ADMINISTRAÇÃO DE REDES DE COMPUTADORES

Hyper Text Transfer Protocol (HTTP)

8/11/2005

Engª de Sistemas e Informática
Licenciatura em Informática

UALG/FCT/DEEI 2005/2006

The World Wide Web - História

Surgiu da necessidade de partilhar informação entre cientistas

Cronologia

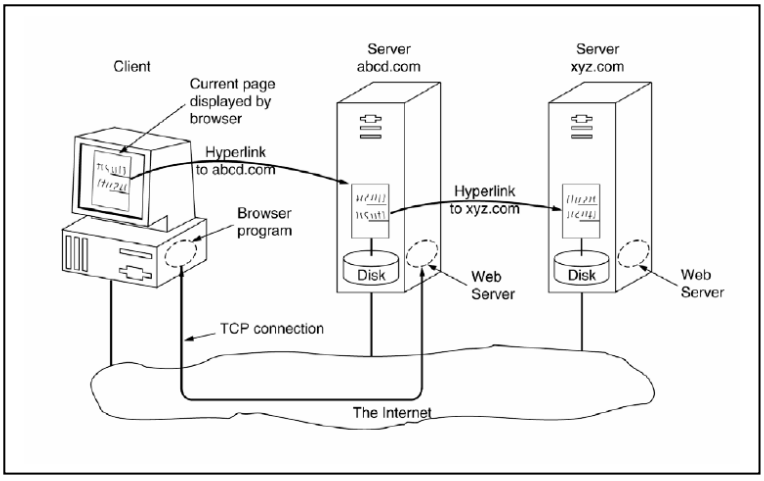
- 1945, July: Vannevar Bush idealiza um sistema semelhante à WWW
- 1989, Março: Proposta por Tim Berners-Lee ao CERN, que impulsionou o lançamento da WWW http://www.w3.org/History/1989/proposal.html
- 1991, 17 Maio: o CERN cria e lança internamente a WWW, mas apenas com um interface textual (info.cern.ch foi o primeiro servidor Web)
- 1991, Dezembro: Primeira demonstração pública na conferência Hypertext '91
- 1991, 12 Dezembro: Primeiro servidor fora da Europa
- 1993, Fevereiro: NCSA cria o Software Mosaic para navegação
- 1994: Netscape cria uma versão comercial de um browser
- 1996: Microsoft Internet Explorer

Tim Berners-Lee 

The World Wide Web - História

A normalização do WWW é realizada no âmbito do Consórcio W3 (www.w3.org), sendo algumas das normas posteriormente publicadas pela IETF como RFCs. A pressão comercial levou a que a normalização ocorra muitas das vezes após o lançamento de produtos com modificações à norma anterior.

The World Wide Web –Ponto de vista do utilizador



Para o utilizador

O WWW é uma forma de obter informação imediatamente disponível na Internet como se fosse um meio contínuo pesquisável.

Recorrendo a saltos e pesquisas em hipertexto, o utilizador navega através de um mundo de informação em parte escrito à mão, e em parte gerado por computador a partir de bases de dados e sistemas de informação existentes.

O WWW define:

• A ideia de um mundo onde cada pedaço de informação tem uma referência pela qual pode ser acedido;

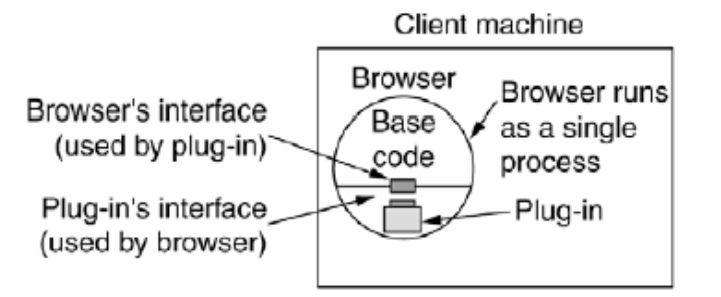
• Um sistema de endereçamento (URL - *Uniform Resource Locator*), que permite endereçar vários tipos de objectos acessíveis através de protocolos já em uso, tais como FTP, NNTP, telnet, e HTTP (e outros mais antigos: Gopher, WAIS);

• Um protocolo de nível aplicação (HTTP – *Hypertext Transfer Protocol*) oferecido pelos servidores WWW genuínos para transferência de ficheiros entre clientes e servidores;

• Uma linguagem de hipertexto com marcas de formatação (HTML - *Hypertext Markup Language*) que todos os clientes WWW devem entender, e informação, tais como texto, imagens, menus e informação sobre a formação da informação no cliente.

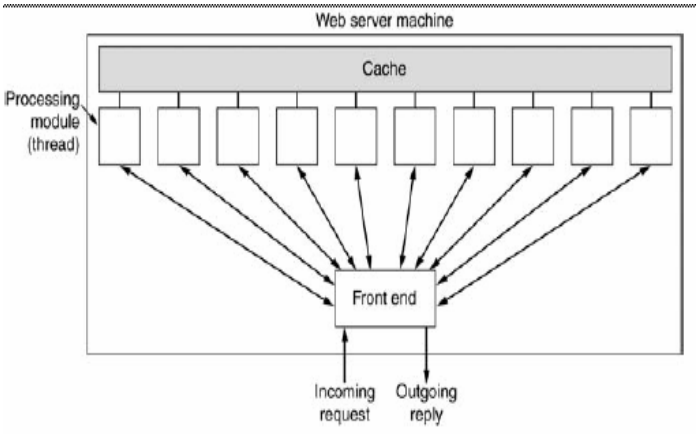
Como funciona a WWW

Utilizando um browser o cliente realiza o carregamento de ficheiros a partir de um servidor e apresenta os ficheiros recebidos.
Caso o *browser* não suporte algum tipo de dados, pode recorrer a um *plug-in* (a) (e.g. Macromedia shockwave) ou a uma aplicação externa (e.g. Acrobat PDF Reader).



The World Wide Web –Ponto de vista do utilizador

Um servidor Web é numa visão simplista um servidor de ficheiros com requisitos elevados de desempenho, que recebe pedidos e os satisfaz (de preferência) em paralelo.



A operação mais lenta é o acesso ao ficheiro, pode ser melhorada com a memorização do conteúdo dos últimos ficheiros abertos.

The World Wide Web –Ponto de vista da Rede

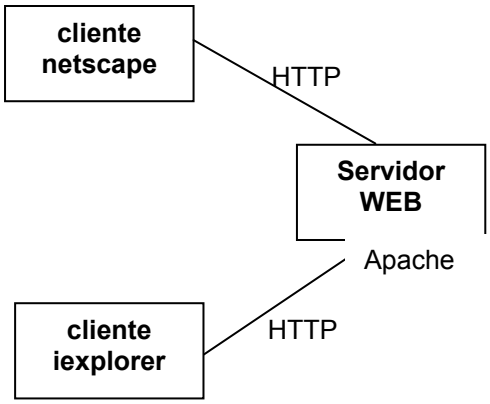
O protocolo HTTP (Hyper Text Transfer Protocol)

-protocolo de nível de aplicação para a Web
-protocolo que formula pedidos inteligíveis em ASCII

-Suportado por TCP mas também AAL5 (redes ATM)

-modelo cliente/servidor
cliente*:* navegador que pede, recebe e mostra objectos Web
servidor servidor Web que envia objectos em respostas a pedidos

http1.0: RFC 1945
http1.1: RFC 2616



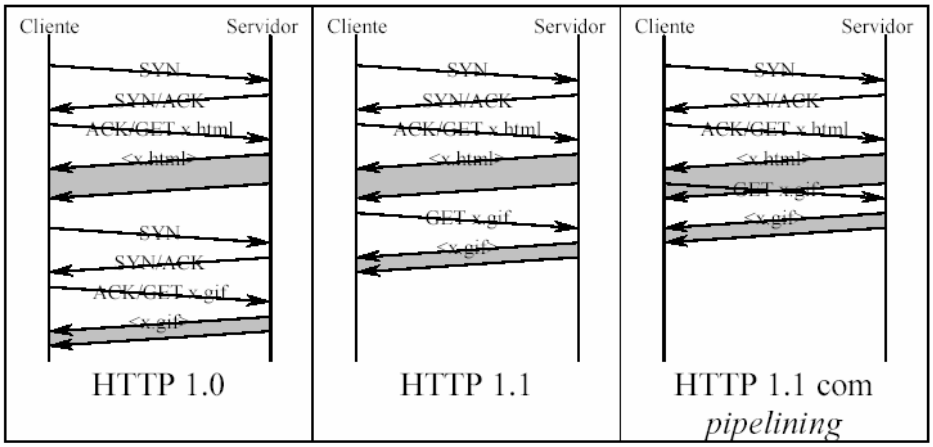
Transação HTTP

- cliente inicia ligação TCP (cria um socket) para o servidor (porto 80 por defeito)
- servidor aceita ligação TCP do cliente.
- mensagens http (mensagens do protocolo da camada de aplicação) são trocadas entre o navegador (cliente http) e o servidor Web (servidor http)
- ligação TCP fechada

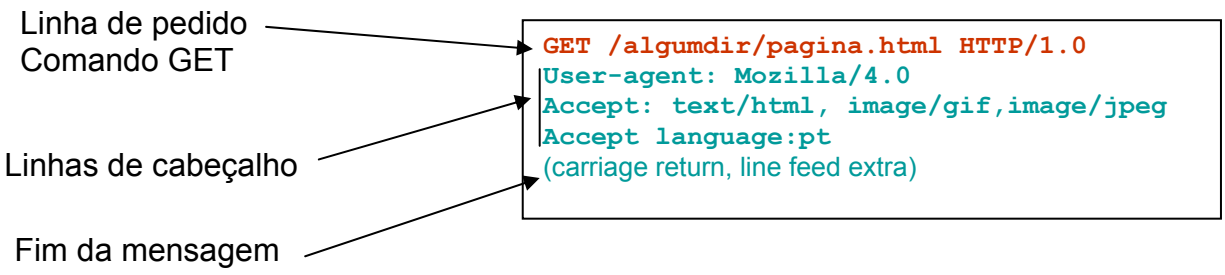
Até 1997 foi usada a versão 1.0 do protocolo (*RFC 1945*) que envia cada pedido a um servidor por uma ligação TCP independente. Cada ficheiro de uma página é enviado por uma ligação diferente. A versão 1.1 (*RFC 2616*) corrige esta limitação, passando a suportar a reutilização de ligações (a ligação termina após um período de inactividade).

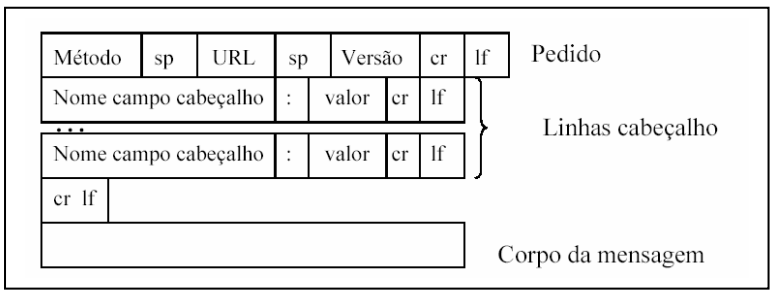
Transação HTTP

A versão 1.1 permite dois modos de funcionamento: Sem *pipelining*, o pedido de cada ficheiro da página é realizado após receber a resposta ao pedido anterior.
Com *pipelining* vários pedidos podem ser enviados sem esperar pela resposta do primeiro, sendo as resposta recebidas pela ordem que são feitos os pedidos.



Formato da mensagem http: pedido

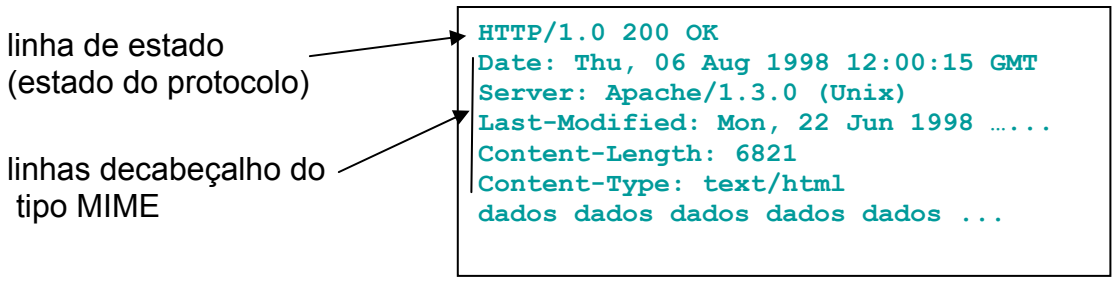




Métodos predefinidos:

GET Requer a leitura de uma página
HEAD Requer a leitura do cabeçalho de uma página
PUT Requer a gravação de uma página
POST Requer o acrescentar de dados a uma página
DELETE Apaga uma página
TRACE Ecoar pedido recebido
CONNECT Reservado para uso futuro
OPTIONS Requer informação sobre opções disponíveis

Formato da mensagem http: resposta



Os códigos de estado devolvidos podem ser:
1xx; Informação: Recusa. pedido, continua processamento
2xx; Sucesso: Acção terminada com sucesso
3xx; Redirecção: Necessárias mais acções para completar
4xx; Erro do cliente: Pedido errado, não pode ser executado
5xx; Erro do servidor: Servidor falhou com pedido válido

Exemplos:

100 Agree: **Servidor aceita processar pedido**
200 OK**: Sucesso, informação retornada no corpo da mensagem**
204 No Content**: Ficheiro vazio**
301 Moved Permanently**: Moveu-se para URL em '**Location:**'**
304 Cached Page Valid: **Página em cache ainda é válida**
400 Bad Request**: Pedido não entendido pelo servidor**
401 Unauthorized**: Requerida autenticação do cliente**
403 Forbidden Page: **Página não acessível**
404 Not found**: O ficheiro pedido não existe**
501 Internal Error
503 Try Again Later
505 HTTP Version Not Suported: **Versão não suportada**

Formato da mensagem HTTP

Cabeçalho	**Tipo**	**Conteúdo**
User-Agent	Pedido	Informação sobre o browser e a sua plataforma
Accept	Pedido	Os tipos de páginas que o cliente suporta
Accept-Charset	Pedido	Os códigos de caracteres suportados pelo cliente
Accept-Encoding	Pedido	As codificações de página suportadas
Accept-Language	Pedido	As línguas suportadas (português, inglês, …)
Host	Pedido	Nome DNS do servidor
Authorization	Pedido	Lista de credenciais do cliente
Cookie	Pedido	*cookies* previamente definidos pelo servidor
Date	Ambos	Data e hora de envio da mensagem
Upgrade	Ambos	Protocolo pretendido pelo emissor
Server	Resposta	Informação sobre o servidor
Content-Encoding	Resposta	Como o conteúdo está codificado (e.g. gzip)
Content-Language	Resposta	Língua usada na página
Content-Length	Resposta	Comprimento da página (bytes)
Content-Type	Resposta	Tipo MIME da página
Last-Modified	Resposta	Data e hora de última modificação da página
Location	Resposta	Comando para enviar o cliente para outro URL

Formato da mensagem HTTP

Accept-Ranges	Resposta	O servidor aceita pedidos de blocos de bytes
Set-Cookie	Resposta	O servidor quer que o cliente guarde um *cookie*

Exemplo de Interacção

Pedido:

GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Accept-language:pt

Resposta:

HTTP/1.1 200 OK
Date: Thu, 23 Oct 2002 12:30:00 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Thu, 20 Oct 2002 10:00:00 GMT
Content-Length: 6821
Content-Type: text/html
… { dados dados dados } …

Cookies

- O protocolo HTTP não guarda nenhuma memória sobre interacções anteriores de um utilizador.
- As extensões para manter um "estado" na interacção utilizador-servidor recorrem a campos de cabeçalho mantidos no cliente, que são enviados em todos os pedidos do cliente.

Cada *cookie* é uma cadeia de caracteres com até 4KB.

Existem 3 versões:

V0 –Netscape (1995) *http://wp.netscape.com/newsref/std/cookie_spec.html*
V1 – RFC 2109 (1997) – acrescenta controlo de versões
V2 – RFC2965 (2000) – usa Set-Cookie2/Cookie2

Descreve-se a V0, suportada por TODOS os browsers:
Set-Cookie: NAME=VALUE; Expires=DATE; Path=PATH;

Caching no cliente

Até quando é que o documento guardado é válido? Heurística baseada na data da última modificação.
O HTTP suporta o campo de cabeçalho "If-modified-since:" que permite validar se um documento ainda é válido.

Pedido:

GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
If-Modified-Since: Thu, 23 Oct 2002 12:00:00 GMT
…

Resposta:

HTTP/1.0 304 Not Modified
Date: Thu, 23 Oct 2002 12:35:00 GMT
Server: Apache/1.3.0 (Unix)

O HTTP permite evitar a utilização da cache no cliente.

N o HTTP 1.0 existe um cabeçalho "pragma: no-cache".

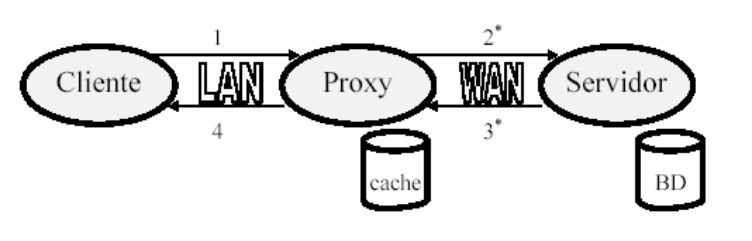
No HTTP 1.1 foi criado um campo de cabeçalho "cachecontrol:" para o pedido e para a resposta onde se pode definir o valor "no-cache", mas também o tempo máximo que a cópia permanece válida "max-age".

Cache em Proxys

Para além dos clientes, as respostas dos servidores também podem ser guardadas em servidores intermediários entre os clientes e os servidores WWW, designados de *proxy*.

Após receber um pedido de um cliente (1), reenvia-o para o servidor pretendido (2). A resposta recebida (3) é enviada de volta ao cliente (4), mas também é armazenada localmente.

Caso um novo cliente faça o mesmo pedido dentro de um intervalo de tempo, o proxy retorna a informação em cache.



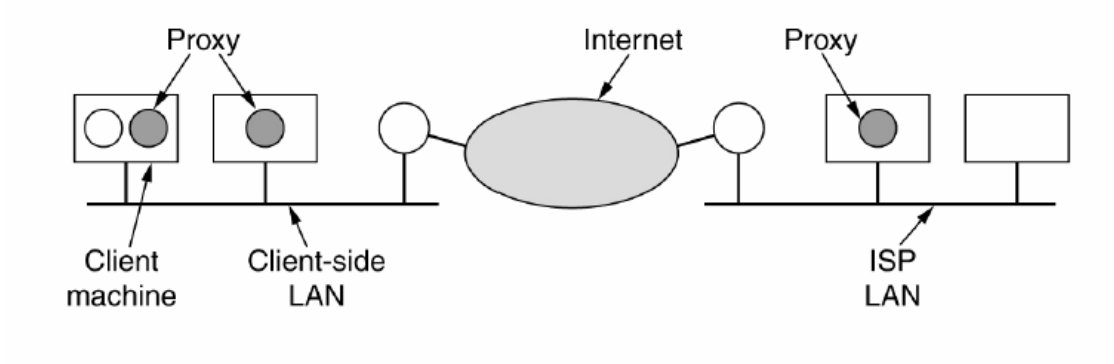
Cache em Proxys

A utilização de *proxies* numa organização com uma rede de informação interna rápida:

- reduz a quantidade de tráfego trocada com o exterior;
- reduz o tempo médio de acesso à rede;
- permite simplificar a configuração de *firewalls* ao permitir limitar o acesso ao exterior (para WWW) apenas à máquina onde o proxy está a correr;
- tem a desvantagem de as páginas em cache poderem estar desactualizadas. Os utilizadores podem sempre forçar uma nova leitura no servidor, usando os campos de controlo de caching.
- O HTTP 1.1 suporta proxies, existindo opções no campo de cabeçalho "cache-control:" destinadas a definir o nível de partilha entre utilizadores de uma página (e.g. *private, public, s-maxage (max-age* para proxies)).

Cache em Proxys

Vários *proxies* podem ser ligados criando um serviço hierárquico de caching distribuído por toda a rede (e.g. Squid, Apache).



Neste caso pode-se usar um protocolo mais complexo de coordenação (e.g. ICP – *Internet Cache Protocol* – RFC 2186).