Protocolos HTTP y TCP/IP

Área de Arquitectura y Tecnología de Computadores – Versión 1.0.915, 24/03/2022

Objetivos

El objetivo de esta sesión es el estudio de los protocolos HTTP, TCP e IP. Los protocolos TCP/IP son la base de todos los servicios que se usan en Internet, como por ejemplo el correo electrónico o la web. El protocolo HTTP es un protocolo de la capa de aplicación que sirve para acceder a páginas web. Durante la sesión se utilizará el analizador de paquetes de red *Wireshark* para analizar el contenido de los paquetes que se transmiten por la red.

Conocimientos y materiales necesarios

Antes de comenzar esta práctica el alumno debe:

- Conocer la organización en capas de los protocolos de red.
- Acudir al laboratorio de prácticas con el libro de apuntes de la asignatura.
- Durante la sesión se plantearán una serie de preguntas que puedes responder en el correspondiente cuestionario en el Campus Virtual. Puedes abrir el cuestionario en otra pestaña del navegador pinchando en el enlace mientras mantienes pulsada la tecla Ctrl.

1. La capa de aplicación. Protocolo HTTP

Vamos a analizar el intercambio de mensajes HTTP en una más alta de la arquitectura de protocolos, la capa de aplicación.

- Abre el archivo `6-peticionHTTPcorta.pcap` que se ha suministrado con el material de la práctica. Lo que ves en el primer panel son las 13 tramas de datos que se han tenido que intercambiar entre la máquina del navegador y la máquina del servidor web para que el navegador reciba el contenido de la página web.
- Utiliza un filtro para mostrar solo los paquetes pertenecientes al protocolo HTTP. Selecciona el primer paquete de la lista filtrada.
- Pulsa en el panel intermedio sobre la línea correspondiente al protocolo HTTP para seleccionar sus bytes y despliega pulsa sobre el símbolo + que hay a la izquierda de **HyperText Trasfer Protocol**. Vemos que el panel inferior no se ha modificado (sigue mostrando en azul los bytes correspondientes al protocolo HTTP) pero en el panel intermedio se nos muestra con más detalle el significado de cada byte. Si vamos pulsando sobre cada una de las líneas que han aparecido, podemos ver en el panel inferior qué bytes son los que proporcionan esa información. Podemos comprobar cómo el navegador web envía bastante información adicional al servidor web además de la ruta del fichero a obtener.
- ¿Qué navegador se ha usado para realizar la petición? Responde en el cuestionario: pregunta 1.
- ¿Qué sistema operativo ejecutaba la máquina del navegador? Responde en el cuestionario: pregunta 2.
- Pulsa ahora sobre la segunda trama del panel superior. En la respuesta del servidor Web? sólo sobre el símbolo + a la izquierda del protocolo HTTP. Como sabrás, el servidor responde con el código **200** que corresponde a una respuesta correcta (**OK**). ¿Con qué bytes se codifica el código 200? Responde en el cuestionario: pregunta 3. Como puedes comprobar, el **200** está codificado en ASCII y no en binario, por que HTTP es un protocolo orientado a texto y no binario de bajo nivel.
- Pulsa el + que hay a la izquierda de la línea denominada **Line-based text data** en el segundo panel. Los bytes que se marcan en azul en el tercer panel corresponden con el archivo HTML enviado por el servidor web al navegador. Puedes ver el archivo tanto en el segundo panel como en el tercero.

Hemos visto que los bytes que corresponden al protocolo HTTP, que es el que usan el navegador y el servidor web para conversar, son sólo una parte de los bytes que componen el total del paquete. El resto de los bytes del paquete son los que se usan para llevar los datos HTTP desde la máquina del navegador hasta la máquina del servidor web.

2. La capa de transporte. Protocolo TCP

Vamos a analizar con detalle TCP, el protocolo de transporte que se ha usado en nuestra captura:

- Pulsa de nuevo sobre el primer paquete que nos muestra *Wireshark* (deberia ser el paquete con la petición `GET`). Si en el panel intermedio hay algún campo expandido, pulsa en los botones ⊟ para compactarlos.
- En el segundo panel, pulsa en cualquier sitio de la línea que pone **Transmission Control Protocol** excepto en el +. Como ejemplo, en el tercer panel se seleccionan de azul los bytes que pertenecen a la cabecera del protocolo TCP. ¿Cuantos bytes contiene? Responde en el cuestionario: pregunta 4. En la figura 1 tienes la estructura de un paquete TCP. Cada fila de esa figura equivale a 32 bits del paquete TCP. De acuerdo con eso, el paquete que tienes seleccionado, ¿contiene campo de opciones y relleno? Responde en el cuestionario: pregunta 5.

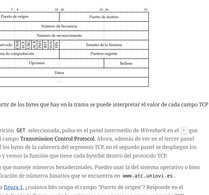

Figura 1. Estructura del paquete TCP

Vamos ahora a comprobar cómo a partir de los bytes que hay en la trama se puede interpretar el valor de cada campo TCP según el esquema de la figura 1.

- Con la trama de la petición `GET` seleccionada, pulsa en el panel intermedio de *Wireshark* en el + que hay a la izquierda del campo **Transmission Control Protocol**. Ahora, además de ver en el tercer panel seleccionados en azul los bytes de la cabecera del segmento TCP, en el segundo panel se despliegan los campos del protocolo y vemos la función que tiene cada byte/bit dentro del protocolo TCP.
- Abre una calculadora que maneje números hexadecimales. Puedes usar la del sistema operativo o bien la aplicación de codificación de números binarios que se encuentra en `www.etc.unioviedo.es`.
- Teniendo en cuenta la figura 1, ¿cuántos bits ocupa el campo "Puerto de origen"? Responde en el cuestionario: pregunta 6. Según la figura, serán los primeros que aparecen en un segmento TCP. Mirando en el tercer panel de *Wireshark*, hallá qué valor, en hexadecimal, tiene el campo "Puerto de origen" y conviértelo a decimal. ¿Qué valor es? Responde en el cuestionario: pregunta 7.
- Pulsa sobre el campo **Source port** en el segundo panel de *Wireshark* y comprueba que has hecho correctamente el apartado anterior. Este puerto se asigna aleatoriamente en cada petición. Se utilizará para dirigir la respuesta a la aplicación que generó la petición.
- Realiza los mismos pasos para el campo "Puerto de destino", ¿Cuál es el valor del puerto de destino en decimal? Responde en el cuestionario: pregunta 8. Por convenio, todos los servidores web utilizan este puerto para recibir las peticiones de los navegadores web; de esta manera, únicamente tenemos que conocer la máquina para localizar al servidor web. Es un *puerto bien conocido*. Comprueba con *Wireshark* que has hecho bien los cálculos. Fíjate que el puerto de origen que *Wireshark* almacenó anteriormente es en el que está esperando el navegador la respuesta. Recuerda que el puerto es un número que sirve para identificar a la aplicación a la que va dirigida la información. En este caso, el puerto 80 identifica al servidor web.
- Mira en la figura 1 cuántos bytes ocupa el campo "Número de secuencia". Mirando en el tercer panel, ¿qué bytes son? Responde en el cuestionario: pregunta 9. Obtén con la calculadora con qué número decimal se corresponden. Comprueba en el segundo panel que has hecho correctamente los pasos anteriores. ¿Coincide el número de secuencia que ha calculado con el que muestra *Wireshark*? ¿Qué valor propone *Wireshark* para este campo? Responde en el cuestionario: pregunta 10. La razón de esta divergencia es que, para facilitar la comprensión del intercambio de datos, *Wireshark* indica los números de secuencia con respecto al primero de la comunicación, que, como se verá más adelante, es el 1749 A004h.
- ¿Para qué sirven los números de secuencia? Responde en el cuestionario: pregunta 11.

Siguiendo el mismo proceso, podrías obtener los valores de cada campo, pero por razones de tiempo vamos a utilizar directamente la información que interpreta *Wireshark*:

- Pasamos ahora al campo "Número de reconocimiento", ¿Qué valor propone *Wireshark* para este campo? Responde en el cuestionario: pregunta 12. Como en el caso anterior, *Wireshark* indica un número relativo y no el valor real.
- ¿Para qué sirven los números de reconocimiento? Responde en el cuestionario: pregunta 13.
- Selecciona el campo **Header length** que es como *Wireshark* denomina al campo "Localización datos". De los bits que *Wireshark* resalta en azul en el tercer panel (50h), solo los cuatro primeros bits (5h) son de este campo; los otros cuatro bits se corresponden con el campo "Reservado". Como puedes comprobar, *Wireshark* indica que la longitud en 20 bytes y no 5. La razón es que lo que indica el campo (5) es el número de bloques de 32 bits (4 bytes) que tiene la cabecera y 5 × 4 = 20.
- A continuación tenemos 8 campos de longitud 1 bit que *Wireshark* denomina "Flags". Según el panel intermedio, ¿qué campos están activos? Responde en el cuestionario: pregunta 14. Según tus apuntes, ¿qué significan estos flags? Responde en el cuestionario: pregunta 15.
- El siguiente campo es "Tamaño de la ventana". Según el panel intermedio, ¿cuál es el tamaño en este caso? Responde en el cuestionario: pregunta 16. ¿Para qué sirve el tamaño de la ventana? Responde en el cuestionario: pregunta 17.

Solo quedan los campos "Suma de comprobación" y "Puntero urgente", ya que hemos llegado a la conclusión de que esta trama TCP no dispone del campo "Opciones", y no los vamos a analizar.

2.1. Establecimiento de conexión en TCP

Borra el contenido del campo "`Filter`" y pulsa el botón `Aceptar`.

Podemos ver que el paquete en el que el navegador envía la petición `GET` al servidor web es el cuarto que se intercambian entre las dos máquinas. Antes de enviar la petición, las capas de transporte de las máquinas del cliente y del servidor han tenido que establecer una conexión. En el libro de teoría tienes explicado con detalle el proceso y es aconsejable que lo repases antes de hacer los siguientes ejercicios.

- Selecciona el primer paquete del primer panel de *Wireshark*. En el segundo panel, expande la información correspondiente a TCP. ¿Qué flags están a 1? Responde en el cuestionario: pregunta 18. Como habrás comprobado al leer el libro de teoría, este es el primer datagrama que se envía al establecer una conexión. Si buscases el valor del campo "Número de secuencia" verías que es 1749A004h. Fíjate que *Wireshark* lo interpreta como el valor 0 para que sea más fácil seguir el intercambio de segmentos TCP y que esta es la razón de que para el paquete del `GET` pusiera el número de secuencia relativo 1.
- Selecciona el segundo paquete en el primer panel. Este paquete lo envía la máquina del servidor web como respuesta a la petición de conexión de la máquina del navegador. ¿Qué flags están a 1 ahora? Responde en el cuestionario: pregunta 19. Comprobamos que, como era de esperar, la máquina acepta la conexión (ACK) y a su vez, intenta establecer una conexión enviando un SYN. Si determinamos el valor del campo "Número de reconocimiento" veras que es F749AD05h. Si que *Wireshark* interpreta como 1. Con este valor el servidor le está diciendo al navegador que ha recibido correctamente el byte 0 de la conexión y que está preparado para recibir el byte 1.
- Si determinases el "Número de secuencia" veras que vale 80C5E38Dh. Como este es el primer paquete que el servidor le envía al navegador, *Wireshark* interpretará ese número como el paquete 0 en ese sentido.
- Selecciona el tercer paquete en el primer panel. ¿Qué flags están a 1? Responde en el cuestionario: pregunta 20. Comprueba que el campo "Número de reconocimiento" se corresponde con el esperado.

Una vez intercambiados estos tres paquetes entre la máquina del cliente y la del servidor, se ha establecido una conexión *"full duplex"* entre ambas máquinas. Esto quiere decir que existen dos conexiones activas:

1. La conexión que "lleva" los datos del cliente al servidor. Esos datos están identificados a partir del número de secuencia que envió el cliente al servidor en el primer paquete.
2. La conexión que "lleva" los datos desde la máquina del servidor a la del cliente. Estos datos están identificados por el número de secuencia que estableció el servidor en el primer paquete que envió al cliente (paquete número 2 del primer panel de *Wireshark*) y que es distinto del que usa el cliente.

También hemos de fijarnos que en cada datagrama que envía el cliente al servidor, además de incluir el número de secuencia de sus datos, incluye también el número de reconocimiento (en el campo "Número de reconocimiento"), que indica el número de secuencia del siguiente byte de datos que el emisor espera recibir. El servidor hace lo propio con cada datagrama que envía al cliente. De esta manera, en cada datagrama que se intercambian, no sólo se identifican los datos que se envían, sino que además se identifican los datos que se han recibido.

De la misma forma que se establecen conexiones, también se terminan. Fíjate en los paquetes número 8, 10 y 11 del paquete 9 en una comunicación con otra máquina que no tiene que ver con los que estamos estudiando aquí. Se han generado al cerrar la página en el navegador. ¿Qué flags están a 1 en los paquetes 8 y 10? Responde en el cuestionario: pregunta 21. ¿Por qué? Si no entiendes el proceso de cierre de una conexión se lo puedes preguntar a tu profesor.

3. La capa de red. El protocolo IP

La capa inmediatamente inferior a la capa de transporte es la capa de red. En la arquitectura de protocolos TCP/IP el protocolo que se encuentra en el nivel de red es *"Internet Protocol"* o IP. Vamos a utilizar *Wireshark* para analizar algunos campos de este protocolo. En la figura 2 tenemos un esquema del contenido de un datagrama IP:

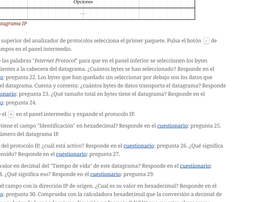

Figura 2. Estructura de un diagrama IP

- En el panel superior del analizador de protocolos selecciona el primer paquete. Pulsa el botón ⊟ de todos los campos en el panel intermedio.
- Pulsa sobre las palabras *"Internet Protocol"* para que en el panel inferior se seleccionen los bytes correspondientes a la cabecera del datagrama. ¿Cuántos bytes se han seleccionado? Responde en el cuestionario: pregunta 22. Los bytes que han quedado sin seleccionar por debajo son los datos que transporta el datagrama. Cuenta y contesta: ¿cuántos bytes de datos transporta el datagrama? Responde en el cuestionario: pregunta 23. ¿Qué tamaño total en bytes tiene el datagrama? Responde en el cuestionario: pregunta 24.
- Pulsa sobre el + en el panel intermedio y expande el protocolo IP.
- ¿Qué valor tiene el campo "Identificación" en hexadecimal? Responde en el cuestionario: pregunta 25. Este es el número del datagrama IP.
- De los flags del protocolo IP, ¿cuál está activo? Responde en el cuestionario: pregunta 26. ¿Qué significa el valor obtenido? Responde en el cuestionario: pregunta 27.
- ¿Cuál es el valor en decimal del "Tiempo de vida" de este datagrama? Responde en el cuestionario: pregunta 28. ¿Qué significa eso? Responde en el cuestionario: pregunta 29.
- Selecciona el campo con la dirección IP de origen. ¿Cual es su valor en hexadecimal? Responde en el cuestionario: pregunta 30. Comprueba con la calculadora hexadecimal que la conversión a decimal de cada uno de sus bytes da lugar a la misma dirección en formato X.Y.Z.W que muestra el programa.
- Como puedes observar, el datagrama también tiene la dirección IP de destino.

4. Visión global

Ahora que ya hemos visto con detalle todas las capas, vamos a recapitular con una visión global. Para esto resulta muy útil el panel superior de *Wireshark*. En él puedes observar cómo la petición de una página web se ha traducido en diez tramas. Las otras tres que hay en la captura son comunicaciones con otras máquinas, como puedes comprobar analizando las direcciones.

Las tres primeras tramas sirvieron para establecer la comunicación. Puedes ver que en el campo info se muestran los datos fundamentales del protocolo: los puertos [1], que están activos, los números de secuencia, el tamaño de la ventana, etc. La cuarta trama es la petición HTTP y la quinta el ACK a esa petición. La cuarta es la respuesta del servidor con la página web y la séptima, el ACK a la trama anterior. Finalmente, como se ha visto en una sección previa, las tramas 8, 10 y 11 realizan la desconexión.

Ahora, ya puedes trabajar con *Wireshark* deberías empezar analizando este panel superior, posiblemente utilizando filtros para ver solo la información de los protocolos o máquinas que te interesen.

5. Ejercicios adicionales

Supongamos que una máquina en la dirección IP: `156.35.122.100` quiere solicitar la página web denominada `index.html` al servidor web cuya dirección IP es `129.168.7.100`.

- Indica sobre la figura 3 el intercambio de paquetes entre estas dos máquinas. Indica con flechas su sentido y etiquetalas con números.
- Describe a continuación cada uno de los paquetes, los protocolos utilizados, y el valor de los campos más significativos usados en dichos protocolos. Indica también para qué se usa cada uno de los protocolos.

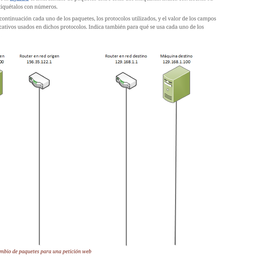

Figura 3. Ejercicio de intercambio de paquetes para una petición web

1. El puerto 1122 que utilizó el Firefox para recibir las respuestas, como se ha explicado, es un puerto escogido al azar. Ese número de puerto está registrado para el programa avalaunt-upg y por eso Wireshark lo indica con ese nombre.