

Tema 2: Aplicaciones en red

Sesión A4

Clase 4: Principios de las aplicaciones en red

Lectura previa:

- Kurose2010, sección 2.1 excepto 2.1.3.
- http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos
- <http://es.wikipedia.org/wiki/HTML>

Conceptos:

- Arquitectura de las aplicaciones en red [2.1.1]
 - Concepto de arquitectura de aplicación
 - Modelo cliente-servidor : Definición y ejemplos
 - Arquitectura P2P : Definición y ejemplos
 - Auto-escalabilidad de la arquitectura P2P
- Comunicación entre procesos [2.1.2]
 - Procesos cliente y servidor
 - Dentro del modelo cliente-servidor
 - Dentro del modelo P2P
 - Interfaz entre el proceso y la red de computadores: *Sockets*
 - Definición
 - Los sockets como interfaz de programación de aplicaciones (API)
- Servicios de transporte en Internet [2.1.4]
 - Identificación (direccionamiento) de los procesos
 - Insuficiencia de la dirección IP
 - El número de puerto
 - TCP : Características y ejemplos
 - UDP : Características y ejemplos
- Protocolos de nivel de aplicación¹ [2.1.5]
 - Necesidad
 - Aspectos de la comunicación entre aplicaciones reguladas por el protocolo de nivel de aplicación
 - Ejemplos de protocolos de nivel de aplicación [Figura 2.5]
- Servicio de transferencia de información multimedia WWW
- Estructura básica de una página web
- Identificación de los objetos: URL [http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos]
 - Definición
 - Formato
 - URL completa y parcial
 - Ejemplos
- Lenguaje HTML [<http://es.wikipedia.org/wiki/HTML>]
 - Definición
 - Breve descripción
 - Concepto de marcadores de control
 - Hiperenlaces
 - Ejemplo mínimo

1 Ampliación de lo ya visto en la sesión A1.

Sesión A5

Clase 5: La Web y HTTP

Lectura previa:

- [Kurose2010, sección 2.2](#)

Consulta:

- [RFC 2616 apartado 9.3 \(GET\), 9.4 \(HEAD\), 9.5 \(POST\)](#)
- [RFC 2616 apartado 8.1 \(Persistent Connections\)](#)

Conceptos:

- Generalidades sobre HTTP [\[2.2.1\]](#)
 - Cliente HTTP: navegador
 - Servidor HTTP: servidor web
 - Funcionamiento básico del protocolo: petición y respuesta
 - Concepto de protocolo “sin estado”
 - Versiones de HTTP
- Formato de un mensaje HTTP [\[2.2.3\]](#)
 - Estructura general del mensaje: línea inicial, cabeceras y cuerpo
 - Formato de una petición HTTP
 - Línea inicial: línea de petición
 - Peticiones (métodos) más comunes: GET, HEAD y POST [\[RFC 2616 secc. 9.3, 9.4 y 9.5\]](#)
 - Cabeceras más comunes en la petición
 - Cabecera obligatoria *host* (HTTP 1.1)
 - Formato de una respuesta HTTP
 - Línea inicial: línea de estado
 - Códigos de estado
 - Cabeceras más comunes en la respuesta
 - Delimitadores
 - Línea, cabeceras y cuerpo

Actividad 1:

Emplear el programa Telnet para conectar con www.upv.es . Tras cada iteración, analizar las cabeceras obtenidas.

Activar echo local

Probar GET / HTTP/1.0

Probar HEAD / HTTP/1.0. Comparad con el caso anterior.

Probar HEAD /noexist.html HTTP/1.0

Probar GET / HTTP/1.1

Añadir cabecera **host**:

Actividad 2:

Observar las cabeceras empleadas por los navegadores más usuales. Para ello se empleará un programa servidor (netcat, sock, hercules, uno propio en Java...)

Lanzar programa servidor en puerto 8888

Acceder con Firefox a <http://localhost:8888> y comprobar las cabeceras

Repetir con Explorer (Windows) o konkeror (Linux).

Alternativa: Acceder a la web <http://web-sniffer.net> seleccionando allí distintos tipos de User-Agent..

- Conexiones persistentes y no persistentes [\[2.2.2\]](#)

- Conexiones no persistentes
 - Funcionamiento
 - Estimación del tiempo de respuesta
 - Definición del tiempo de ida y vuelta (RTT)
 - Estimación del RTT
 - Problemas con las conexiones no persistentes
 - Conexiones no persistentes en paralelo
- Conexiones persistentes [\[RFC 2616 apartado 8.1\]](#)
 - Funcionamiento
 - Cabecera “**Connection:**”
 - *Pipelining*

Actividad 3:

Demostración de conexiones no persistentes. Emplear Hercules (comprobar si funciona con netcat, y si no con Telnet) para comprobar la persistencia sobre www.redes.upv.es

1. Comprueba que cierra la conexión frente a la petición :

```
GET / HTTP/1.0<cr><lf><cr><lf>
```

2. Comprueba que no la cierra frente

```
GET / HTTP/1.1<cr><lf>host: www.redes.upv.es <cr><lf><cr><lf>
```

3. Comprueba que la cierra frente

```
GET / HTTP/1.1<cr><lf>host: www.redes.upv.es <cr><lf>connection: close<cr><lf><cr><lf>
```

4. Comprueba que no la cierra frente

```
GET / HTTP/1.1<cr><lf>host: www.redes.upv.es <cr><lf>connection: keep-
alive<cr><lf><cr><lf>
```

5. Finalmente, comprueba si permite *pipelining* mediante dos órdenes consecutivas (en la misma línea del Hércules). La mejor forma es generar un log, y luego editarlo

```
GET / HTTP/1.1<cr><lf>host: www.redes.upv.es <cr><lf>connection: keep-
alive<cr><lf><cr><lf> GET / HTTP/1.1<cr><lf>host: www.redes.upv.es <cr><lf>connection:
keep-alive<cr><lf><cr><lf>
```

NOTA: Si el servidor de redes da problemas, probar con www.google.com.

Actividad 4:

Repetir los ejercicios anteriores con www.upv.es, y observar que es el servidor el que finalmente decide si se cierra o no la conexión. No procede el ejercicio 5 de *pipelining*.

- Interacción usuario-servidor: *Cookies* [\[2.2.4\]](#)
 - Necesidad
 - Funcionamiento
 - Cabeceras implicadas
 - Usos más habituales
- Proxy (caché web) [\[2.2.5\]](#)
 - Peticiones a través del proxy
 - Funcionamiento
 - Ventajas
- GET condicional [\[2.2.6\]](#)
 - Caché del cliente
 - Funcionamiento de la petición condicional
 - Cabeceras implicadas

Sesión A6

Clase 6: El correo electrónico

Lectura previa:

- Kurose2010, sección 2.4
- RFC 2045 (MIME) apartado 6.8 (Base-64)
- http://es.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions

Consulta:

- RFC 5321 (SMTP y ESMTP) secciones 4.1.1.1 a 4.1.1.5 y 4.1.1.10
- RFC 1939 (POP3) secciones 1 a 6; 7 (sólo órdenes TOP, USER y PASS), 9 y 10

Conceptos:

- Esquema de funcionamiento del correo electrónico [2.4]
 - Componentes y protocolos implicados
- Las direcciones de correo
 - Formato: buzón y dominio de correo
- Formato de los mensajes de correo [2.4.3]
 - Estructura general del mensaje: cabecera y cuerpo
 - Cabeceras básicas
 - Ejemplos
- *Simple Mail Transfer Protocol* (SMTP) [2.4.1] [RFC 5321 del 4.1.1.1 al 4.1.1.5 y 4.1.1.10]
 - Características básicas
 - Formato de las peticiones SMTP
 - Órdenes básicos del protocolo (HELO, MAIL FROM, RCPT TO, DATA, RSET y QUIT)
 - Formato de las respuestas SMTP
 - Códigos de respuesta
 - Ejemplo de funcionamiento de SMTP
 - Extended SMTP (ESMTP)
 - Nuevas características respecto a SMTP (EHLO)

Actividad 1:

Emplear el programa **Telnet** para conectar con `zoltar.redes.upv.es`, puerto 25 y seguir los pasos:

```
helo redes.upv.es
mail from: profe@disca.upv.es
rcpt to: profesor@zoltar.redes.upv.es
data
From: profe@disca.upv.es
To: profesor@redes.upv.es
Subject: Prueba de correo en la clase de redes

Hola. Esto es una prueba de correo
Para acabar el texto, añadir una línea con un punto.
.
quit
```

Actividad 2:

Emplear el programa **Telnet** para conectar con `zoltar.redes.upv.es`, puerto 110 y seguir los pasos:

```
user profesor
pass secreto
stat
list
retr 1
retr 2
dele 1
list
rset
list
stat
quit
```

- Protocolos de acceso al correo electrónico [2.4.4]
 - *Post Office Protocol (POP3)*
 - Características y funcionamiento
 - Fases de una sesión POP3
 - Órdenes básicas del protocolo POP3 (USER, PASS, QUIT, STAT, LIST, RETR, TOP, DELE, RSET, QUIT)
 - Formato de las respuestas del servidor POP3
 - Ejemplo de sesión POP3
 - *Internet Message Access Protocol*
 - Limitaciones del modelo de funcionamiento POP3
 - Modelo de funcionamiento IMAP
 - Correo electrónico web
- *Multipurpose Internet Mail Extensions (MIME)*
[\[http://es.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions\]](http://es.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions)
 - Necesidad y funcionamiento
 - Cabeceras relacionadas
 - Tipos de contenidos MIME²
 - Codificación Base-64 [RFC 2045 secc. 6.8]

2 De los subtipos “*Multipart*” únicamente: *Mixed* y *Alternative*

Sesión A7

Clase 7:

DNS: servicio de directorio de Internet

Lectura previa:

- [Kurose2010, sección 2.5](#)

Consulta:

- <http://www.root-servers.org/>
- <http://www.nic.es/>

Conceptos:

- Necesidad y objetivo del servicio de nombres de dominio [2.5]
- Sintaxis de un nombre de dominio [2.5.1]
 - Dominios de primer nivel (TLD)
 - Genéricos y geográficos
- Servidores de nombres [2.5.2]
 - Organización: jerarquía
 - Servidores raíz [<http://www.root-servers.org/>]
 - Servidores TLD
 - Servidores autorizados
 - Secuencia básica de resolución de un nombre de dominio
 - Servidor local
 - Consultas recursivas e iterativas
 - Caché de nombres
 - Definición y funcionamiento
 - Respuestas “no authoritative”
- Registros y mensajes DNS [2.5.3]
 - Registros de recursos
 - Formato
 - Tipo de registros: A, AAAA, NS, CNAME, MX y PTR
 - Mensajes DNS:
 - Formato de las consultas y las respuestas
- Inserción de registros en la base de datos DNS
 - Agente registrador [<http://www.nic.es/>]

Actividad 1:

Emplear el programa **nslookup** en modo depuración (`set debug`) para realizar las siguientes accesos DNS

1. Tipo A, `zoltar.redes.upv.es`

2. Tipo A, ftp.rediris.es

Comparar los registros de autorización.

3. Cambiar el servidor a `sun.rediris.es` (`server sun.rediris.es`) y repetir 1. y 2. Evaluar los registros de autorización.

4. Volver a emplear el DNS por defecto (`server x.x.x.x`). Seleccionar modo no recursivo (`set norecurse`).

5. Resolver www.toyota.jp
6. Cambiar server y volver a 6 hasta encontrar al servidor autorizado.
7. Volver a emplear DNS por defecto.

Actividad 2:

8. Emplear el programa **nslookup** en modo depuración 2 (set d2) para realizar las siguientes accesos DNS
 1. Cambiar el tipo a NS.
 2. Buscar servidores de nombres de upv.es, amazon.com, app.net y gmail.com.
 3. Observar registros adicionales (A).
 4. Cambiar tipo a MX.
 5. Buscar servidores de correo de upv.es, amazon.com, app.net y gmail.com. Comparar con ejercicio 2.
 6. Cambiar el tipo a CNAME.
 7. Preguntar por www.redes.upv.es, www.disca.upv.es, zoltar.redes.upv.es y www.upv.es. Explicar por que los servidores suelen ser alias.
 8. Cambiar tipo a PTR.
 9. Resolver 23.4.42.158.in-addr.arpa.

Ejercicios

Ejercicio 1:

Cuando enviamos la orden **GET /muestra.htm HTTP/1.0** seguida de una línea en blanco a un servidor HTTP, obtenemos la respuesta mostrada más abajo (la numeración no forma parte de la respuesta, ha sido añadida para mayor claridad). Acerca de esta respuesta, contesta las siguientes preguntas:

- Señala las partes que la componen, indicando el rango de líneas que compone cada parte.
- Indica qué líneas son obligatorias y cuáles opcionales.
- A la vista de la petición y la respuesta, ¿quién y cuándo iniciará el cierre de la conexión entre el cliente y el servidor? Justifica la respuesta.

```
1. HTTP/1.0 404 Not Found
2. Date: Fri, 27 Mar 2009 07:00:01 GMT
3. Server: Apache/1.3.26 (Unix) ApacheJServ/1.1 mod_ssl/2.8.10
4. Content-Type: text/html
5.
6. <html>
7. <head><title> ERROR </title></head>
8. <body>
9. No se encontró el documento [404]
10. <P> Puede que la página solicitada ya no exista.
11. <P> Compruebe si escribió correctamente la dirección.
12. </body>
13. </html>
```

Ejercicio 2:

Sobre el protocolo HTTP:

- Indica la URL completa que habrá de introducirse en un navegador para acceder al documento "index.html" del servidor HTTP que esta instalado en el puerto 7070 del computador "www.servidor.com".
- Describe la **petición** HTTP que generará el navegador si está configurado para usar la versión 1.1 de HTTP. Utiliza únicamente las cabeceras imprescindibles.
- Describe la **respuesta** HTTP (excepto el cuerpo) que generará el servidor si el documento solicitado está disponible. La respuesta debe incluir las cabeceras necesarias para indicar el formato del documento (*text/html*) y su longitud (335 bytes).
Suponiendo que el documento "index.html" contiene dos imágenes asociadas...
- Si se empleara el protocolo HTTP versión 1.0, ¿Cuántas conexiones TCP sería necesario establecer entre cliente y servidor para descargar la página web completa? Quién inicia el establecimiento y quién el cierre de estas conexiones?
- Supongamos ahora que la conexión se realiza a través de un proxy. Las imágenes están contenidas en la caché del proxy, pero no el documento HTML. ¿Cuántas conexiones TCP será necesario establecer entre cliente y proxy para descargar la página web completa? ¿Y entre el proxy y el servidor? (En ambos casos se emplea el protocolo HTTP versión 1.0).

Ejercicio 3:

Iniciamos nuestro navegador y cargamos la página www.upv.es, que contiene un documento HTML y 2 gráficos (3 objetos en total). Se supone que se conoce la dirección IP asociada a dicho URL, y además todas las imágenes están almacenadas en el mismo servidor web que la página HTML. Indica el tiempo en unidades de RTT (Round Trip Time) que tarda en cargarse dicha página según se empleen:

- a) Conexiones HTTP no persistentes y secuenciales.
- b) Conexiones HTTP no persistentes y paralelas.
- c) Conexiones HTTP persistentes sin pipelining.
- d) Conexiones HTTP persistentes con pipelining.

El tiempo de transmisión de los objetos se supone despreciable en relación al RTT. Además, cada objeto cabe en un segmento.

Ejercicio 4:

Tras conectarte a un servidor SMTP mediante **Telnet**, describe la secuencia de órdenes necesaria para enviar un mensaje a la dirección redes@disca.upv.es con el tema: "ejercicio de redes" y una sola línea de datos en el cuerpo del mensaje. El remitente del correo es alumno@disca.upv.es.

Solución:

```
HELO smtp.disca.upv.es
... //respuesta del S
MAIL FROM: <alumno@disca.upv.es>
... //respuesta del S
RCPT TO: <redes@disca.upv.es>
... //respuesta del S
DATA
... //respuesta del S
Subject: ejercicio de redes

esto es una prueba
. // indica el fin del mensaje
... //respuesta del S
QUIT
```

Ejercicio 5:

El cuadro siguiente reproduce la secuencia de órdenes que un cliente ha enviado (no se muestran las respuestas del servidor) tras conectarse al puerto 25 de un servidor SMTP. Indica qué errores contiene esta sesión SMTP.

```
HELO lab1.poipoi.es
USER adolfo
PASS secreto
MAIL FROM: <adolfo@poipoi.es>
RCPT TO: <lamonda@asdase.es>
DATA
Esto es un mensaje de prueba
Adios.
EXIT
```

Ejercicio 6:

El usuario alumno@etsinf.upv.es desea enviar un correo electrónico a destino@gmail.com a través de su correo web, ofrecido por el servidor www.mail.upv.es. Sabiendo que el servidor de correo predeterminado de la UPV es mail.upv.es, y suponiendo que el usuario destino recibirá el correo en su teléfono móvil a través de POP3 (dejando una copia en el servidor), contesta a las siguientes preguntas:

1. ¿Sobre qué protocolos de comunicación se transfiere el mensaje? Para cada transmisión del mismo, indica el host origen, destino y protocolo empleado. Si es necesario, puedes construir el nombre de hosts como *servicio.dominio* (por ejemplo, *mail.gmail.com* sería el servidor de correo predeterminado del dominio *gmail.com*).
2. Para cada protocolo del apartado anterior, indica una secuencia de órdenes válidas (desde el lado del cliente) que conduzcan a la transferencia del correo. En caso de que el mismo protocolo se emplee en más de una ocasión, cita en la segunda transferencia únicamente las diferencias con respecto a la(s) anterior(es).

Ejercicio 7:

Hacemos una consulta de tipo MX al servidor de nombres preguntando por el dominio *disca.upv.es*. ¿Cómo debemos interpretar la información obtenida? Describe el significado de cada uno de los campos de la respuesta.

```
disca.upv.es MX preference = 10, mail exchanger = smtp1.cc.upv.es
disca.upv.es MX preference = 15, mail exchanger = smtp2.cc.upv.es
disca.upv.es MX preference = 25, mail exchanger = mail.rediris.es
smtp1.cc.upv.es internet address = 158.42.250.32
smtp2.cc.upv.es internet address = 158.42.250.31
mail.rediris.es internet address = 130.206.1.2
```

Ejercicio 8:

Mediante la orden `nslookup` se realiza una consulta al servidor de nombres local (158.42.1.5), obteniéndose el resultado que se muestra a continuación:

```
c:\>nslookup
> set norecurse
> www.uni-bonn.de
Servidor: mirzam.ccc.upv.es
Address: 158.42.1.5
Nombre: www.uni-bonn.de
Served by:
- C.ROOT-SERVERS.NET
192.33.4.12
- D.ROOT-SERVERS.NET
128.8.10.90
. . .

> www.uni-bonn.de 192.33.4.12
Servidor: [192.33.4.12]
Address: 192.33.4.12
Nombre: www.uni-bonn.de
Served by:
- DNS.DENIC.de
194.246.96.79
- DNS2.DE.NET
194.246.96.49
```

1. Explica el significado de la opción `norecurse` que se activa en la línea 2.
2. Explica qué funciones realizan las máquinas `C.ROOT-SERVERS.NET` y `DNS.DENIC.de`.
3. Manteniendo la opción `norecurse`, ¿cuál sería el paso siguiente para averiguar la dirección IP de la máquina www.uni-bonn.de?

Ejercicio 9:

Una empresa decide nombrar su servidor web como “*avionet.com*”, nombre que coincide con su dominio de correo. Además, su servidor de correo SMTP será “*smtp.avionet.com*” y

“dns1.avionet.com”, “dns2.avionet.com” sus servidores de nombres.

1. Indica si la decisión anterior sería correcta. Justifica la respuesta. En caso de que no sea correcta, propón una solución alternativa.
2. Indica los registros DNS que sería necesario incluir en los servidores dns1 y dns2 para poder acceder a los servidores web y SMTP de la empresa. Para cada registro indica el contenido de los campos tipo de registro, nombre y valor.

Ejercicios de diversos protocolos.

Ejercicio 10:

Asocia cada uno de los servicios en red con un protocolo de aplicación en la tabla siguiente. Indica si funcionan sobre UDP, sobre TCP o sobre ambos.

Servicio en red	Protocolos de aplicación	Prot. de transporte
Transferencia de ficheros		
Web		
Correo electrónico		
Traducción de nombres de dominio		
Terminal remoto		

Ejercicio 11:

Indica en qué protocolos de aplicación se emplean cada una de estas órdenes y cual es su significado.

Orden	Protocolo	Significado de la orden
GET		
PORT		
HELO		
STAT		
MAIL FROM		
HEAD		
DATA		
LIST		
RETR		
RCPT TO		

Ejercicio 12:

Desde el computador **rtv1.retevision.es** se desea enviar un mensaje de correo. El remitente es **alumne@retevision.es** y el destinatario **profe@redes.upv.es**. El DNS local de este computador es **dns.retevision.es** y el servidor SMTP del remitente **smtp.retevision.es**. El servidor de correo para el dominio **redes.upv.es** es **smtp.upv.es**. Las cachés DNS de los equipos mencionados están vacías. Indica la secuencia total de peticiones DNS (las respuestas, no) necesaria para que el correo llegue a su destino. Suponemos que los TLD's conocen todos los servidores de nombres autorizados. El servidor local DNS de **smtp.upv.es** es **dns.upv.es**. Muestra el resultado en una tabla de seis columnas: orden (de los mensajes enviados: 1, 2, etc) origen, destino, tipo de consulta, recursiva/iterativa, objeto de la consulta.

Ejercicio 13:

La conexión entre el cliente **cliente.upv.es** (puertos 1025 y superiores) y el servidor web **www.uji.es** (puerto 80) se realiza a través del proxy **proxy.upv.es** (puerto servidor 8080; puerto cliente 5000 y superiores). El navegador instalado en el cliente únicamente soporta la versión 1.0 de HTTP, mientras que la parte cliente del proxy y el servidor final emplean la versión 1.1 de HTTP (sin pipeline y persistente). El cliente inicia la descarga de la página web *index.html* y los objetos asociados *fig1.jpg* y *fig2.jpg*. Supondremos que todos los objetos solicitados existen en el servidor final, y además, el objeto *fig2.jpg* está en la caché del proxy.

- Enumera, en orden cronológico, las conexiones TCP que se establecen.
- Enumera, en orden cronológico, la **línea inicial** de los mensajes HTTP que se intercambian, indicando el número de conexión TCP por el que tiene lugar el intercambio (según la tabla anterior) y el emisor del mensaje.
- Enumera los mensajes DNS intercambiados para hacer posible la interacción HTTP anterior. Supondremos que el cliente y el proxy únicamente conocen la dirección IP del servidor de nombres local y que el servidor de nombres local tienen en su caché las direcciones IP de los servidores TLD necesarios.

Ejercicio 14:

Deseamos obtener una página web, consistente en un documento HTML y cuatro imágenes incrustadas, dos de ellas situadas en otro servidor. ¿Cuántos RTT's son necesarios para ello en cada uno de los siguientes casos?

- Empleando HTTP 1.0 sin conexiones concurrentes.
- Empleando HTTP 1.0 con un máximo de dos conexiones concurrentes.
- Empleando HTTP 1.1 sin conexiones concurrentes y con pipelining.

Considera los tiempos de transmisión despreciables. Expresa de forma gráfica el intercambio de segmentos en cada uno de los casos.