

Chapter 2

Application Layer

Reti di Elaboratori

Corso di Laurea in Informatica

Università degli Studi di Roma "La Sapienza"

Canale A-L

Prof.ssa Chiara Petrioli

Parte di queste slide sono state prese dal materiale associato al libro
Computer Networking: A Top Down Approach, 5th edition.

All material copyright 1996-2009

J.F Kurose and K.W. Ross, All Rights Reserved

Thanks also to Antonio Capone, Politecnico di Milano, Giuseppe Bianchi and
Francesco LoPresti, Un. di Roma Tor Vergata

Trasporto dei messaggi

- Supponiamo che un client richieda una pagina HTML di un server al cui interno sono contenuti i riferimenti ad altri oggetti (ad esempio 10 figure che compongono la pagina e che occorre visualizzare insieme al testo HTML).

Antonio Capone Home Page

Title: Assistant Professor
Section: [Telecommunication](#)
Research Area: [Telecommunication Networks](#)
Courses: (in italian)

- [Infrastrutture e Protocolli per Internet](#)
- [Comunicazioni elettriche e reti di telecomunicazione DU](#)
- [Reti di Telecomunicazione II \(Prof Fratta\)](#)
- [Laboratorio di Fondamenti di Telecomunicazione](#)

Politecnico di Milano
Dipartimento di Elettronica e Informazione
piazza Leonardo da Vinci 32
20133 Milano, Italy

floor: 3
room: 330
phone: +39.02.2399.3413
fax: +39.02.2399.3413

capone@elet.polimi.it

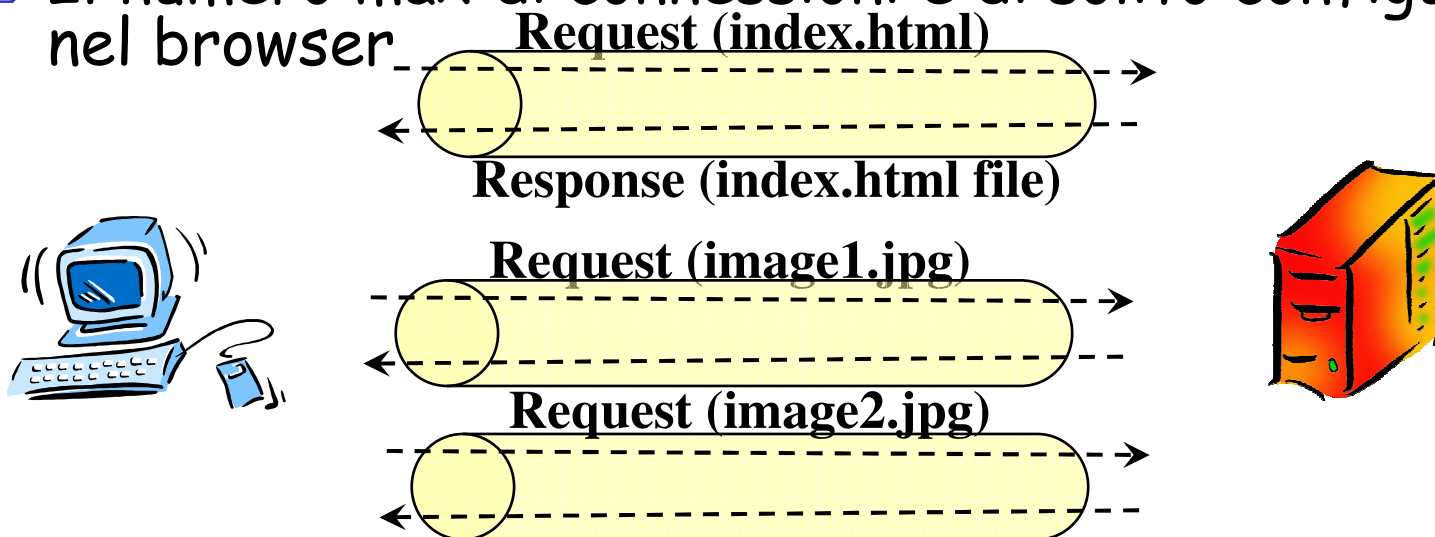
[Thesis work proposals \(in italian\)](#)

■ nel trasferimento dell'insieme di oggetti sono possibili 2 modalità

- Non-persistent connection (default mode di HTTP 1.0)
- Persistent connection (default mode di HTTP 1.1)

Non persistent

- ❑ Viene aperta una connessione per una sola request-response: inviato l'oggetto, il server chiude la connessione
- ❑ La procedura viene ripetuta per tutti i file collegati al documento HTML base
- ❑ Le connessioni TCP per più oggetti possono essere aperte in parallelo per minimizzare il ritardo
- ❑ Il numero max di connessioni è di solito configurabile nel browser

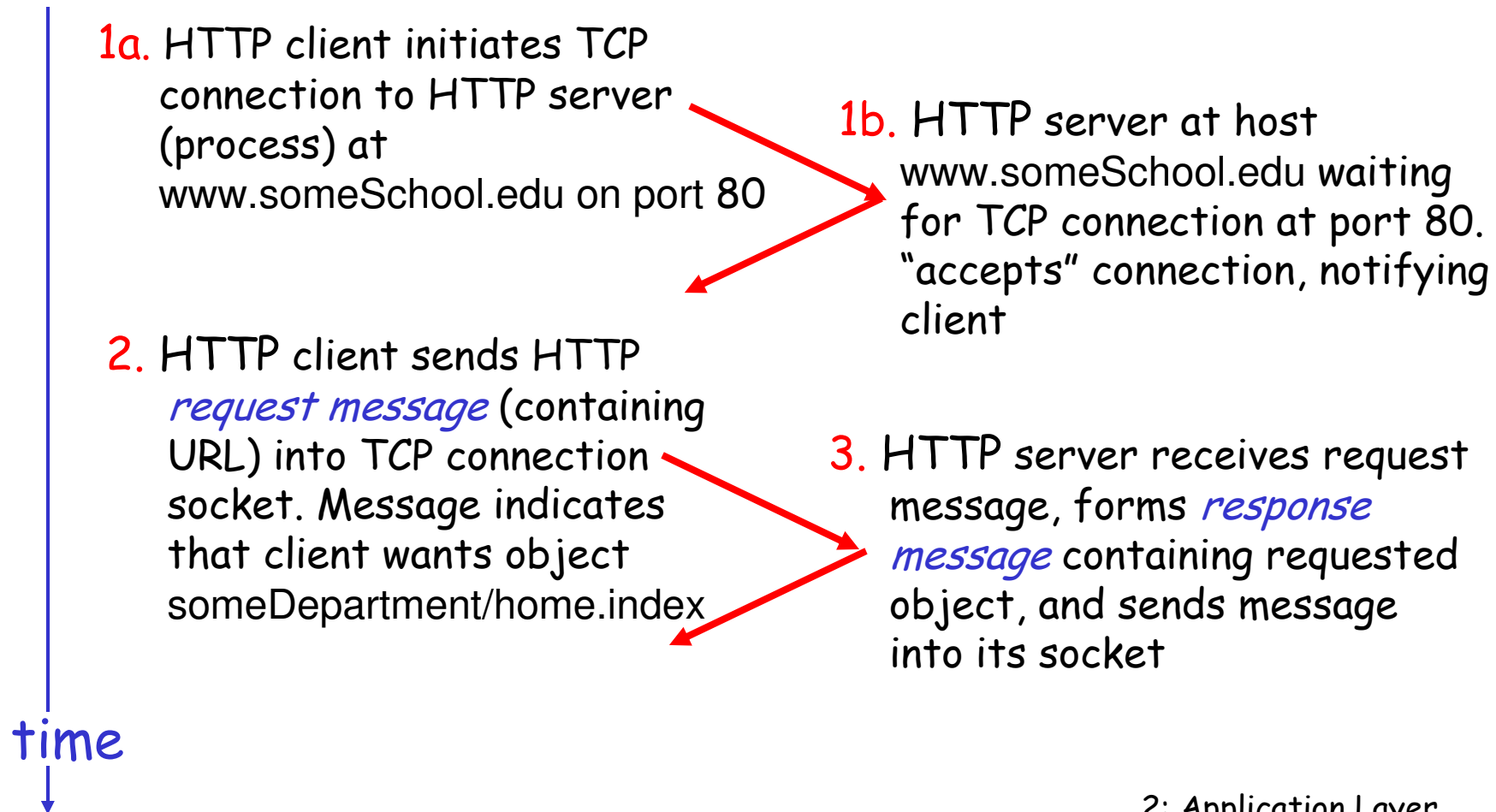


Nonpersistent HTTP (HyperText Transfer Protocol)

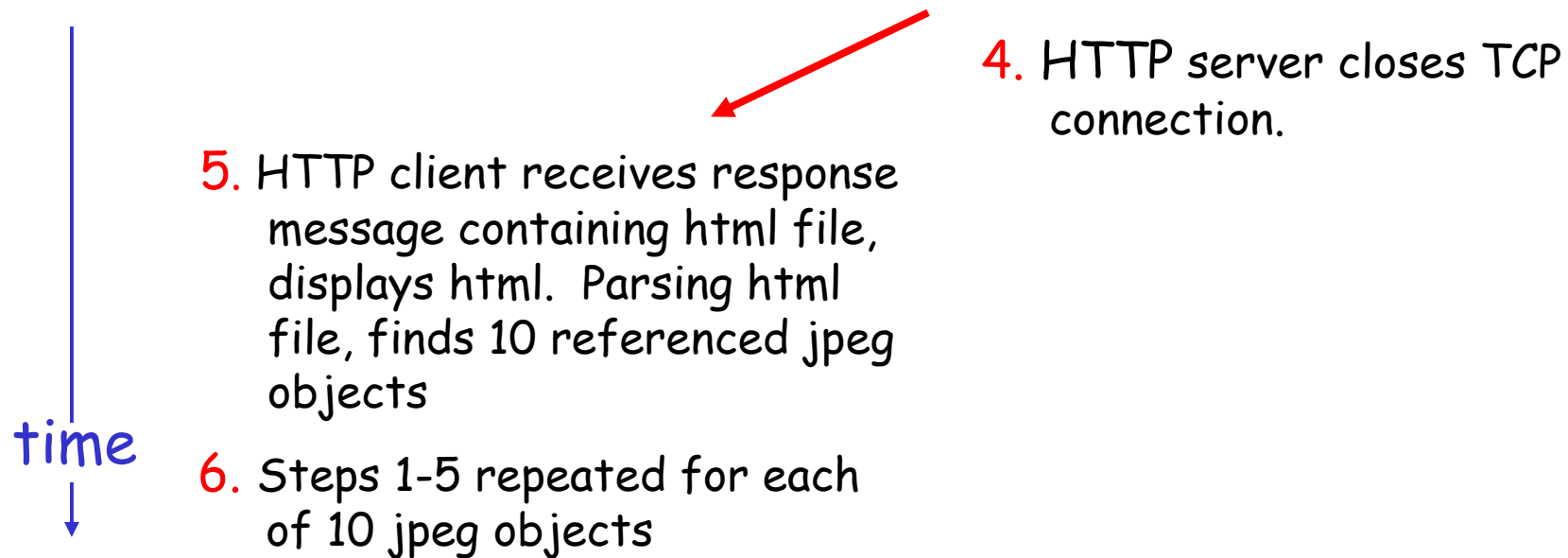
Suppose user enters URL

`www.someSchool.edu/someDepartment/home.index`

(contains text,
references to 10
jpeg images)



Nonpersistent HTTP (cont.)



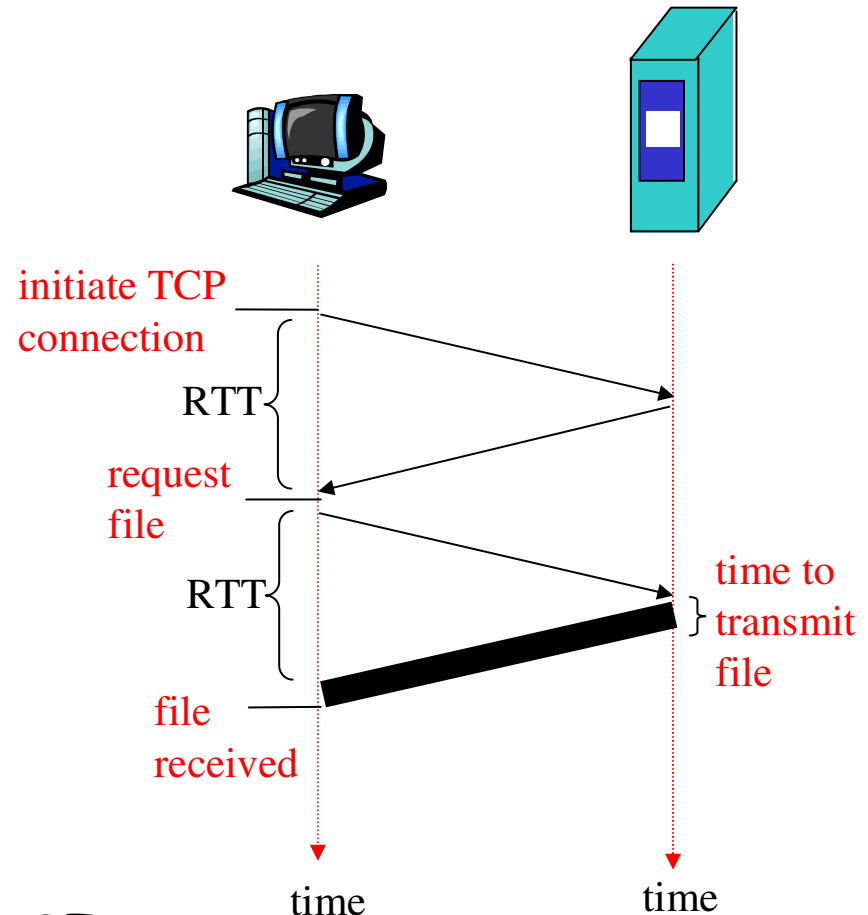
Response time modeling

Definition of RRT: time to send a small packet to travel from client to server and back.

Response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time

total = $2RTT + \text{transmit time}$



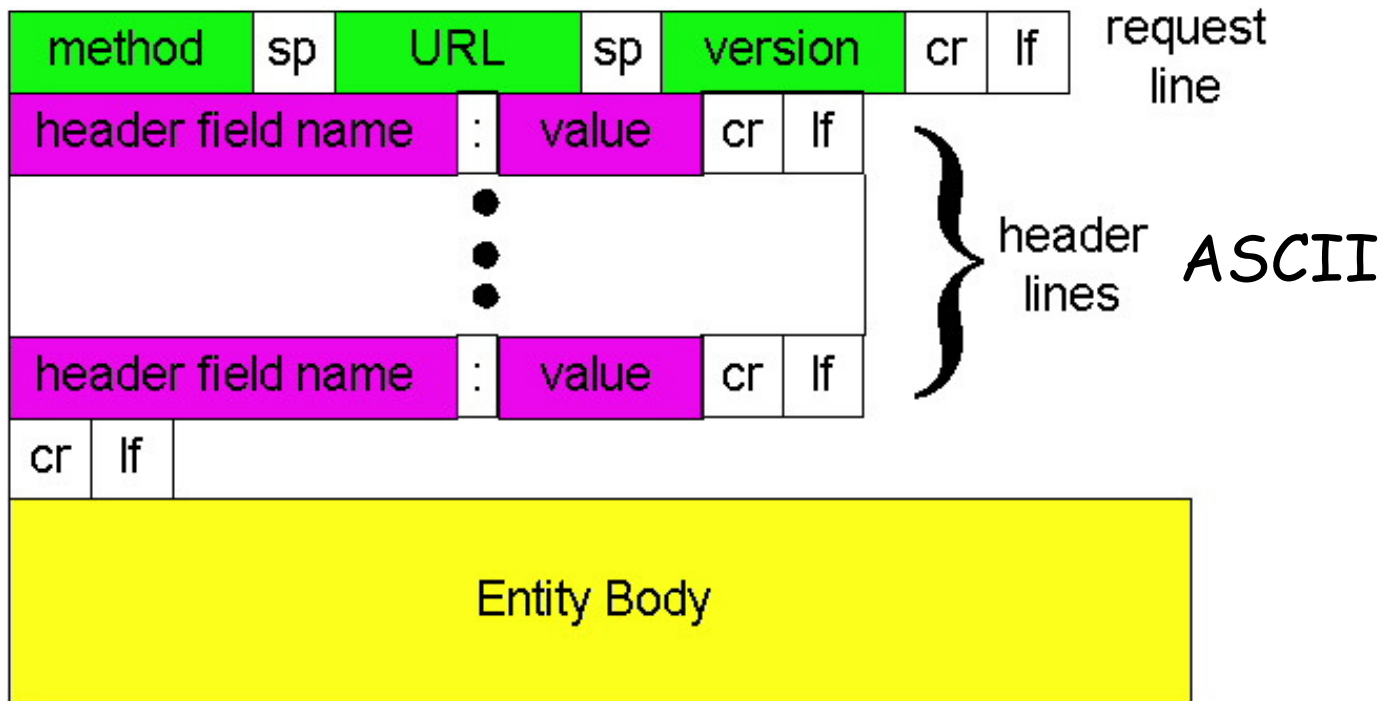
Problema: OGNI VOLTA CHE SI APRE UNA CONNESSIONE SI PAGA 2RTT PRIMA DI SCAMBIARE DATI

HTTP messages

- ❑ Sequence of octets sent over a transport connection; basic unit of communication in HTTP
- ❑ Can be a request message from a client to a server or
- ❑ a response sent back from a sender to a client

HTTP Messages

Request



HTTP Message

Methods:

GET	E' usato quando il client vuole scaricare un documento dal server. Il documento richiesto è specificato nell'URL. Il server normalmente risponde con il documento richiesto nel corpo del messaggio di risposta.
HEAD	E' usato quando il client non vuole scaricare il documento ma solo alcune informazioni sul documento (come ad esempio la data dell'ultima modifica). Nella risposta il server non inserisce il documento ma solo degli header informativi.
POST	E' usato per fornire degli input al server da utilizzare per un particolare oggetto (di solito un applicativo) identificato nell'URL.
PUT	E' utilizzato per memorizzare un documento nel server. Il documento viene fornito nel corpo del messaggio e la posizione di memorizzazione nell'URL.

■ Altri methods:

- DELETE, LINK, UNLINK, ...

Per inviare info al server

E' possibile anche usare GET

GET /sercah.cgi?string=greek-architects HTTP/1.0

Header

Header name : **Header value**

- ❑ Gli header servono per scambiare informazione di servizio aggiuntiva
- ❑ E' possibile inserire più linee di header per messaggio
- ❑ Esempi

Cache-control	Informazione sulla cache
Accept	Formati accettati
Accept-language	Linguaggio accettato
Authorization	Mostra i permessi del client
If-modified-since	Invia il doc. solo se modificato
User-agent	Tipo di user agent

Header Types

- General: used in request and response messages
- Request header (e.g. to express preferences on the nature of the response, include additional info with the request, to specify a constraint on the server in handling the request)
- Response header: to provide additional info about the response or to request additional info from the user
- Entity header (both in request and response messages) to provide info on the entity such as the last time it was modified

Header Types-Request

- **General: used in request and response messages**

Date	Indicates the date and time of the message origination, e.g. Date: Tue 16 May 2000 11:29:32 GMT
Pragma	Permits to send directives to the recipient, requesting it to behave in a particular way while handling a request or response. Pragma: no-cache informs proxies on the path not to return a cached copy

- **Request header (e.g. to express preferences on the nature of the response, include additional info with the request, to specify a constraint on the server in handling the request)**

Authorization	Includes credentials required to access a resource (e.g. Authorization: Basic YXZpYXRpS29lDizDizM1NA== where Basic is the authentication scheme and YXZpYXRpS29lDizDizM1NA== an encoding of user id and password)
From	The user may include e-mail address for identification (From: gorby@moskvax.com)
If-Modified-Since	Indicates not to return a copy of the resource if it has not been modified after the specified date. Used for caching at a proxy or browser (saves the time needed for downloading of the resource). GET /foo.html HTTP/1.0 If-Modified-Since: Sun, 21 May 200 07:00:25 GMT
Referer	Includes the URL from which the requested URL was obtained. Can be used for locating obsolete links. Referer: http://www.cnn.com
User-Agent	User browser used, client machine OS, etc User-Agent: Mozilla/4.04 [en]C-WorldNet (win95;l)

Header Types

- Entity header (both in request and response messages) to provide info on the entity such as the last time it was modified

Allow	Indicates the list of valid methods that can be applied to a resource PUT /foo.html HTTP/1.0 Allow: HEAD, GET, PUT
Content-Type	Media type of the entity body (e.g. image/gif, text/html etc.)
Content-Encoding	Indicates how the resource representation can be decoded into the format indicated in the Content-Type field (e.g. if the file has been compressed with gzip) Content-Encoding:g-zip
Content-Length	Length of the entity body (bytes). Allow to check whether all the body was received
Expires	The entity should be considered stale after the time specified in this header (after this time the content could still be stored in cache but should not be returned to users without first validating with the origin server)
Last-Modified	Specify the time at which the resource was modified last

Un esempio...

□ Esempio: di richiesta oggetto

HTTP è testuale (ASCII)

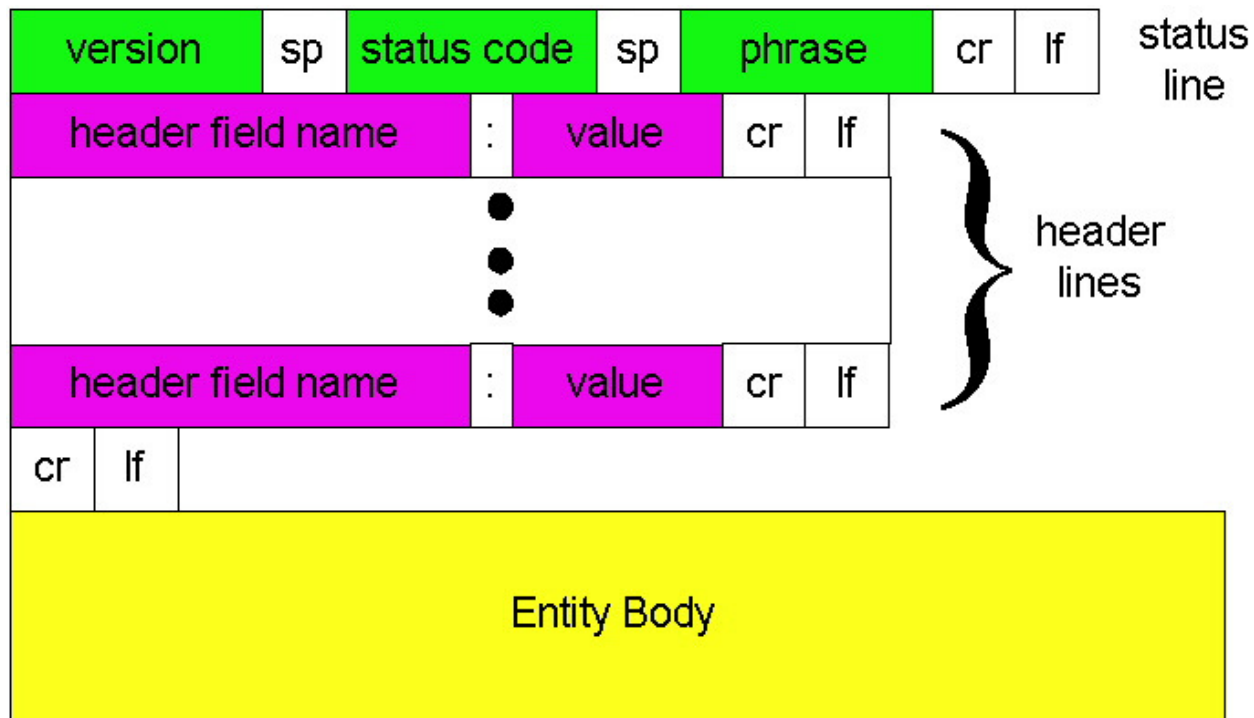
```
GET /ntw/index.html HTTP/1.0
Date: Wed, 22 Mar 2000 09:09:01 GMT
Pragma: No-cache
From: gorby@moskvax.com
User-agent: Mozilla/4.3
```

□ Esempio: risposta

```
HTTP/1.0 200 OK
Date: Wed, 22 Mar 2000 09:10:01 GMT
Server: Apache/1.3.0 (Unix)
Content-Length: 6821
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Type: text/html
data data data data data ...
```

Message

Response



Header Types-Response

- Response header: to provide additional info about the response or to request additional info from the user

Location	Used to redirect the request to where the resource can be found Location: http://www.foo.com/level1/twosdown/Location.html
Server	Origin server SW version number and configuration related info Server: Apache/1.2.6 Red Hat
WWW-Authenticate	Request for etransmit the request with appropriate credentials according to a given scheme

Messaggi

■ 1xx Informational

Status codes:

■ 2xx Success

200 OK:	La richiesta ha avuto successo; l'informazione è inclusa
201 Created:	La risorsa e' stata creata con successo in seguito ad UN POST (se non puo' essere creata subito 202 Aaccepted)
202 Accepted:	The request has been received but has not yet been handled in full (e.g., since a program must run which requires a long time → the user agent can continue with its task without waiting for the action to complete at the origin server)
204 No content:	Successfully recived, no change required in what the user actually sees

Messaggi

■ **3xx Redirection**

301 Moved Permanently:	L'oggetto è stato spostato nell'URL indicato
302 Moved Temporalily:	L'oggetto è stato spostato nell'URL indicato
304 Not Modified:	L'oggetto non modificato dal tempo incluso nella richiesta

■ **4xx Client error**

400 Bad Request:	errore generico (sintassi errata(irriconoscibile))
401 Unauthorized:	Autenticazione fallita (e.g., Accesso senza necessari account e password)
403 Forbidden:	La richiesta e' stata ricevuta e compresa ma il server ha stabilito di non servirla. Le ragioni possono essere indicate nell'entity body.
404 Not Found:	l'oggetto non esiste sul server

■ **5xx Server error**

500 Internal server error	Generico. Errore o guasto nel server
501 Not implemented	Funzione non implementata
503 Service unavailable	Servizio non disponibile

Un esempio

□ Esempio: di richiesta oggetto

```
PUT /motd HTTP/1.0
Date: Wed, 22 Mar 2000 08: 10:07 GMT
From: gorby@moskvax.com
User-agent: Mozilla/4.0
Content-length:23
Allow:GET,HEAD,PUT
data data data data data ...
```

HTTP è testuale (ASCII)

□ Esempio: risposta

```
HTTP/1.1 200 OK
Date: Wed, 22 Mar 2000 09: 10:07 GMT
Server: Apache/1.3.0 (Unix)
...
```

Cache locale: get condizionato

- E' possibile evitare di scaricare oggetti memorizzati nella memoria locale se non sono stati modificati

Client:

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
If-modified-since: Mon, 22 Jun 1998 09:23:24
```

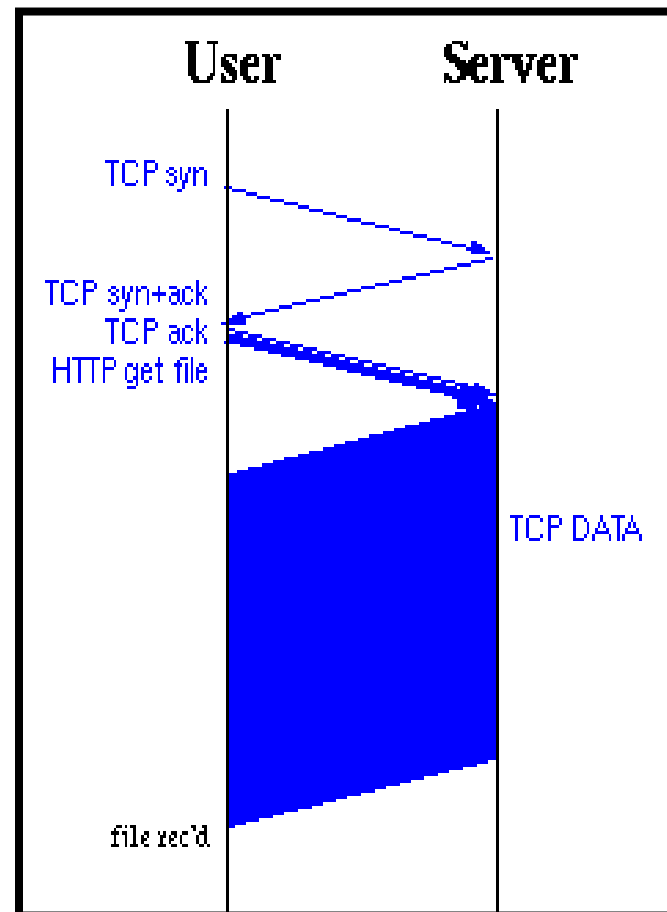
Server:

```
HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
(empty entity body)
```

- E' possibile anche usare il metodo HEAD

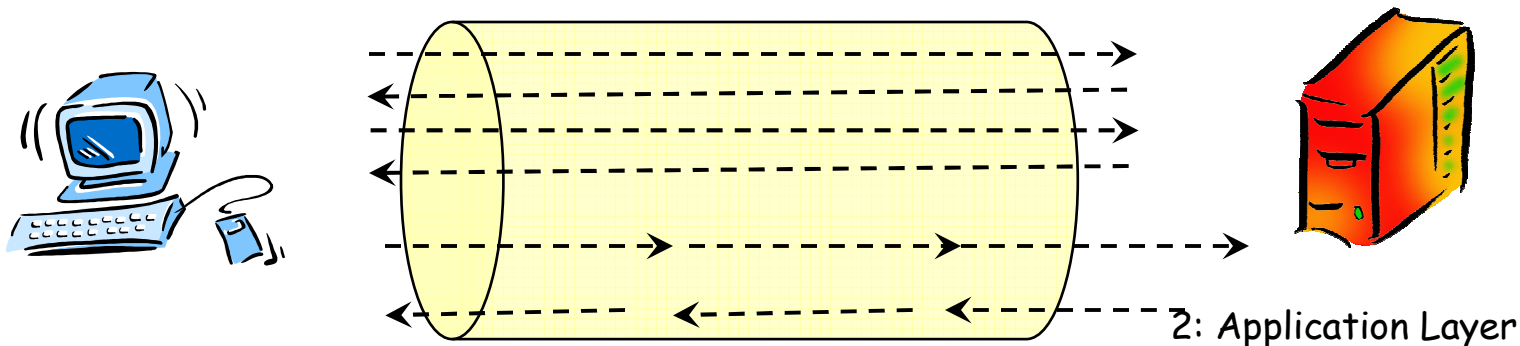
Performance drawbacks

- Mandatory roundtrips
 - TCP three-way handshake
 - get request, data return
 - new connections for each inlined image (parallelize)



Persistent connection (HTTP v1.1)

- ❑ Nel caso *persistent* il server non chiude la connessione dopo l'invio dell'oggetto
- ❑ La connessione rimane aperta e può essere usata per trasferire altri oggetti della stessa pagina web o anche più pagine
- ❑ I server chiudono di solito la connessione sulla base di un *time-out*
- ❑ ***without pipelining***: il client invia una nuova richiesta solo dopo aver ricevuto la risposta per la precedente
 - ***with pipelining***: più richieste vengono inviate consecutivamente dal client



Persistent HTTP

Nonpersistent HTTP issues:

- ❑ requires 2 RTTs per object
- ❑ OS must work and allocate host resources for each TCP connection
- ❑ but browsers often open parallel TCP connections to fetch referenced objects

Persistent HTTP

- ❑ server leaves connection open after sending response
- ❑ subsequent HTTP messages between same client/server are sent over connection

Persistent without pipelining:

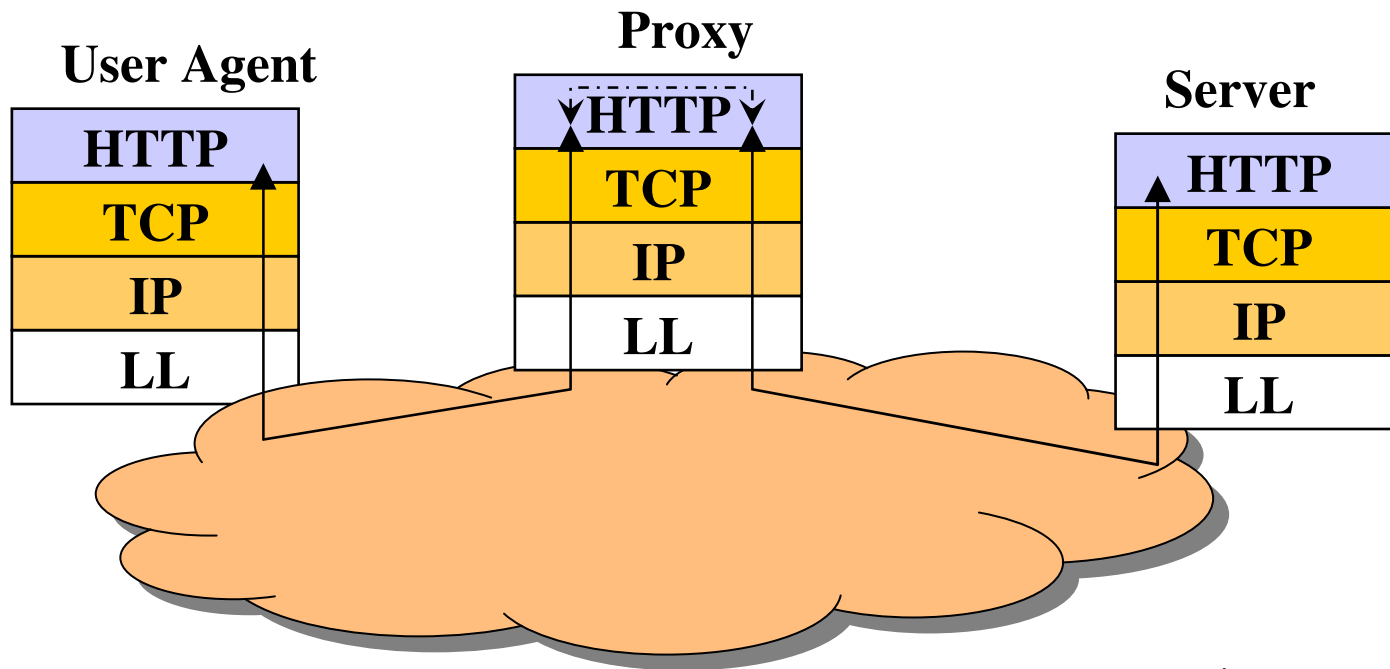
- ❑ client issues new request only when previous response has been received
- ❑ one RTT for each referenced object

Persistent with pipelining:

- ❑ default in HTTP/1.1
- ❑ client sends requests as soon as it encounters a referenced object
- ❑ as little as one RTT for all the referenced objects

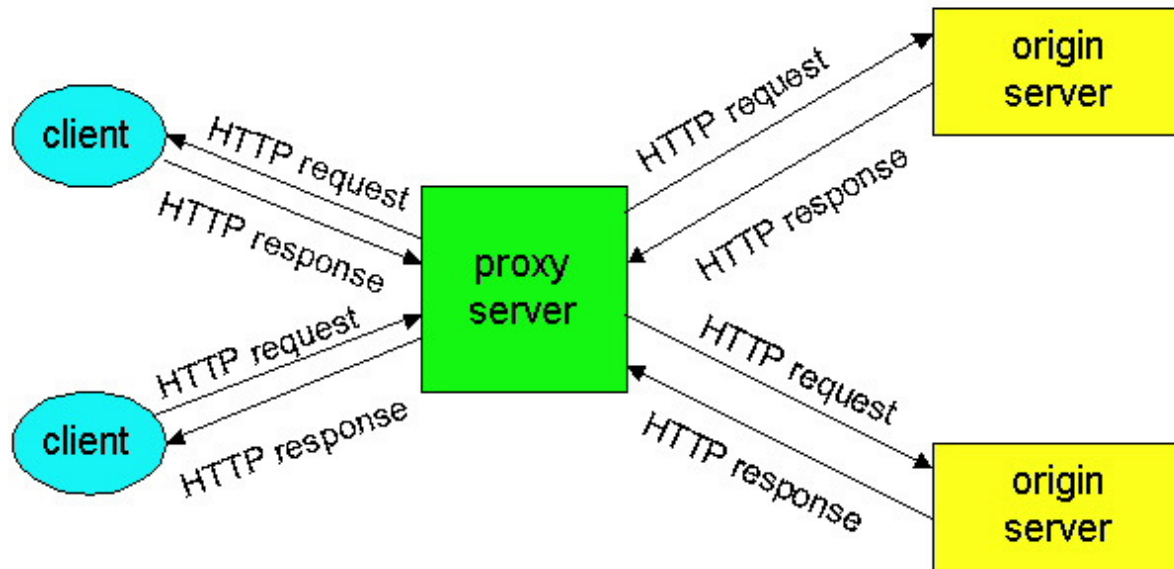
Proxy *An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients.*

- ❑ I proxy sono degli instradatori di messaggi di livello applicativo
- ❑ Devono essere sia client che server
- ❑ Il server vede arrivare tutte le richieste dal proxy (mascheramento degli utenti del proxy)



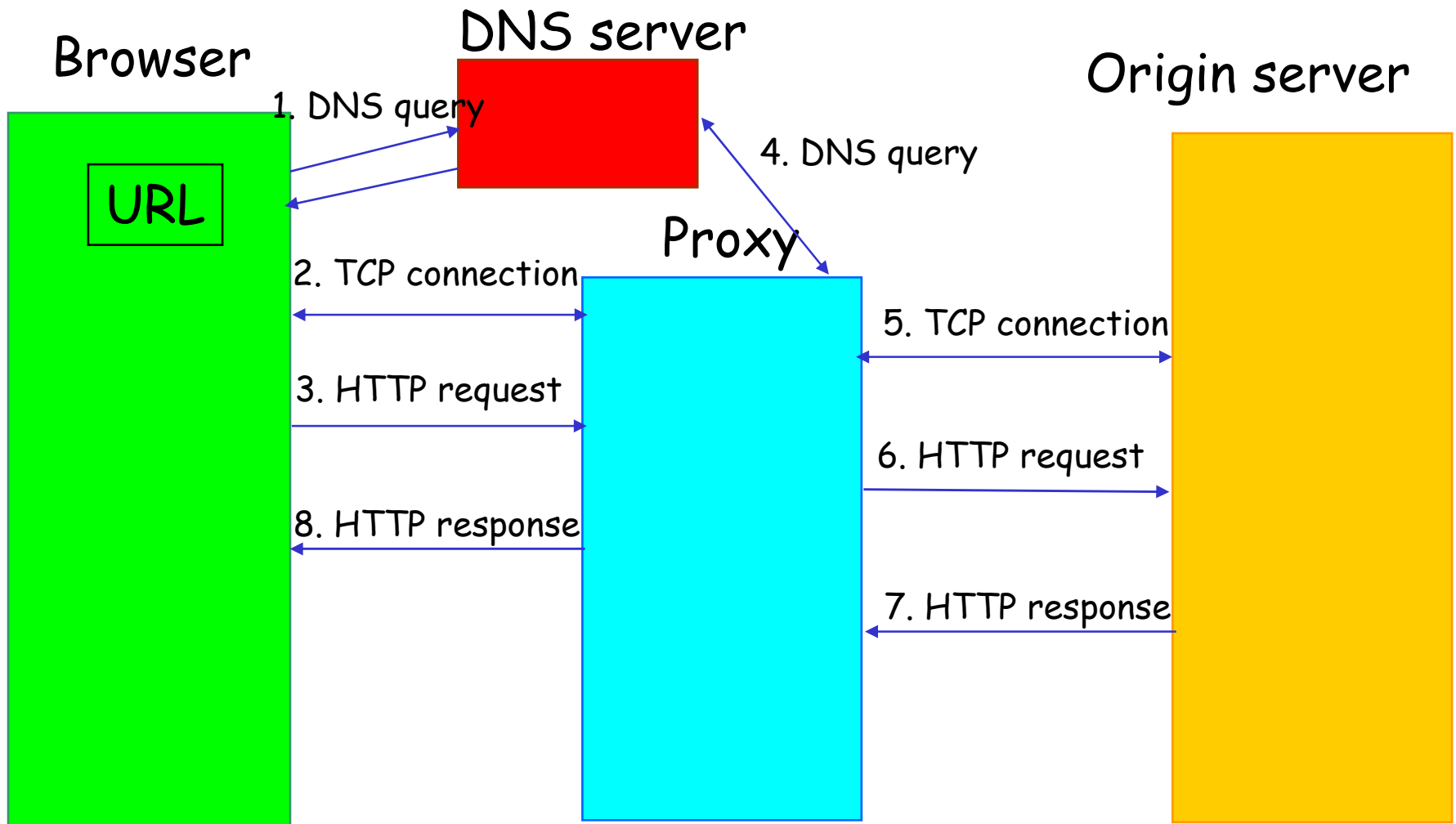
Cache di rete: uso dei proxy

- Compito principale dei proxy è fornire una grande memoria di cache
- Se un documento è contenuto nella cache viene scaricato più velocemente sul client
- Razionale: i server forniscono le stesse info o info fortemente ridondanti a comunita' locali di utenti



- Consistency: cached pages might not be the most recent...
- Ad clickthrough counts: how does Yahoo know how many times you accessed their pages, or *more importantly*, their ads?

Download attraverso un proxy...



Tipi di proxy

- ❑ Regular proxies
 - forwards requests and responses
- ❑ Caching proxy
 - has a cache of responses received in the past
 - when the proxy receives a request that can be satisfied by the cache the request message is not forwarded and the response is returned directly by the proxy
- ❑ Forwarding behavior
 - Transparent proxy
 - Only superficially changes the request (e.g., adding its ID)
 - Non transparent proxy
 - Can modify request/response
 - Anonymization
 - Media type transformation (e.g. changing image format to reduce size)
 - Language transformation ...
 - Entrambi i transparent/non-transparent proxy possono avere associata una cache

Uso dei Proxy(1)

- Caching: *'storage of a response obtained earlier for later use, when clients request the same resource'*. Se la pagina richiesta e' nella cache il proxy puo' inviarla immediatamente al client purché sia fresh (ovvero non sia stata modificata rispetto alla copia in cache)
 - latency minore MA cache consistency importante. Strong consistency/weak consistency
- Agisce come front-end per un gruppo di utenti che condividono l'accesso al web (caching ma anche suddivisione delle risorse—pro: se piu' utenti richiedono la stessa risorsa ad un server un'unica connessione deve essere stabilita; -- con: possibili ritardi dovuti al fatto che piu' utenti condividono le stesse risorse)

Uso dei Proxy(2)

- ❑ Mascheramento degli utenti del proxy: Il server vede arrivare tutte le richieste dal proxy e quindi non e' in grado di riconoscere l'utente. Il proxy conosce le informazioni di accesso relative agli utenti ma e' considerata un'entita' trusted.
 - ❑ Alcuni proxy non permettono l'anonimizzazione e indicano nell'header informazioni sul client
 - ❑ Mascheramento non completo: e.g., user agent e cookies
- ❑ Customizzazione delle risposte
- ❑ Filtraggio delle richieste e delle risposte
 - per limitare l'accesso a certi siti
 - filtrare richieste a siti non adatti ai minori
 - verificare la presenza di virus prima di inviare al client le risposte
 - togliere alcune informazioni dall'header prima di inviare la richiesta (es. e-mail dell'utente)

HTTP operation with intermediaries

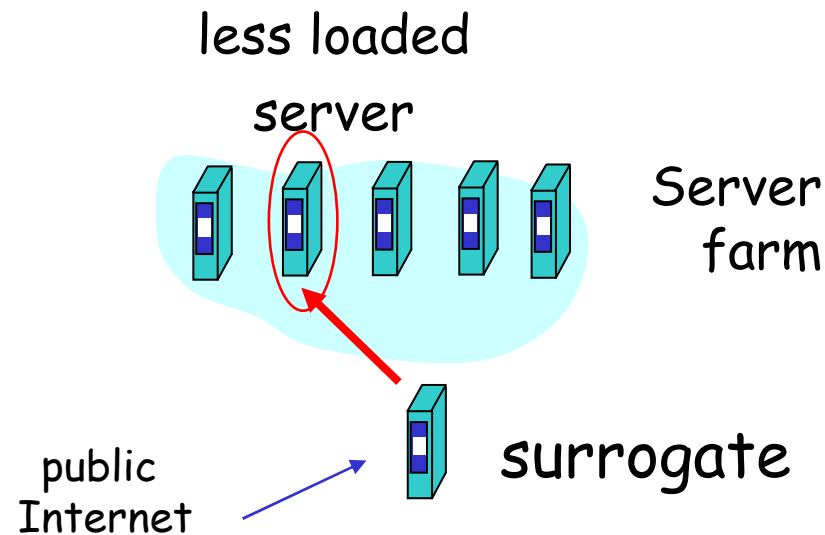
- Three forms of intermediaries:
 - Tunnel: intermediary program acting as a blind relay between two connections. Once active, a tunnel is not considered a party to the HTTP communication. No caching. E.g. in case of encrypted communication
 - Gateway: server acting as intermediary for some other server (typically used for 'speaking' to a non HTTP server). Unlike a proxy, a gateway receives requests *as if it were the origin server for the requested resource*; the requesting client may not be aware that it is communicating with a gateway.
 - HTTP Proxy (speaks HTTP, cache usually associated)

Some rules proxies must obey

- ❑ Semantic neutrality
 - If you can handle the request locally do it, otherwise (also if the proxy cannot understand the request) forward it to the origin server
- ❑ Identification (in HTTP 1..1 all proxies must identify themselves, e.g. to avoid potential loops in the path)
- ❑ Cannot lie on the HTTP version it supports (it can happen the client speaks HTTP 1.0, the proxy HTTP 1.1 or viceversa)
- ❑ Info must be added when returning a resource (e.g. the time since the resource was validated from the origin server).
- ❑ Connections must be maintained operational while the client is ON
- ❑ Must handle the high number of connections with clients and origin servers
- ❑ Proxies passes cookies to the client. They are forbidden to add their cookie-related headers ...

Other types of proxies

- Reverse proxy (surrogates)
 - Proxy placed close to the origin server to reduce the load on them. Load balancing among servers of a server farm



- Interception proxy. Are invisible to the user. All messages pass through the interception proxy that can intercept them, examine the request, generate a response locally or redirect the request to another place (e.g. to the **'best replica'**)