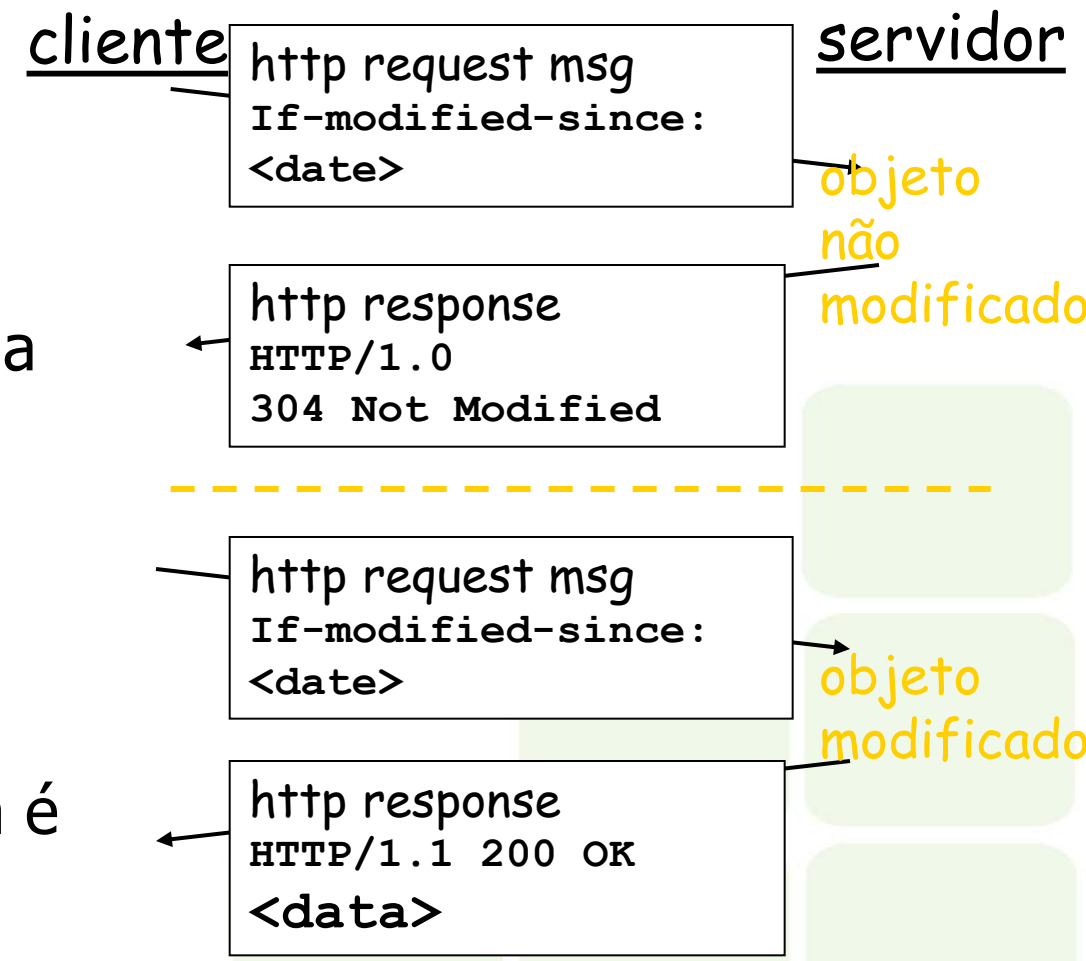
GET Condicional Cache no Cliente

- Razão: não enviar objeto se a versão que o cliente já possui está atualizada.
- Cliente: especifica data da versão armazenada no pedido HTTP

```
If-modified-since:
  <date>
```

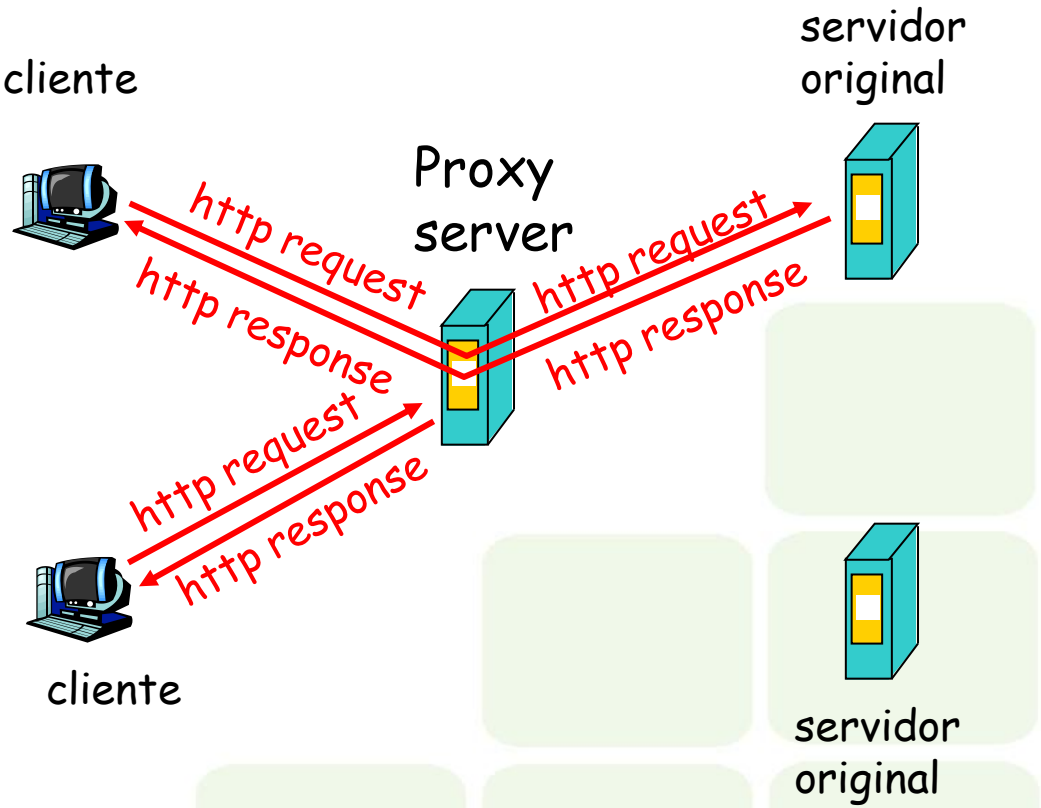
- servidor: resposta não contém objeto se a cópia é atualizada:

```
HTTP/1.0 304 Not
  Modified
```



GET Condicional
Web Caches - Server Proxy

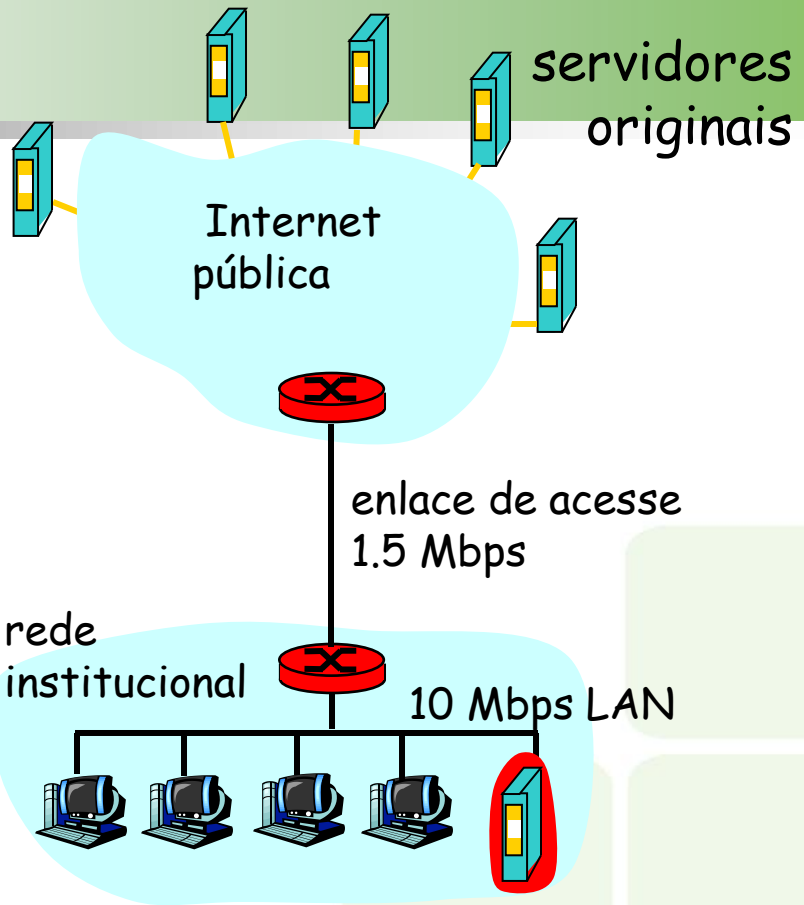
- Armazena e conserva cópias de Objetos recentemente requisitados
- Atende o Cliente sem envolver o Servidor Web originador da informação.
- Usuários configuram browser para que acessos à WEB sejam dirigidas ao WEB CACHE
  - Se o objeto existe no WEB CACHE: WEB CACHE retorna para o Cliente;
  - Senão, solicita objeto do Servidor original, envia ao Cliente e armazena.
- O Proxy se comporta como Server e Cliente



Porque Web Caching?

Vantagens:

- Reduz consideravelmente o tempo de resposta pois a Largura de Banda entre Cliente/Proxy é geralmente maior que entre Cliente/Server.
- Reduz o tráfego de uma instituição à Internet. Isso reduz custos com ampliação de Largura de Banda, além do tréfego total da Rede (75% WEB, em 1998)

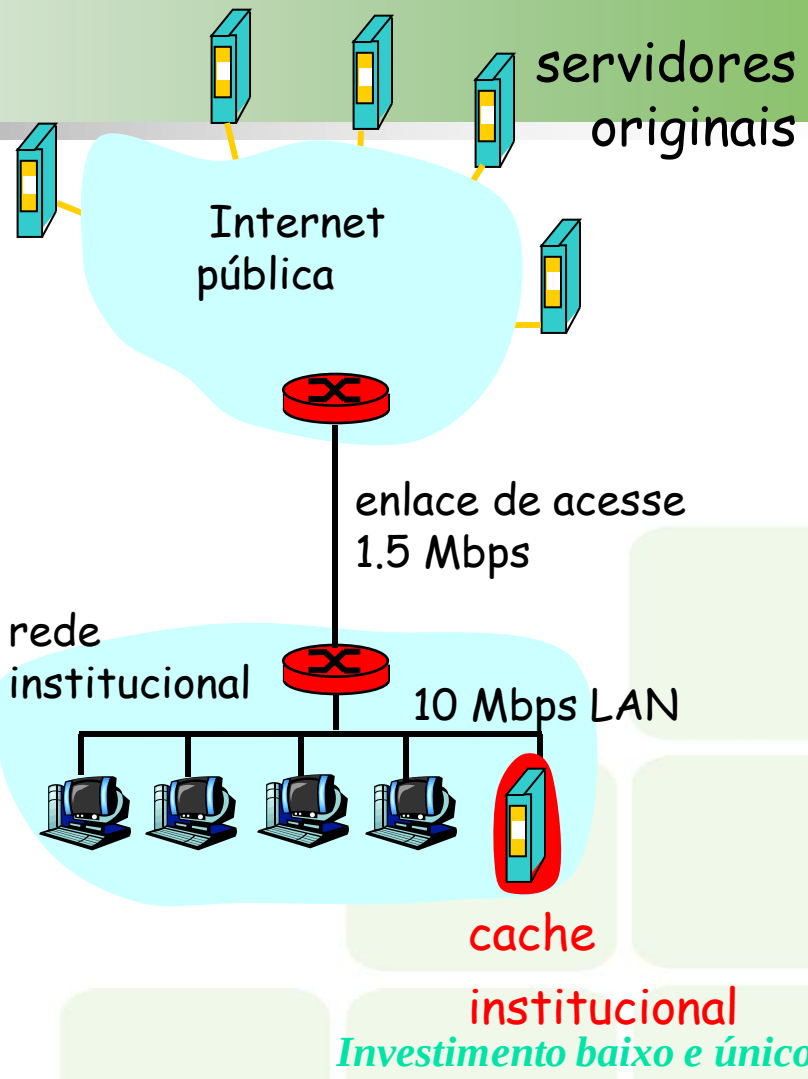


Investimento baixo e único

Porque Web Caching?

Vantagens:

- Para uma Internet com alta densidade de Caches (Institucional, Regional e Nacional), otimiza a distribuição de conteúdo pertencentes a Servidores de baixa velocidade devido a capacidade de processamento ou largura de banda dos enlaces que os suportam.
- O custo de aquisição e instalação de WEB Cache (requer PC e software de domínio público), além de ser um único investimento, é bem menor que a contratação de aumento de banda do acesso.



Caches Cooperativos

- WEB Caches, localizados em diferentes lugares da Internet, podem funcionar de modo cooperativo de forma a melhorar o desempenho geral
- Por exemplo, um Cache Institucional pode ser configurado para enviar requisições HTTP para um Cache do IAP do backbone Nacional. Sempre que um Objeto passa por uma Cache Institucional ou Nacional, este armazena uma cópia em seu arquivo local.
- O NLANR é um exemplo de Sistema Cooperativo de Cache, formado por Caches de backbones nos EUA que fornecem serviços aos Caches Institucionais, Regionais e Nacionais no mundo inteiro. Os Caches obtêm Objetos uns dos outros usando uma combinação de HTTP e ICP (Internet Caching Protocol)
- O ICP é um protocolo da camada de aplicação que permite uma consulta muito rápida de um WEB Cache a outro sobre a existência de um dado documento. Então, caso afirmativo, o WEB Cache extrai o Objeto usando o HTTP.
- O ICP é largamente usado em Sistemas de Cache Cooperativos.

Atividade

- Fazer o recebimento de documentos web utilizando comandos modo terminal através de telnet
- Monitorar as conexões e os comandos do protocolo HTTP
  - Verificar as conexões persistentes
  - Verificar as conexões não-persistentes e versões do protocolo

Referências

- Comer, Douglas E., Interligação de Redes Com Tcp/ip
- James F. Kurose, Redes de Computadores e a Internet
- Escola Superior de Redes, Arquitetura e Protocolos de Redes TCP/IP