PRÁCTICA 1: PROTOCOLOS DE APLICACIÓN

El objetivo de esta práctica es profundizar en el funcionamiento de las aplicaciones Web y DNS y, en particular, en el diálogo que se mantiene entre los diferentes procesos que componen estas aplicaciones distribuidas. Por ello, es recomendable que recuerdes lo contado en clase respecto a estas dos aplicaciones. Si todavía te quedan dudas sobre el funcionamiento de dichas aplicaciones, puedes consultar otras fuentes como el libro de referencia o las notas del nivel de aplicación disponibles en la página de la asignatura.

La herramienta para analizar los mensajes intercambiados entre cliente y servidor será Wireshark. Después de haber realizado la práctica 0, debes estar familiarizado con su funcionamiento.

Debes responder a las preguntas que se formulan en una hoja de respuestas donde no debes olvidar poner tu nombre en la parte superior. Las respuestas deberán ser entregadas en la siguiente sesión de prácticas.

LA APLICACIÓN WEB

BLOQUE 1. INTERACCIÓN BÁSICA

Comenzaremos explorando el protocolo HTTP descargándonos un fichero HTML sencillo (no contiene referencias a otros objetos dentro de la página como imágenes, etc…) de un servidor web. Haz lo siguiente:

1. Arranca un navegador.
2. Arranca el analizador de protocolos Wireshark tal y como hiciste en la práctica 0 (no comiences las captura de paquetes todavía). Escribe “http” (sólo las letras, sin ‘”’) en el campo de fitrado para que sólo se vean los mensajes del protocolo HTTP en la ventana de captura de paquetes.
3. Espera un minuto y después comienza la captura de paquetes en Wireshark.
4. Introduce la siguiente dirección en el navegador:
   `http://masai.us.es/index.html`
5. Ya puedes parar la captura de paquetes en Wireshark.

La ventana de Wireshark debería ser parecida a la mostrada en la Figura 1.

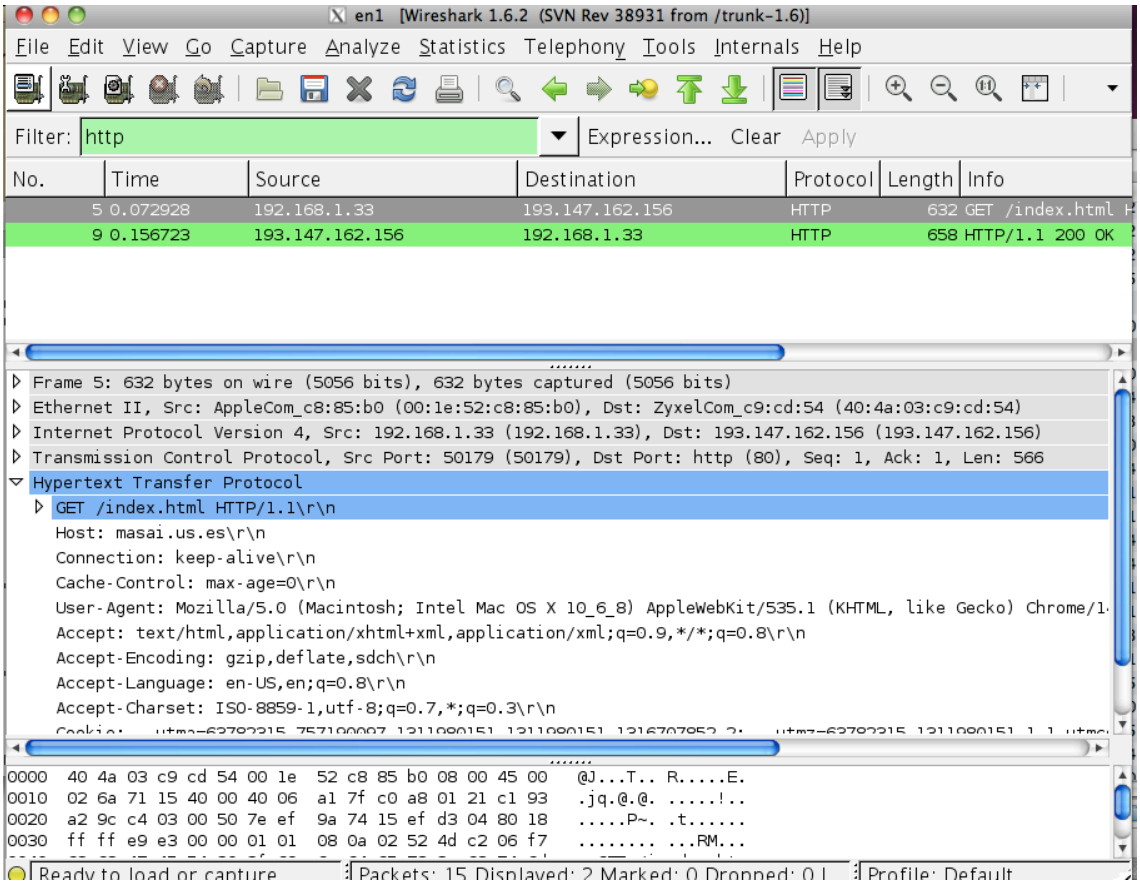

Ilustración 1. Captura de pantalla de Wireshark

En el ejemplo mostrado en la Figura 1, la lista de mensajes HTTP capturados sólo contiene el mensaje `GET` (desde tu navegador hacia el servidor `masai.us.es`) y el mensaje de respuesta desde el servidor hacia tu nagevador. Recuerda que los mensajes del protocolo de aplicación HTTP van encapsulados dentro de segmentos TCP, los cuales a su vez viajan encapsulados dentro de datagramas IP, que a su vez viajan dentro de las tramas Ethernet. Wireshark muestra las cabeceras de todos estos protocolos, pero nosotros ahora estamos interesados sólo en analizar la información del protocolo de aplicación HTTP, por lo tanto, minimizaremos la información mostrada sobre el resto de protocolos.

Puede que en tu captura no aparezcan sólo mensajes idénticos a los mostrados en la Figura 1. Ten en cuenta varias cosas para explicar esto:

(a) Wireshark captura todo el tráfico que lea la tarjeta de red y por lo tanto, en un enlace de acceso múltiple puedes capturar paquetes de otros compañeros (a eso se le llama modo de funcionamiento promiscuo). Para averiguar cuál es la IP de tu ordenador puedes utilizar el comando `ipconfig /all` en Windows o en Linux `/sbin/ifconfig`. Puedes indicar en el filtro de Wireshark que sólo incluya los paquetes con origen o destino en tu ordenador con el filtro " http && ip.addr==193.147.162.146" (donde 193.147.162.146 debe ser reemplazado con la IP de tu ordenador).

(b) Si no obtienes al menos, los dos mensajes anteriores (`GET` y `200 OK`), puede que tu navegador ya haya guardado la página (si lo has intentado anteriormente) en su caché local y ahora utilice la cabecera `conditional-get` para asegurarse de su copia local esta actualizada.

Para solucionar esto, simplemente debes decirle al navegador que borre los datos en su caché.

(c) Puede que el navegador solicite otro `GET` buscando el objeto favicon.ico. Esto es simplemente la referencia a un icono que se mostrará en el navegador al lado de la dirección web. Simplemente ignora este mensaje y su correspondiente respuesta.

(d) Recuerda que puedes decirle a Wireshark que ignore los paquetes que desees. Para ello selecciona un paquete en la ventana donde se listan los paquetes capturados y, con el botón derecho del ratón, puedes indicar que se elimine (ignore) dicho paquete de la lista mostrada.

Si después de 10 minutos no logras tener una captura satisfactoria, puedes descargarte un archivo de capturas de http://masai.us.es/practica1/Bloque1captura

Inspeccionando la información del primer mensaje HTTP `GET` y su correspondiente respuesta, responde a las siguientes preguntas (recuerda que el protocolo se encuentra definido en la RFC 2616). Cuando estés respondiendo deberías tener visibles el contenido de ambos mensajes, e indicar en qué parte del mensaje (campo del mensaje) has encontrado la respuesta a las siguientes preguntas.

1. ¿Tu navegador esta ejecutando la versión del protocolo HTTP 1.0 o 1.1? ¿Qué versión del protocolo HTTP esta ejecutando el servidor?
2. ¿qué idiomas aceptaría el navegador del servidor web?
3. ¿cuál es la dirección IP de tu ordenador?¿cuál es la dirección IP del servidor Web?
4. ¿cuál es el código de estado (status code) indicado en la respuesta del servidor?
5. ¿cuándo ha sido modificado por última vez el objeto `index.html` en el servidor?
6. ¿cuántos bytes de contenido viajan en la respuesta del servidor? (el contenido es campo cuerpo del mensaje, donde viaja el objeto solicitado)
7. ¿qué frases puedes observar en el texto HTML del contenido del mensaje que hayas visto a través del navegador?
8. Observa que en el objeto que viaja como contenido existen marcas del lenguaje HTML. Por ejemplo la marca <H1> indica que lo que se escriba después debe representarse en el navegador con un tamaño de letra apropiado para una cabecera. </H1> indica el final de la marca. ¿Qué otras marcas observas en el objeto?

BOQUE 2. LA INTERACCIÓN GET/RESPUESTA CON CONDITIONAL GET

Además de los proxys o caché web, los navegadores también tienen su caché local donde almacenan los objetos que reciben de los servidores. Antes de continuar esta práctica, asegúrate de borrar la cache del navegador que estés usando. (para hacer esto en Internet Explorer, selecciona Tools->Internet Options ->Delete file, en Firefox selecciona Tools ->Clear Private Data). Ahora haz lo siguiente.

- Arranca el navegador y asegúrate de que borras su caché local.

- Arranca Wireshark (si lo habías cerrado) y comienza una nueva captura de paquetes.
- Introduce la siguiente dirección en el nagevador: http://masai.us.es/index.html . El navegador mostrará la página del ejercicio anterior.
- Rápidamente, haz click en el botón de refrescar la página actual (o reescribe la misma dirección) en el navegador.
- Para la captura de paquetes en WireShark, e introduce "http" en la ventana de filtrado, para que sólo se muestren los mensajes capturados del protocolo HTTP.

Nota: si no eres capaz de realizar la captura satisfactoriamente, puedes descargártela de http://masai.us.es/practica1/Bloque2captura

Responde a las siguientes preguntas:

9. Inspecciona el contenidos de la primera petición HTTP `GET` del navegador. ¿Ves la línea de cabecera "`IF-MODIFIED-SINCE`" en dicha petición?
10. Inspecciona los contenidos de la primera respuesta del servidor (respuesta a la petición anterior). Mira el cuerpo del mensaje. ¿Devuelve el servidor la página web solicitada en su respuesta?
11. Ahora inspecciona el contenido de la segunda petición HTTP `GET` del navegador. ¿Ves una línea de cabecera "`IF-MODIFIED-SINCE`"? ¿qué información viaja como valor de la cabecera?
12. ¿Cuál es el código de estado y la frase devuelta como respuesta en el servidor a la segunda petición HTTP GET? ¿devuelve el servidor explícitamente el objeto solicitado en su respuesta?

BLOQUE 3. PETICIÓN DE DOCUMENTOS CON REFERENCIAS A OTROS OBJETOS

En los ejemplos anteriores, la página web solicitada consiste en un único objeto (un único fichero `html` de pequeño tamaño). Veamos qué ocurre cuando solicitamos un documento de mayor tamaño que a su vez incluye referencias a varios objetos.

Haz lo siguiente.

- Arranca el navegador y asegúrate de que borras su caché local.
- Arranca WireShark (si lo habías cerrado) y comienza una nueva captura de paquetes.
- Introduce la siguiente dirección en el nagevador: http://masai.us.es/research/ . El navegador mostrará una página web que incluye una referencia a varios objetos de tipo imagen.
- Para la captura de paquetes en WireShark y asegúrate de que sólo se muestran los paquetes con información del protocolo HTPP..

Nota: si no has sido capaz de realizar la captura, puedes descargarte una de http://masai.us.es/practica1/Bloque3captura

Ahora responde a las siguientes preguntas:

13. ¿cuántas peticiones HTTP GET han sido realizadas por parte del navegador?¿qué objeto se pedía en cada petición?¿el servidor ha respondido positivamente a todas las peticiones?
14. Cuantas respuestas (cuantas tramas han tenido que ser recibidas) se han recibido con el contenido del objeto principal (documento en html)? (fíjate en la línea justo entre el comienzo del texto de la cabecera HTTP y TCP para ver si la información mostrada ha sido consecuencia del reensamblaje de varios segmentos o no).

BLOQUE 4. APLICACIÓN DNS

Ya debes estar familiarizado con la estructura del Sistema de Nombres de Dominio (DNS) como una base de datos distribuida en la que el cliente DNS realiza peticiones a su servidor DNS local. El servidor DNS local a su vez realiza peticiones a otros servidores DNS que pueden ser los responsables de un nombre de dominio (servidores autoritativos como **us.es**), los de nivel superior del dominio (Top-Level-Domain, como .**es**), o los servidores raiz (root). Una vez resuelta la petición, el servidor DNS local envía al cliente la respuesta, que también guarda en su caché durante un tiempo predeterminado. Podéis ampliar información sobre DNS y su funcionamiento en la norma: http://www.rfc-es.org/rfc/rfc1034-es.txt.

`nslookup` es un probrama cliente DNS accesible a los usuarios. Abre un terminal (ejecuta cmd en Windows o una aplicación terminal en Linux) y prueba lo siguiente.

- `%> nslookup masai.us.es`
- `%> nslookup www.hotmail.es`

Nota: en linux, el comando `host` es equivalente a `nslookup`.

Ahora podéis responder a las siguientes preguntas:

15. En la primera consulta al DNS (la de masai), ¿cuál es la IP del servidor local DNS que responde?¿cuál es la dirección IP de `masai.us.es`?
16. En la segunda consulta DNS, ¿cuál es el nombre canónico del host www.hotmail.es? ¿qué otros nombres puede recibir (alias)?
17. ¿hay un sólo host con una única IP o por el contrario hay mas de uno que puede responder a ese nombre?. Si es que hay mas de uno,¿qué dirección IP tienen los hosts que pueden responder a cualquiera de los nombres asociados a www.hotmail.es?

PARTE OPTIONAL DE LA PRÁCTICA:

BLOQUE 5. TELNET AL PUERTO 80

Telnet es una aplicación que también utiliza TCP para el transporte de sus mensajes. El servidor telnet ejecuta los comandos remotos tecleados enviados por el cliente y devuelve la respuesta resultante de la ejecución de los mismos. Si solicitamos una conexión TCP al puerto 80 de un servidor web mediante telnet, engañaremos al servidor web, que pensará que somos un navegador y, una vez establecida la conexión, podremos escribirle un mensaje HTTP directamente al servidor web.

- Desde un terminal, ejecuta el siguiente comando:
  %> `telnet masai.us.es 80`
- Después escribe el siguiente mensaje de petición HTTP:

```
GET /index.html HTTP/1.1
Host: masai.us.es
```

Pulsa dos veces INTRO al terminar la linea `Host` para enviar el mensaje. Deberías obtener la página solicitada como respuesta.

Intenta ahora volver a conectarte y envíale un método erróneo, por ejemplo:

```
JET /index.html HTTP/1.1
Host: masai.us.es
```

18. ¿qué mensaje recibes como respuesta?

Intenta ahora solicitar un objeto que no existe, por ejemplo:

```
GET /ittex.html HTTP/1.1
Host: masai.us.es
```

19. ¿qué mensaje recibes como respuesta ahora?

BLOQUE 6. CAPTURA DE MENSAJES DNS

Los mensajes DNS de petición y respuesta tienen el mismo formato (la misma sintaxis), mostrada en la figura 2. Recuerda que los RR son cuátuplas del tipo (name, value, type, ttl)

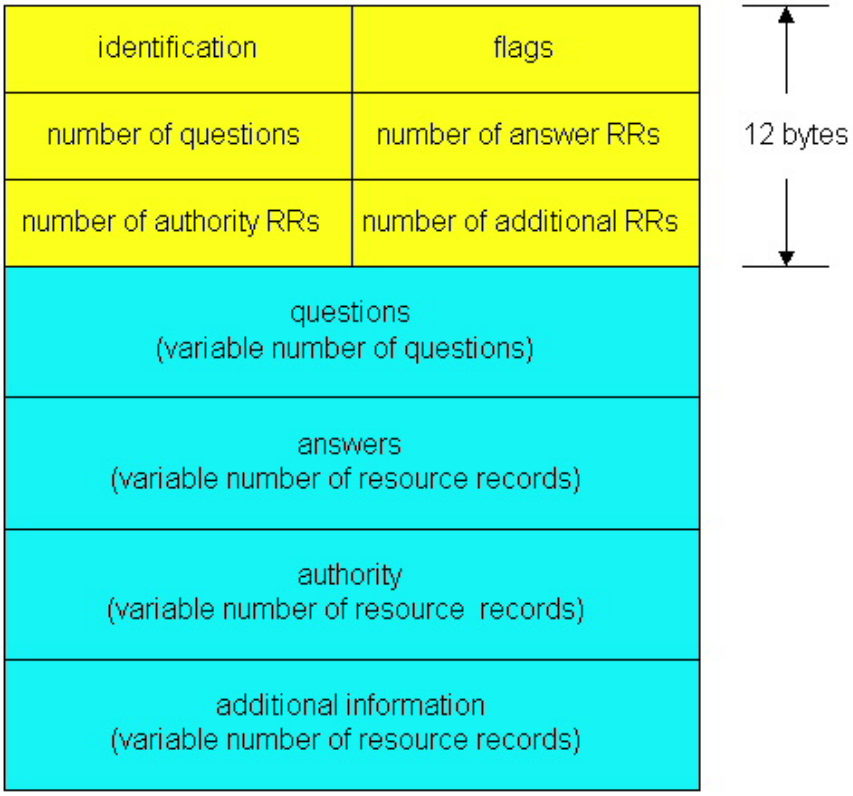

Ilustración 2. Formato genérico de los mensajes del protocolo DNS

donde en los primeros 12B, se incluyen un identificador para poder correlacionar la petición con la respuesta, una serie de bits de bandera (flags) usadas para indicar si el mensaje es de petición o es de respuesta, y cómo se desea que se resuelva la petición (de forma recursiva, etc..). En el resto de campos, se encuentran RRs donde se expresan las consultas (questions), las respuestas a las consultas (answers), RR que apuntan hacia el servidor de una autoridad (authority) e información adicional. Puedes encontrar información detallada sobre el protocolo y los Registros de Recursos (RR) en la norma correspondiente http://www.ietf.org/rfc/rfc1035.txt.

Con el programa ipconfig podréis ver cuál es la IP de los servidores DNS locales y comprobar que os ha respondido uno de los dos. Desde un terminal en windows escribe:

```
%> ipconfig /all
```
(en linux puedes mirar el contenido del fichero /etc/resolv.conf)

Puedes borrar la caché local de tu equipo (con las direcciones ya resueltas que puedes ver con `ipconfig /displaydns`) con el comando `ipconfig /flushdns`.

Ahora podemos capturar los paquetes DNS que se genera el navegador si no conoce la dirección IP del servidor escrito en la URL. Haz lo siguiente:

- Utiliza ipconfig para borrar la cache DNS local de tu equipo.
- Abre un navegador y borra su caché.
- Abre Wireshark e introduce "ip.addr==IP_tu_equipo" donde puedes obtener la IP de tu equipo con ipconfig
- comienza la captura de paquetes en wireshark
- Escribe en el navegador la dirección de la página web: http://www.ietf.org
- Para la captura de paquetes en wireshark.

Examina el contenido de los mensajes del protocolo DNS y responde a las siguientes preguntas:

20. Localiza los mensajes de petición y respuesta del protocolo DNS. ¿son enviados mediante TCP o UDP? ¿cuál es el puerto destino para el mensaje de petición DNS?¿cuál es el puerto origen en el mensaje de respuesta DNS?

21. Examina el mensaje de petición DNS. ¿qué tipo ("type") de consulta DNS es?¿contiene el mensaje de petición DNS alguna respuesta en el campo "answers"?

22. Examina el mensaje de respuesta DNS. ¿cuantas "respuestas" (answers) se ofrecen?¿para qué sirve cada una de esas respuestas?