Introducción

Dos procesos en dos sistemas finales distintos se comunican intercambiando mensajes a traves de una red de computadores. Modelo **cliente-servido**

- Cliente envía peticiones al servidor.
- Servidor recibe las peticiones, las procesa y envía la respuesta. Modelo **Peer to peer**: los dos extremos realizan un servicio y solicitan servicios. Protocolo de nivel de aplicación:
- Definen el formato y el orden de intercambio de los mensajes.
- Acciones en la transmisión o recepción de mensajes.

Web

Es una aplciación/servicio más de Internet, combina cuatro ideas que no eran nuevas:

- Hipertexto: formato de la información que permite moverse de una parte a otra de un documento o entre documentos mediantes conexiones internas entre estos documentos, hiperenlaces o enlaces.
- Identificadores de recursos: permiten localizar un recurso en la red, URL o Uniform Resource Locator, URI o Uniform Resource Identifier)
- Modelo cliente-servidor
- Lenguaje de marcas: caracteres o códigos embebidos en texto que indican estructura, semántica o recomendaciones para su presentación. Componentes:
- **Página Web**: archivo HTML base + objetos
- **Navegador**: agente de usuario para la Web.
- **Servidor Web**: almacena objetos Web direccionables a través de una URL.
- **Protocolo HTTP**: permite comunicarse al servidor y al navegador.

URI

Identificador que permite acceder a un recurso web, por ejemplo, una página web. Estructura:
http://www.udc.es/lista.html?urlmenu=servizos/#Final

- **Esquema**: especifica el protocolo utilizado para acceder al recurso. Por ejemplo, http, ftp, https, ...
- **Parte jerárquica**:
  - Autoridad: nombre del servidor. Puede incluir el número de puerto o información del control del acceso (www.udc.es)
  - Ruta: para acceder al recurso. Similar a los directorio. (/lista.html)
- **Fragmento** (opcional): identifica una subdirección dentro de un recurso (#Final)
- **Consulta** (opcional): información adicional, normalmente variables y sus valores. Por ejemplo, para enviar los campos de un formulario. (?urlmenu=/servizos)

URL vs. URI

Normalmente denominamos URLs a todas las direcciones de recursos en Internet, aunque el URI es más completo que la URL. URL = URI - Fragmento.

HTTP

El Protocolo de Transferencia de Hipertexto, conocido como HTTP (por sus siglas en inglés, HyperText Transfer Protocol), es el protocolo utilizado para la comunicación entre clientes como los navegadores web y servidores en la World Wide Web. Ha sido especificado en diferentes documentos RFC a lo largo del tiempo, entre ellos el RFC 1945 para HTTP/1.0, el RFC 2616 para HTTP/1.1, el RFC 7540 para HTTP/2 y el RFC 9114 para HTTP/3. A lo largo de sus versiones, HTTP ha mantenido una compatibilidad general hacia atrás, aunque HTTP/2 introduce cambios en la forma de transmisión que no son compatibles con versiones anteriores. HTTP utiliza el protocolo TCP como medio de transporte, lo que garantiza una conexión orientada y fiable, asegurando que cada mensaje HTTP enviado por el cliente o el servidor llegue a su destino sin modificaciones. Su funcionamiento se basa en un modelo de solicitud-respuesta en el que el cliente envía una petición HTTP al servidor, y este responde con una respuesta HTTP que puede incluir el recurso solicitado. Es importante señalar que HTTP es un protocolo sin estado, lo que significa que el servidor no conserva información sobre las solicitudes previas del cliente, y cada nueva petición se procesa de manera independiente.

Conexiones no persistentes

HTTP/1.0 usa conexiones no persistentes. Pasos en la petición de una URL

1. El cliente HTTP inicia la conexión TCP con el servidor www.fic.udc.es en el puerto 80.
2. El cliente HTTP envía al servidor el mensaje de petición solicitando el objeto/presentación.
3. El servidor HTTP recibe la petición, busca el objeto, lo encapsula en el mensaje HTTP de respuesta y lo envía.
4. El servidor finaliza la conexión TCP.
5. El cliente HTTP recibe la respuesta y finaliza la conexión TCP.
6. El cliente extrae el archivo del mensaje de repuesta, examina el archivo HTML y encuentra referencias a otros objetos HTML.
7. Para cada objeto, volver al paso 1. Dependiendo del navegador, las nuevas conexioens podrían ser en paralelo. Inconevenientes:

- Se necesita una conexión (buffers, variables, timeouts, ...) para cada objeto solicitado.
- Retardo de dos veces el RTT (Round-Trip Time): establecimiento de conexión + pretición y recepción del objeto. Por defecto, en HTTP/1.1 el servidor HTTP deja abierta la conexión TCP, esperando nuevas peticioens/respuestas.
- Sin pipeline: el cliente sólo envía una nueva petición cunado ha recibido la respuesta previa.
- Con pipeline: el cliente realiza una petición tan pronto encuentra una referencia a un objeto.

HTTP/2 y 3

HTTP/2 - RFC 7540 Se basa en el protocolo SPDY de Google, no cambia el protocolo: métodos, códigos de estado, ... son los mismos. Cambia la manera en la que se envían datos. Mejoras:

- Multiplexación total sobre una conexión TCP: descarga de objetos web asíncronamente. Soluciona el problema head-of-line (HOD) en HTTP/1.1
- Protocolo en formato binario y compresión de cabeceras.
- Server Push: el servidor puede enviar objetos no solicitados por el cliente para almacenar en caché. HTTP/3 - RFC 9114 Utiliza el protocolo QUIC de Google (Quick UDP Internet Connections), un

protoclo de código abierto basado en UDP. Mejoras: mantiene multiplexación total, mejor latencia, control de flujo por stream e incluye Transport Layer Security, TLS, 1.3.

Petición HTTP

Línea de petición + línea en blanco: obligatorio Línea de petición: Método URL HTTP/Versión Método:

- GET: utilizado cuando el navegador solicita un objeto.
- HEAD: el servidor responde con un mensaje HTTP, pero sin incluir el objeto solicitado, sólo la cabecera.
- POST: incluye datos en el cuerpo de entidad, frente al GET que los codifica en la URL.
- PUT: permite a un usuario cargar un objeto en la ruta especificada.
- DELETE: permite borrar un objeto de un servidor Web. URL: objeto que se hace referencia Versión Host: especifica el host en el que reside el objeto User-agent: especifica el tipo de navegador que está haciendo la petición POST: utilizado comúnmente cuando un usuario rellena un formulario, el cuerpo de entidad contiene los datos introducidos por el usuario. GET: tambiñen soporta el envío de datos introducidos por el usuario. Se envían codificados en la URL real.

Respuesta HTTP

Línea de estado, compuesta por versión, código de estado + Frase. Códigos de estado agrupados en 5 tipos:

- Informativo: 1xx, por ejemplo, 100 Continue
- Éxito: 2xx, por ejemplo 200 OK
- Redirecciones: 3xx, por ejemplo, 301 Moved Permanently.
- Error del cliente: 4xx, por ejemplo, 404 Not Found.
- Error del servidor: 5xx, por ejemplo 500 Internal Server Error. Date: fecha y hora en la que se creó y envió la respuesta HTTP. Server: especifica el número de bytes del objeto enviado. Last-Modified: indica la fecha y hora en que el objeto fue creado o modificado por última vez. Content-Lenght: indica el número de bytes del objeto enviado. Content-Type: indica el tipo de objeto incluido en el cuerpo de entidad.
- La extensión del archivo no especifica formalmente el tipo de objeto.

Respuesta HTTP: Content-Type

Usa los tipos Multipurpose Internet Mail Extensions, MIME, para definir el tipo de contenido. **MIME**: estándar que indica el tipo de contenido. Gestionado por la IANA. Estructura básica: tipo/subtipo, por ejemplo text/html, image/gif ... Discreto: un único documento de un único tipo

- Aplicación: application/pdf, application/zip, application/octetstream
- audio: audio/mpeg
- image: image/jpeg, image/png, image/gif
- text: text/plain, text/html, text/csv
- video: video/mp4 **Multipart**: encapsula múltiples archivos, posiblemente de distintos tipos, en una única transacción

HTTP: Cookies

HTTP no tiene memoria, se utiliza un mecanismo que permite a un servidor web guardar información en el navegador, conocido como cookies.

HTTP: GET condicional

La utilización de una caché reduce los retardos de recuperación de objetos y reduce el tráfico que circula por la red. Problema: la copia de un objeto en caché puede ser obsoleta. Solución: GET + If-Modificed-Since

- Solo devuelve el objeto si ha sido modificado después de la fecha indicada. Solicitar un objeto por primera vez: `GET /images/udc.git HTTP/1.1 User-agent: Mozilla/4.4` Recibir la respuesta del servidor: `HTTP/1.1 200 OK Date: Sat, 1 Jan 2000 12:00:15 GMT Server: Apache/1.3.0 (Unix) Last-Modified: Fri, 24 Dic 1999 13:03:32 GMT Content-Type: image/gif (datos)...` Pasado un tiempo, se vuelve a solicitar el mismo objeto: `GET /images/udc.git HTTP/1.1 User-agent: Mozilla/4.4 If-modificed-since: Fri, 24 Dic 1999 13:03:32 GMT` Si no se ha modificado, el servidor no envía el objeto de nuevo `HTTP/1.1 304 Not Modified Date: Wed, 5 Jan 2000 20:30:43 GMT Server: Apache/1.3.0 (Unix)`

Correo electrónico

Inventado por Ray Tomlinson en 1971, el correo electrónico es un medio asíncrono de comunicación, compuesto por: lectores de correo o agentes de usuario, servidores de correo y Simple Mail Transfer Protocol, SMTP

SMTP

Definido en el RFC 5321. Permite el intercambio de mensajes entre servidores de correo, el remitente actúa como cliente y el destinatario actúa como servidor. El cliente SMTP establece una conexión TCP con el puerto 25 del servidor SMTP, si el servidor está fuera de servicio se intentará más tarde. Se realiza la sincronización entre emisor y receptor, se indica la dirección de correo electrónico del remitente. El cliente envía el mensaje, este proceso se repite si hay más mensajes para el mismo servidor; y después se cierra la conexión TCP.

SMTP utiliza mensajes en formato ASCII, si el mensaje tiene caracteres no ASCII o binario, tiene que ser codificado. Es un protocolo de oferta, el cliente nvía al servidor, frente a HTTP que es un protocolo de demanda.

Proceso de envío de un mensaje de correo electrónico

1. El usuario, mediante su lector de correo, crea el mensaje, por ejemplo john.doe@udc.es
2. El lector de correo envía el mensaje al servidor de correo del emisor.
3. El servidor de correo, actuando como cliente SMTP, se conecta al servidor de correo del destinario.
4. El cliente SMTP envía el mensaje.
5. El servidor SMTP recibe el mensaje y lo almacena en el buzón del destinatario.
6. El destinatario utiliza su lector de correo para obtener el mensaje.

Formato correo electrónico

Un mensaje de correo electrónico consta de dos partes, cabecera y cuerpo, separadas por una línea en blanco.

- Cabecera: información sobre el corre.
- Cuerpo: el propio correo electrónico. Algunos campos de la cabecera son:
- **From**: sólo una por mensaje
- **To**: una o más por mensaje.
- **Cc** y **Bcc**
- **Subject**: tema del mensaje
- **Date**: fecha y hora en que el mensaje fue enviado.
- **Message-Id**: identificador de cada mensaje, insertado por el ordenador remitente.
- **Recieved**: información sobre el envío del mensaje, como las máuinas por las que pasño el mensaje.
- **Reply-To**: dirección a la que se debe responder.

MIME

Multipurpose Internet Mail Extensions Permite enviar contenidos distintos de texto ASCII en mensajes de correo electrónico, por ejemplo, idiomas con acentos, no lations, sin alfabetos o contenido binario. Sólo afecta a los agentes de usuario, ya que para SMTP es transparente. Campos MIME:

- MIME-Version
- Content-Description: cadena de texto que describe el contenido. Es necesaria para que el destinatario decida si descodificar y leer el mensaje
- Content-Id
- Content-Transfer-Encoding: indica la manera en que está codificado el cuerpo del mensaje.

Protocolos de acceso al correo

Post Office Procolv3, POP3 Protocolo de acceso al correo muy simple. Modo de operación en tres fases:

- Autorización: login y password
- Transacción: recuperar los mensajes, marcar para borrado y estadísticas de correo
- Actualización: cuando finaliza la sesión, el servidor de correo borra los mensajes marcados. Dos configuraciones del cliente POP3:
- Descargar y borrar
- Descargar y guardar Internet Mail Access Protocol, IMAP, permite crear y gestionar buzones remotos en el servidor de correo; IMAP asocia cada mensaje con un buzón, inicialmente al INBOX. Proporciona comandos para crear buzones, mover mensajes , buscar entre sesiones... Dispone de comandos para recuperar componentes de los mensajes.