Nombre: ..... Fecha: / 10 /2010

Grupo: 1 ☐ 2 ☐ 3 ☐ 4 ☐

PRÁCTICA 9
INTERNET. CAPA DE TRANSPORTE (TCP, UDP). CAPA DE INTERNET. DNS.

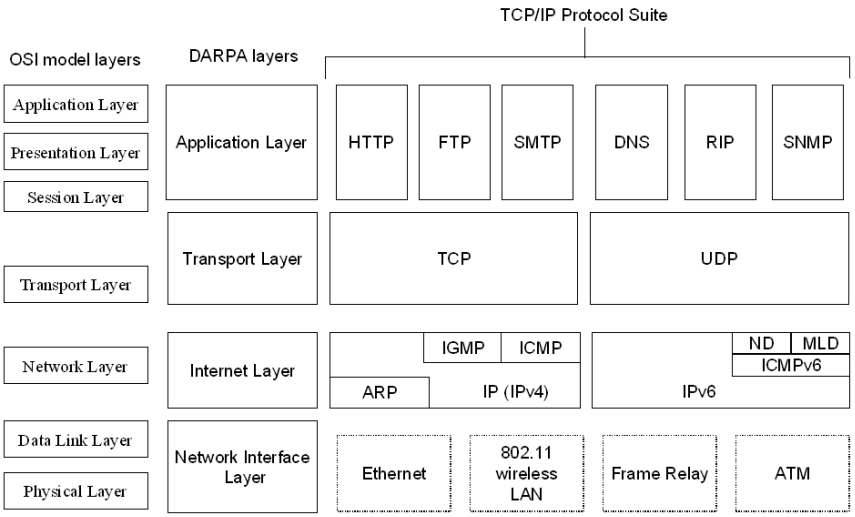
En esta segunda práctica sobre la pila de protocolos TCP/IP nos vamos a ocupar de lo que se conoce como capas de transporte y de red o Internet. Puedes observar en la imagen inferior la posición que ocupan dentro de la pila de protocolos TCP/IP.

La capa de transporte se encarga, entre otras cosas, de empaquetar la información en paquetes de tamaño adecuado (limitarlo, por ejemplo, a las características de la red física, que podría no soportar paquetes de tamaños superiores a…), de verificar que todos los paquetes de una comunicación han llegado a destino (dependiendo del protocolo que usemos), de comprobar que los paquetes no estén corruptos…

Los dos protocolos principales de la capa de transporte son TCP (Transmission Control Protocol) y UDP (User Datagram Protocol). A lo largo de la práctica veremos algunas de sus características.

La capa de red o de Internet es la que nos permite interconectar diversas redes independientes y que pueden tener distintas estructuras. Entre los protocolos que componen esta capa podemos encontrar IPv4 ó IPv6 (Internet Protocol, versión 4 ó 6) o ICMP (Internet Control Message Protocol). Más adelante veremos sus propiedades.

También recuperaremos el protocolo DNS (Domain Name Server) y veremos la necesidad de contar con servidores DNS en nuestro ordenador para poder acceder a Internet, cómo se pueden configurar los servidores DNS y cómo funciona el protocolo.



Antes de empezar la práctica vamos a recuperar el programa que usamos también en la práctica anterior para monitorizar el tráfico de red (http://www.wireshark.org/). Instálalo en tu ordenador. Ejecútalo, y dentro del menú "Capture", en la opción "Interfaces", elige la interfaz de red activa ("Network Connection"). Ejecuta "cmd" y recupera y anota la IP de tu

ordenador por medio del comando "ipconfig" (de las distintas interfaces de red que aparezcan debes seleccionar la correspondiente a "Adaptador de Ethernet – Conexión de área local"). Ahora crea un filtro en Wireshark como el siguiente para conseguir aislar sólo el tráfico que tiene como origen o destino tu máquina:

ip.addr eq la_ip_de_tu_maquina

A lo largo de la práctica, a medida que Wireshark captura más paquetes, es probable que su comportamiento se ralentice. En ese caso puedes detenerlo ("Capture -> Stop") y reiniciarlo ("Capture -> Interfaces -> Network Connection") sólo al realizar los distintos ejercicios (ten en cuenta que empezará una nueva sesión o captura cada vez que lo arranques y se perderán los paquetes ya capturados).

1. Abre 5 pestañas en tu navegador. Trata de acceder con él a las siguientes direcciones:

193.146.250.30
74.125.159.105
66.220.153.11
10.0.1.31
193.144.2.30

¿Qué páginas se han abierto? Filtra en Wireshark toda la actividad DNS que ha habido al abrir las páginas anteriores (escribe "dns" en la ventana "Filter:"). El protocolo DNS es el que permite, a partir de una dirección URL, resolver su IP para que las cabeceras de la capa de red se puedan formar correctamente. ¿Ha necesitado el ordenador conectarse a los servidores DNS para resolver las anteriores direcciones? ¿Se ha conectado antes o después de hacer las solicitudes HTTP?

2. Vamos ahora a ver cómo funciona la capa de transporte. Dentro de la misma podemos encontrar principalmente dos protocolos, TCP y UDP. Filtra en Wireshark la comunicación que ha tenido lugar entre tu ordenador y la página web en 74.125.159.105. Puedes crear un filtro como:

ip.addr eq la_ip_de_tu_maquina and ip.addr eq 74.125.159.105

Observa la lista de paquetes que has recuperado. ¿A qué protocolos pertenecen? Recupera el mensaje de protocolo HTTP cuya "info" sea exactamente "GET / HTTP/1.1" y ábrelo (botón derecho sobre el mismo, opción "Show Packet in New Window").

Las cabeceras del protocolo HTTP ya las observamos en la práctica anterior. Observa las cabeceras TCP. Anota el valor de los puertos "Src" (puerto de tu máquina) y "Dst" (puerto del servidor). La comunicación entre máquinas siempre se hace a través de puertos de las mismas. Mientras dura una comunicación, cada máquina mantiene el mismo puerto escuchando y mandando mensajes. Algunos protocolos tienen un puerto asignado por defecto; por ejemplo, cuando te conectes a un servidor a través del protocolo http, lo normal es que el servidor utilice su puerto 80 para esa

comunicación (¿es cierto en este caso?). Si te conectas por ftp, por defecto el servidor utilizará el puerto 21.

Vamos a observar ahora la comunicación que ha habido a través del puerto de tu computadora. En primer lugar, vamos a filtrar los paquetes de los que hemos sido origen (o src). Utiliza el siguiente filtro de Wireshark:

tcp.srcport eq puerto_de_tu_maquina

¿Cuál es la IP de origen de dichas comunicaciones? ¿Y la de destino? ¿Qué recursos web se han solicitado por HTTP a través de ese puerto?

Veamos ahora los paquetes de los que hemos sido destinatarios (dst). Utiliza el filtro:

tcp.dstport eq puerto_de_tu_maquina

¿Cuál es la IP de origen de dichas comunicaciones? ¿Y la de destino? ¿Qué mensajes HTTP se han enviado a través de ese puerto?

Aparte de la información sobre los puertos, las cabeceras TCP contienen información que nos va a permitir asegurar la correcta recepción de los paquetes. Copia los números "Sequence number" y "Next Sequence Number" dentro de la cabecera TCP del mensaje "GET / HTTP/1.1".

Ahora crea el siguiente filtro:

tcp.ack eq next_sequence_number

¿Qué mensajes contienen el número next_sequence_number? ¿La respuesta del servidor "HTTP/1.1 200 OK" contiene el next_sequence_number? Pulsa el botón derecho sobre el mensaje cuya "info" es "HTTP/1.1 200 OK", opción "Show Packet in New Window" y trata de comprobar, en la sección de cabeceras de "Reassembled TCP Segments", qué paquetes han sido "reensamblados" para construir el mensaje de respuesta HTTP. Anota sus números y comprueba si aparecen también entre los mensajes filtrados por la regla "tcp.ack eq next_sequence_number".

Tratemos de dar una explicación al anterior hecho. Cuando tu ordenador hace una solicitud a un servidor, genera números "Sequence Number" y "Next Sequence Number". Estos números son generados de forma aleatoria. Cuando el servidor responda a tu mensaje, siempre incluirá un campo "Acknowledgment number" en el que enviará el número que nuestro ordenador le envió como "Next Sequence Number". Así nuestro ordenador consigue distinguir las respuestas a las distintas solicitudes que le haya hecho a ese servidor.

De no existir los números "Sequence" y "Acknowledgment", nos sería imposible distinguir las distintas comunicaciones con una misma máquina, y por tanto no podríamos reconstruir sus mensajes.

El protocolo TCP tiene aún más funciones. Veamos otra de ellas. Selecciona uno de los mensajes de la comunicación con 74.125.159.105 y pulsa sobre el mismo el botón derecho. Selecciona la opción "Follow TCP Stream". Observa los tres últimos mensajes de la comunicación. Si observas su "info", verás que dos de ellos contienen la orden "[FIN, …]". ¿En qué orden se ha puesto fin a la comunicación? ¿Quién es el que manda el primer mensaje avisando del fin de la misma?

Abre uno de los mensajes conteniendo "[FIN,…]" y trata de ver, en sus cabeceras "TCP", dónde se ha especificado que el mismo ponía fin a la comunicación (porque ya había recibido todos los paquetes TCP de la misma, y había reconstruido adecuadamente el mensaje recibido).

Puedes observar algo parecido en los dos primeros mensajes de la comunicación, que contienen la etiqueta "[SYN, …]" en su campo "info". En primer lugar un mensaje TCP con la etiqueta "[SYN]" por parte de nuestro ordenador solicitando la conexión. A continuación, un mensaje "[SYN,ACK]" del servidor, respondiendo a la solicitud y solicitando "Acknowlegdement" a nuestro ordenador. En tercer lugar, el mensaje con el flag "[ACK]" por parte de nuestro ordenador, en el que anuncia al servidor que está esperando el inicio de la comunicación. Esto es lo que se conoce como "negociación en tres pasos" de la conexión.

Como has podido observar, son mensajes TCP los que se encargan de establecer la conexión "[SYN]" y de terminarla "[FIN]".

Ya para terminar con el protocolo TCP, recapitulamos alguna de las limitaciones físicas del protocolo. Los mensajes TCP tienen una limitación de tamaño debido a los distintos protocolos de red que interactúan. Lee en http://es.wikipedia.org/wiki/Unidad_m%C3%A1xima_de_transferencia el límite de la unidad máxima de transferencia para redes Ethernet. Recupera algunos de los paquetes TCP de tus comunicaciones con 74.125.159.105 y comprueba que este límite se cumple. Anota en tu informe los límites correspondientes y el tamaño de los paquetes que has comprobado.

3. Seguimos trabajando con la capa de transporte. Vamos a ver ahora alguna de las propiedades del protocolo UDP. Reinicia Wireshark "Capture-> Interfaces" (en caso de que lo hayas parado en el ejercicio anterior; podría ser conveniente hacerlo para eliminar todos los paquetes indexados hasta ahora).

Abre la dirección www.rediris.es y filtra en Wireshark todos los mensajes que hayan aparecido del protocolo UDP. ¿Qué protocolos aparecen que reconozcas?

Vamos a seleccionar los mensajes de protocolo DNS de la "query" y la "response" de la url www.rediris.es. Observa la imagen de la pila de protocolos de la primera página de la práctica. ¿Sobre qué protocolo de la capa de transporte se encuentra el protocolo de la capa de aplicación DNS?

Abre los paquetes en una nueva ventana y observa sus cabeceras del protocolo de transporte (UDP). Apunta el tipo de información que se almacena en las cabeceras del protocolo UDP.

4. Para terminar con los protocolos de la capa de transporte, vamos a introducir otra de sus propiedades. El protocolo TCP sobre IP se define como un protocolo "orientado a conexión". Esto quiere decir que los mensajes deben ser recibidos por el destinatario en el mismo orden en que han sido enviados (recuerda que en el protocolo TCP sólo disponíamos de un número "ACK" que indicaba que un mensaje TCP era respuesta a una solicitud determinada, pero no había ningún indicador de cuáles eran los mensajes previos o posteriores a ese; si los mensajes no llegan en el mismo orden en que han sido enviados, la máquina receptora no podrá recomponerlos).

Sin embargo el protocolo UDP es un protocolo no orientado a conexión. Los distintos paquetes en que es dividida una comunicación no es necesario que lleguen de forma ordenada al destinatario. Sin embargo, los mensajes son mucho más breves (contienen menos información en sus cabeceras). Esto puede ser de gran utilidad para mensajes como los DNS, que generalmente ocupan un solo paquete. También lo es para aplicaciones de vídeo o audio, que requieren de un tráfico fluido de datos, y quizá no tan preciso (puedes encontrar más información al respecto en http://en.wikipedia.org/wiki/User_Datagram_Protocol#Applications).

Para que entiendas la diferencia entre un protocolo orientado a conexión (TCP) y otro que no lo es (UDP) puedes usar el "applet" en http://www3.rad.com/networks/infrastructure/packet/vcdd.htm#_app. Si pulsas la opción "Datagram" (y "Start") verás cómo se transmiten los paquetes en un protocolo no orientado a conexión (p.ej. UDP). Si eliges la opción "Virtual" verás el comportamiento de los paquetes en una red con protocolo orientado a conexión (TCP).

Compara la información que has recopilado sobre UDP (cabeceras, tipo de protocolos de uso, tipo de protocolo) con la que tienes sobre TCP. Puedes usar las ideas propias que has adquirido en los ejercicios 2 y 3 y la información en http://es.wikipedia.org/wiki/UDP#Comparativa_entre_UDP_y_TCP_.28Transmission_Control_Protocol.29. Incide en aspectos como cómo se establece y cierra la conexión en ambos protocolos, qué tipos de "flags" o información pueden transportar….

5. Vamos ahora a trabajar con la capa de protocolos de Internet. En primer lugar repasa la imagen al inicio de la práctica y observa algunos de los protocolos que se encuentran en la capa de red o de Internet. Vuelve a reiniciar la captura de Wireshark ("Capture -> Interfaces").

Vamos a utilizar una aplicación llamada "ping". El programa "ping" es una utilidad que se usa para diagnosticar el estado de una red (por ejemplo, que nuestro ordenador es capaz de conectarse a Internet, o que la página a la que nos queremos conectar es accesible). Para ello, el programa envía paquetes de datos a la máquina de destino, y espera que la misma devuelva los mismos datos.

Vamos a comprobar cómo funciona. Ejecuta "cmd":

- Ejecuta el mandato "ping 88.221.92.8".
- Filtra en Wireshark los paquetes por medio de "ip.addr eq 88.221.92.8". ¿A qué protocolo pertenecen esos mensajes? Encuéntralo en la pila de

protocolos TCP/IP. Selecciona un mensaje etiquetado como "request". Observa el campo "data" y compáralo con el del mensaje "reply" que le sigue.
- Ejecuta el mandato "ping 10.0.1.31". Anota el resultado (positivo o negativo).
- Ejecuta el mandato "ping www.uned.es". Anota el resultado y la IP de la máquina de destino.

Que un servidor no responda a una petición por "ping" puede deberse a varios motivos. Uno es que el servidor no esté disponible. Otro, muy común, es que el servidor prohíba las solicitudes por "ping" porque las mismas podrían dar lugar a un ataque de denegación de servicio sobre el servidor (http://es.wikipedia.org/wiki/Ataques_de_denegaci%C3%B3n_de_servicio, http://es.wikipedia.org/wiki/Ping_de_la_muerte, http://es.wikipedia.org/wiki/Ping_flood).

6. Vamos a hacer uso ahora de una segunda aplicación, "tracert", que nos permite seguir la ruta que sigue un paquete que sale de nuestra máquina dirigido a otra. También provee información sobre la "distancia" (en tiempo) entre nuestra máquina y el servidor requerido.

En la consola de MSDOS:

- Ejecuta "tracert 88.221.92.8". ¿Por cuántos servidores ha pasado el mensaje antes de llegar a destino? ¿Cuál es la primera dirección a la que llega el mensaje cuando sale de tu máquina? ¿Cuántos de esos pasos han sido en la red local (IP 10. … . … . …)?
- Ejecuta "tracert www.unirioja.es" (puedes observar entre corchetes la dirección IP de www.unirioja.es que ha resuelto nuestro servidor DNS; también lo podrías comprobar en Wireshark).
- Ejecuta "tracert 10.0.1.31" (ahora el servidor DNS no actúa).
- Ejecuta "tracert www.uned.es". ¿Por cuántos servidores ha pasado el mensaje antes de llegar a destino? ¿Cuál es la primera dirección a la que llega el mensaje cuando sale de tu máquina? ¿Cuántos de esos pasos han sido en la red local (IP 10. … . … . …)?

En el último de los casos, apunta la última dirección IP a la que ha llegado nuestro mensaje. Esa dirección será la puerta de entrada a la red en la que se encuentra la máquina donde se aloja www.uned.es.

Tanto el programa "ping" como "tracert" envían sus mensajes por medio de un protocolo llamado ICMP (Internet Control Message Protocol). Este mensaje forma parte de la capa de Internet. Puedes encontrar más información sobre el mismo en http://es.wikipedia.org/wiki/Internet_Control_Message_Protocol.

7. Vamos a ver ahora algunas de las propiedades del protocolo de red IP (IPv4). El protocolo IP es el que da nombre a Internet (Internet Protocol) y el que permite comunicar máquinas de distintas redes, siempre y cuando ambas tengan una dirección IP conocida.

Utiliza Wireshark para filtrar los mensajes del protocolo IP (simplemente escribe "ip" en la pestaña "Filter"). Como puedes observar, prácticamente todos los mensajes de la sesión pertenecen a este protocolo (algunos

mensajes usan ya protocolo IPv6, que no deja de ser una ampliación del protocolo IP).

En una pestaña del navegador, accede a la dirección www.rae.es. Busca el paquete http de la solicitud. Observa que tienes varias opciones. Una es recuperar primero el mensaje DNS con la IP de la página www.rae.es, y entonces filtrar por su IP. Otra es aplicar un filtro por http y recuperar todos los mensajes de ese protocolo. Elige la opción que consideres más conveniente.

Selecciona el mensaje que contenga la solicitud "GET /rae.html HTTP/1.1". Observa sus cabeceras "Internet Protocol". Anota la IP de destino y de origen de la comunicación. Anota también cada uno de los campos (no los valores) que aparecen en las mismas. Puedes encontrar en http://es.wikipedia.org/wiki/Cabecera_IP una explicación detallada de cada uno de los campos.

Entre sus características más relevantes está que no es orientado a conexión (aunque TCP sobre IP sí lo sea, por sus cabeceras adicionales), que contiene información referente a (la IP de) las máquinas de origen y destino, del protocolo (TCP, UDP, ICMP…) al que corresponde el paquete que transporta, de la vida útil del paquete ("Time to live" define el número de routers por los que debe pasar un paquete antes de destruirse, para evitar que haya paquetes enrutados por la red de forma indefinida), y que todos los paquetes que viajan por Internet lo hacen a través del protocolo IPv4 ó IPv6.

Para terminar la práctica, vamos a profundizar un poco en la idea de la utilidad y necesidad de los servidores DNS. Los servidores DNS son los encargados de convertir direcciones URL (de la forma www.nombrededominio.es) a números IP de las máquinas en que los mismos se alojan.

8. Visita la página web www.nic.es. NIC (Network Information Center) es la entidad encargada de asignar los nombres de dominios de Internet a personas o empresas para que los mismos, a través de un DNS, puedan montar sus sitios web mediante un proveedor de hospedaje.

Consulta en www.nic.es algunos dominios y observa cuál es el proceso y la información necesaria para el registro de los mismos. Comprueba también algunos dominios conocidos (unirioja.es, reinaleonor.es, a-prima.es) y los datos de sus propietarios, que son públicos.

9. Vamos a conseguir ahora algunas direcciones de servidores DNS. Entra en la página http://www.adslayuda.com/index.php?module=FSDns&order=info y anota la dirección de algunos de los servidores DNS de los que hay disponibles. Anótalas en tu informe de prácticas.

10. Vamos ahora a la configuración de red de tu ordenador ("Panel de Control->Redes e Internet->Ver el estado y las tareas de red->Conexión de área local->Propiedades->Protocolo de Internet Versión 4->Propiedades"). Modifica las direcciones de los servidores DNS. Escribe, por ejemplo, "1.1.1.1" y "2.2.2.2.". Cierra y guarda la configuración. En tu navegador

trata de abrir una página web cualquiera. Anota en tu informe el nombre de la página y el error que has encontrado.

En Wireshark, recupera el mensaje de protocolo DNS (filtro DNS) de la página web que has intentado abrir. ¿Cuál ha sido la IP de destino del mensaje DNS?

Si los servidores DNS no están bien configurados, el ordenador no es capaz de resolver ningún nombre de dominio, evitando que podamos navegar por la red (a no ser que conozcamos las IPs de las máquinas que queremos visitar; prueba a abrir en tu navegador la IP de la página www.rae.es y observa el resultado).

11. Vuelve a abrir la configuración de red e introduce las IPs de los servidores DNS que has apuntado en el punto 9. Trata de abrir una página web en tu navegador. ¿Cuál es el resultado obtenido?

Restituye la configuración de los servidores DNS a su estado original.

12. Por último, vamos a hacer uso de la utilidad "nslookup". Este programa nos permite conocer la respuesta de los servidores DNS a una solicitud de IP (a partir de un nombre de dominio).

Para cada nombre de dominio existen al menos dos servidores DNS autoritativos, que son aquellos en los que el nombre de dominio y su IP fueron introducidos directamente. Cualquier otro servidor DNS será capaz de responder a las solicitudes DNS, pero obtendría las IPs por medio de una búsqueda en otros servidores DNS.

Comprueba la anterior información por medio de los siguientes mandatos:

- nslookup www.unirioja.es (anota la respuesta y si el servidor DNS es autoritativo o no);
- nslookup pop.unirioja.es (anota la respuesta y si el servidor DNS es autoritativo o no);
- nslookup www.rae.es (anota la respuesta y si el servidor DNS es autoritativo o no).

Ahora vamos a hacer las solicitudes de DNS a un servidor distinto. Recupera una de las direcciones de servidores DNS que has apuntado en el ejercicio 9. Sustitúyela por "ip_de_un_servidor_dns" en los siguientes comandos:

- nslookup www.unirioja.es ip_de_un_servidor_dns (anota la respuesta y si el servidor DNS es autoritativo o no);
- nslookup pop.unirioja.es ip_de_un_servidor_dns (anota la respuesta y si el servidor DNS es autoritativo o no);
- nslookup www.rae.es ip_de_un_servidor_dns (anota la respuesta y si el servidor DNS es autoritativo o no).

13. Sube el informe de la práctica a belenus y enlázalo desde tu página de inicio en el mismo.