

HTTP

Introducción

- Protocolo de capa de aplicación utilizado para la transferencia de Recursos u objetos.
 - Opera sobre TCP típicamente en el puerto 80
 - Simple
 - Stateless
 - Genérico
 - Utiliza las extensiones MIME.
 - Transporte de páginas HTML, imágenes, datos binarios, o incluso resultados dinámicamente generados (query a una base de datos).
 - Identificación de recursos mediante URL.
- Arquitectura Cliente – Servidor
 - Cliente realiza la petición (Browser)
 - Server responde



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Stateless

- Stateless
 - Cada petición y su correspondiente respuesta da lugar a una nueva conexión TCP entre el servidor y el cliente
 - Descargar una página web requiere de múltiples conexiones.
 - No se lleva registro de las conexiones anteriores.

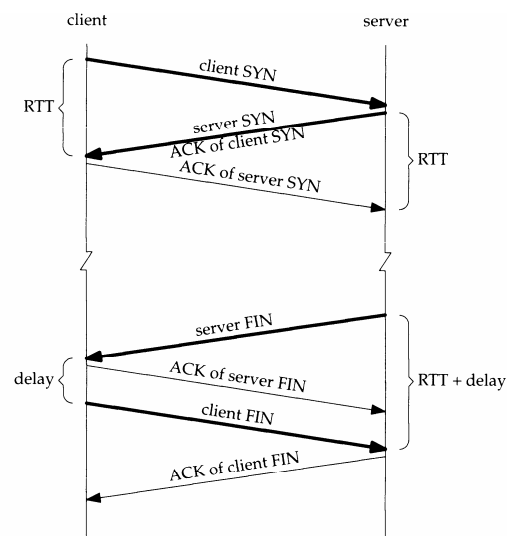


Figure 14.8 TCP three-way handshake and connection termination.



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

URL

- [URL](#) (Uniform Resource Locator)
 - Identificación de recursos
 1. Protocolo
 2. ://
 3. Host (FQDN o IP address)
 4. :
 5. Port Number
 6. /
 7. Path

Ej: `http://www.fi.uba.ar:80/materias/tcom`



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Messages

- Mensajes
 - Una transacción HTTP consiste de un mensaje de petición de cliente y un mensaje de respuesta del servidor.
 - Formato de mensaje HTTP
 - **First line** - describe el tipo de mensaje
 - **Header(s)** - contiene opciones del mensaje
 - **Blank line** - CRLF
 - **Body** – texto o datos binarios (opcional)
 - First line y Headers son siempre leibles, codificados en ASCII



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Messages

- **HTTP Requests**

- request_line
 - <GET|HEAD|POST> <request_URL> <HTTP_version>
Ej.: GET /data/new/index.html HTTP/1.0
- headers
- <linea en blanco>
- body

- **Responses**

- status_line
- headers (0 o más)
- <linea en blanco>
- body



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Headers

Header name	Request?	Response?	Body?
Allow			•
Authorization	•		
Content-Encoding			•
Content-Length			•
Content-Type			•
Date	•	•	
Expires			•
From	•		
If-Modified-Since	•		
Last-Modified			•
Location		•	
MIME-Version	•	•	
Pragma	•	•	
Referer	•		
Server		•	
User-Agent	•		
WWW-Authenticate		•	

Figure 13.3 HTTP header names.



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Ejemplo de transacción

- Requerimiento del siguiente URL
`http://www.somehost.com/path/file.html`
- Se establece la conexión al host **www.somehost.com**, port 80 (Se utiliza el puerto 80 porque no se especifico en el URL). Luego se envia lo siguiente sobre la conexión establecida.

```
GET /path/file.html HTTP/1.0
From: someuser@someserver.com
User-Agent: HTTPTool/1.0
[blank line here]
```



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Ejemplo de transacción

- El servidor debe responder algo similar a esto:

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354
<html>
<body>
<h1>Happy New Millennium!</h1>
(more file contents) . . .
</body>
</html>
```



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Response codes

Response	Description
1yz	Informational. Not currently used.
200	Success.
201	OK, request succeeded.
202	OK, new resource created (POST command).
204	Request accepted but processing not completed.
301	OK, but no content to return.
302	Redirection; further action need be taken by user agent.
304	Requested resource has been assigned a new permanent URL.
400	Requested resource resides temporarily under a different URL.
401	Document has not been modified (conditional GET).
403	Client error.
404	Bad request.
500	Unauthorized; request requires user authentication.
501	Forbidden for unspecified reason.
502	Not found.
503	Server error.
	Internal server error.
	Not implemented.
	Bad gateway; invalid response response from gateway or upstream server.
	Service temporarily unavailable.

Figure 13.4 HTTP 3-digit response codes.



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Methods

- Un HTTP Method es un comando que se ubica en la First line del HTTP request
 - CONNECT,
 - DELETE,
 - **GET**,
 - **HEAD**,
 - OPTIONS,
 - **POST**,
 - PUT,
 - TRACE



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Methods

- GET Method
 - Se utiliza para recibir un recurso requerido a un servidor HTTP mediante un URL.
- POST method
 - Se utiliza para enviar datos al servidor para procesarlos. Las páginas HTML usan POST para enviar datos de formulario
 - Se envían datos en el cuerpo del mensaje. Adicionalmente, se envían headers extra para describir el cuerpo del mensaje como ser **Content-Type:** y **Content-Length:**.
 - El “*request URI*” no es un recurso a ser obtenido sino que es un programa que procesara los datos enviados.
 - La respuesta HTTP es típicamente la salida de un programa y no un archivo estático.
- HTTP HEAD method
 - Se utiliza para obtener información acerca de un recurso sin traer el recurso. La respuesta contiene únicamente el First line y los Header(s).



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP Proxies

- Un proxy *HTTP* es un programa que actúa como intermediario entre el cliente y el server.
 - Recibe los requerimientos del cliente -> Envía ese pedido al servidor.
 - Recibe la respuesta del servidor -> Envía esa respuesta al cliente
 - Los requerimientos a los proxies deben incluir el URL completo, no solamente el path. Por Ejemplo,
`GET http://www.somehost.com/path/file.html HTTP/1.0`
- Seguridad
 - En las redes se permite el acceso HTTP a un único equipo
- Performance
 - Permite el almacenamiento temporal de ciertos recursos, evitando la múltiple transferencia de los mismos
 - Los requerimientos a los proxies deben incluir el URL completo, no solamente el path.



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP

Problemas HTTP 1.0

HTTP/1.0 no tiene en cuenta completamente los siguientes asuntos:

- proxies jerárquicos
- caching
- conexiones persistentes
- hosts virtuales



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP 1.1

Host: Header

- Un servidor HTTP con una dirección IP puede ser *multi-homed*,
 - Se responde sobre múltiples dominios
 - "www.host1.com" y "www.host2.com.ec" pueden vivir en el mismo servidor.
- Cada HTTP request debe especificar el “host name” y el puerto TCP al que el requerimiento está dirigido.
- Ejemplo

```
GET /path/file.html HTTP/1.1
Host: www.host1.com:80
[blank line here]
```



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP 1.1

Conexiones Persistentes

- En HTTP 1.0,
 - Conexiones TCP se cierran luego de cada “request – response”
 - Paginas WEB constan de varios recursos
 - Abrir y cerrar conexiones consume tiempo y recursos como CPU, memoria y ancho de banda
- En HTTP 1.1,
 - Se mantienen conexiones persistentes
 - Se envían varios “requests” en serie sobre la misma conexión y se reciben los distintos recursos en el orden que fueron solicitados.
 - el header "**Connection: close**"
 - El cliente lo incluye en el request si considera que el recurso es el ultimo.
 - Si el servidor lo incluye en la respuesta, el cliente deberá abrir otra conexión para transferir mas datos



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES

HTTP 1.1

Manejo de Cache

- If-Modified-Since:
 - Solamente envía el recurso si ha sido modificado desde la fecha indicada
- If-Unmodified-Since
 - Lo contrario
- Respuesta

```
HTTP/1.1 304 Not Modified
Date: Fri, 31 Dec 1999 23:59:59 GMT
[blank line here]
```



Redes de Datos – Ing. Marcelo Utard / Ing. Pablo Ronco

FACULTAD DE INGENIERIA
UNIVERSIDAD DE BUENOS AIRES