***Redes de Computadores***
***Estudio Experimental Sesión de laboratorio 2:***
***Aspectos básicos de la interfaz de Wireshark. Programación de sockets.***

Departamento de
Tecnología Electrónica
Última actualización: 14/10/19

**Estudio experimental**

El estudio experimental de esta práctica consta de tres partes. En cada una de ellas se describen todos los pasos que el alumno debe realizar. **Si tiene cualquier duda consulte con el profesor encargado de la sesión práctica**. En el caso de no completar todas la partes del estudio experimental, antes de abandonar el laboratorio debe realizar el punto 60.

**(NO ENCIENDA EL PC HASTA QUE SE LO INDIQUEN)**

## Pasos previos

1. Asegúrese de que está conectado a la red ETSII.
2. Encienda su PC, restaure (**si así se lo indica el profesor**) y arranque Windows 7. Desactive el firewall tal y como hizo en los pasos 1 al 4 de la primera sesión de laboratorio.
3. Desconecte su PC de la red ETSII y conéctelo a la Intranet del laboratorio, concretamente al **SWITCH_EUROPA** (en G1.31) o al **SWITCH_SUDAMERICA** (en G1.33), de forma similar a como lo hizo en los pasos 40 al 44 de la primera práctica, sólo que ahora se conecta a un SWITCH y no a un HUB. **Si no hubiese puertos libres en el SWITCH adecuado indíqueselo al profesor.**
4. ¿Cómo puede estar seguro de que tiene conectividad a nivel físico entre su PC y el SWITCH?

5. En la Intranet del laboratorio, LAB_DTE, los PC conectados a ella requieren de una configuración TCP/IP específica que se obtiene automáticamente de un servidor DHCP. Realice los pasos oportunos para obtener de forma automática la nueva configuración TCP/IP de su PC como hizo en el apartado 46 de la práctica anterior. Anote la dirección IP de su PC.
6. Use un comando que le permita comprobar que tiene conectividad a nivel de red (intercambio de R_PDUs) con su router frontera (puerta de enlace predeterminada). Este tipo de comprobaciones aprendió a hacerlas en la primera práctica. No siga adelante hasta que tenga conectividad.

## Primera Parte: Continuando con el aprendizaje del manejo del analizador Wireshark

Como ya sabemos de la práctica anterior, Wireshark es una herramienta de software libre multiplataforma (Windows, Linux y MacOS) que puede descargar de forma gratuita de http://www.wireshark.org. Wireshark es un programa que, ejecutado en un sistema final, es capaz de capturar todo el tráfico de red recibido o enviado por dicho sistema. Es una herramienta de gran utilidad, pues no sólo captura el tráfico sino que además es capaz de analizarlo, mostrando al usuario información detallada de los protocolos de cada uno de los niveles (desde el nivel de enlace de datos hasta el nivel de aplicación). Es por ello que recibe el nombre de analizador de protocolos. Es una herramienta muy completa y compleja, por lo que vamos a centrarnos en el manejo básico de la interfaz de la misma y a la vez utilizar algunos comandos y opciones fundamentales para el trabajo habitual con esta herramienta.

7. Arranque el analizador de protocolos **Wireshark**
8. Wireshark es capaz de utilizar los servicios de DNS para, en sus diversas ventanas, mostrarnos siempre nombres de host y dominio, en lugar de mostrarnos las direcciones IP equivalentes, en el formato numérico xxx.xxx.xxx.xxx habitual. Esa característica nos será de mucha utilidad en esta práctica. Entre en "Edit" → "Preferences", pulse "Name Resolution" en el panel de la izquierda, **active** la opción **"Resolve Network (IP) addresses"**. Wireshark también es capaz de mostrarnos, en lugar de los números de puerto TCP y UDP, el nombre del protocolo que usa habitualmente dicho número de puerto. En esta práctica concreta no nos interesa habilitar esta funcionalidad de Wireshark. Entre en "Edit" → "Preferences", pulse "Name Resolution" en el panel de la izquierda, **desactive** la opción **"Resolve Transport Name"** y pulse **"OK"** para cerrar la ventana y que tengan efecto los cambios.
9. Haga que Wireshark comience a capturar el tráfico que entra y sale de su conexión de red Ethernet, tal y como hizo en la primera práctica en el apartado 55.
10. Genere tráfico de red abriendo el navegador Mozilla Firefox y visitando la página web:
    **http://www.redes.lab/index.html**
11. Podrá ver que Wireshark le muestra en el panel superior de su ventana principal las tramas que ha capturado, fruto del tráfico de datos entre el cliente y el servidor web. Detenga la captura de tráfico cuando observe que la carga de la página anterior ha terminado.

12. Como ya sabe de la primera práctica, en el listado de tramas podemos ver mucha información de cada trama, organizada en columnas. Las que aparecen por defecto son:
    a) La primera columna se llama "No." y nos muestra el número de orden en el que se han ido capturando las tramas, de la 1 a la N.
    b) La segunda columna se llama "Time" y en ella Wireshark nos muestra, en segundos, información temporal del instante en que fue capturada esa trama. Por defecto este tiempo se mide desde el instante en que se capturó la primera trama, por lo que en la trama número 1 es 0.000000.
    c) La columna "Source" muestra información del equipo que envió la trama (o el que envió alguna PDU encapsulada en dicha trama, depende de cómo hayamos configurado Wireshark).
    d) La columna "Destination" es análoga a la anterior, mostrándonos información del equipo destino.
    e) La columna "Protocol" muestra información de protocolo de más alto nivel encapsulado en esa trama y que es capaz de analizar.
    f) La columna "Length" muestra el número de bytes de la trama. En la última sesión de laboratorio se verá qué campos de la trama (E_PDU) incluye.
    g) La columna "Info" muestra información resumida del protocolo de más alto nivel que es capaz de analizar en esa trama.
13. Es posible quitar y añadir columnas de información al listado de tramas, para adaptarlo a nuestras necesidades en cada momento. En esta práctica será necesario añadir dos nuevas columnas que nos presenten información de los puertos de origen y de destino de las T_PDU de los protocolos TCP y UDP. Con esto podremos identificar el número de puerto usado para identificar al proceso de aplicación cliente y servidor en una trama que encapsule protocolos hasta el nivel de aplicación. Para hacerlo debe seguir estas instrucciones:
    a) Entre en "Edit" → "Preferences", y pulse en la rama "Columns" (dentro de "User Interface" en el panel de la izquierda).
    b) Pulse el botón "Add" una vez para añadir una nueva columna.
    c) Haga clic en el texto "New column" que ha aparecido, y edítelo escribiendo como título de la nueva columna el texto "SrcPort" y pulsando "Intro" en el teclado.
    d) En el campo "Field Type" debe seleccionar de la lista desplegable el valor "Src Port (unresolved).
    e) Repita los pasos b), c) y d) para crear otra columna con título "DstPort" y que tenga "Dest Port (unresolved)" de "Field Type".
    f) Pulse "OK" para cerrar la ventana "Preferences".
    g) Observe que en el listado de tramas aparecen las dos nuevas columnas en la parte de la derecha (si no puede verlas desplácese hacia la derecha). Utilice el ratón para reordenar las columnas y colocar las dos nuevas delante de la columna “Info” o bien ajuste el ancho de la columna "Info" para que se muestren todas ellas en pantalla sin tener que desplazarse.
14. Como ya sabe, la ventana principal de Wireshark está dividida en tres paneles. Ya hemos repasado el panel superior, el listado de tramas. Los otros dos paneles están muy relacionados con el panel superior, pues nos muestran información de la trama que hayamos seleccionado en el listado de tramas.
15. El panel central, "Detalles de la trama", muestra diversa información de la trama y de su contenido, de forma ordenada y estructurada por niveles. En primer lugar se muestra información de la trama completa y luego se va mostrando información de cada uno de los niveles, empezando desde el nivel de enlace, a continuación red, transporte y aplicación (si es que aparecen todos, cosa que no siempre ocurre). En cada línea hay un "+" a la izquierda para desplegar la información del protocolo asociada a cada nivel (una vez interpretada por la herramienta). No toda la información que aparece de un determinado protocolo forma realmente parte de dicho protocolo. A veces Wireshark añade información que ha determinado como resultado de un análisis que ha realizado a nivel global, en cuyo caso esta información aparece entre corchetes [ ]. Por otro lado, tampoco todo lo que aparece detrás de un "+" es necesariamente un protocolo. Por ejemplo, Wireshark es capaz de analizar diferentes formatos de ficheros como GIF, PNG, JPG, etc. y los muestra a la derecha de un "+". Seleccione con el ratón una trama que en la columna "Protocol" muestre HTTP y fíjese en los nombres de los protocolos que aparecen en el panel central. En caso de no encontrar ninguna, porque haya hecho mal la captura de los apartados 9, 10 y 11, deberá volver a iniciar la captura en Wireshark y ordenarle a Firefox que haga una recarga de la página actual (**http://www.redes.lab/index.html**), pulsando sobre el icono de recarga (la flecha gris enroscada que está a la derecha de la URL de la página). Espere a que acabe la carga y detenga la captura en Wireshark. Nótese que pulsando F5 en Firefox también se recarga la página actual como si se pulsase sobre el icono de la flecha enroscada.
16. El panel inferior, "Bytes de la trama", muestra un volcado en hexadecimal y en ASCII del contenido de la trama seleccionada. Los datos en hexadecimal (en la parte izquierda) se presentan en filas de 16 bytes, junto con una primera columna que indica la posición relativa (dentro de la trama) del primer octeto de la fila. Si en el panel central se hace clic en alguno de los niveles (o en algún campo dentro de estos) se resaltan con fondo oscuro en el panel inferior los bytes asociados a aquello sobre lo que hemos hecho clic. Al revés también funciona, pulsando sobre bytes del panel inferior y viendo en el panel central como se selecciona el campo de información correspondiente. Haga clic en "detalles de trama" en "Hi-

pertext Transfer Protocol" para seleccionar el protocolo HTTP. ¿Qué información aparece en ASCII en "bytes de tramas"? Pulse sobre el "+" que aparece al lado de "Hipertext Transfer Protocol" en "detalles de trama", observará el contenido de la HTTP_PDU. Haga clic varias veces en diferentes líneas de cabecera y observe como se ve en ASCII esa información. ¿Cómo se muestra en ASCII los códigos de control '\r' y '\n"?

17. Como sabe, el contenido mostrado en el listado de tramas puede ser guardado en un archivo (entrando en File → Save o haciendo clic en  ), el cual puede ser cargado en cualquier otro momento (entrando en File → Open o haciendo clic en  ). Esto le puede ser de gran utilidad si le faltase tiempo para completar esta práctica, pues podría llevarse la captura a su casa y acabar allí la parte del estudio experimental que no haya podido terminar.

Respecto a las marcas de tiempo, mostradas en la columna "Time" del listado de tramas, la primera trama tiene por defecto la marca 0.000000 segundos y el resto de marcas van incrementándose respecto a esta. No obstante, es posible establecer una marca de referencia en cualquier trama de forma que sea el "cero" para todas las tramas a continuación de ella, que verán su marca de tiempo modificado considerando esa referencia, lo cual es útil para medir tiempos entre tramas desde una primera que será la que tomemos como "referencia". Para ello seleccionamos la trama que queremos marcar como referencia con clic-derecho y elegimos "Set Time Reference (Toggle)", apareciendo ***REF*** en esa trama y modificándose el tiempo de las tramas siguientes. Si repetimos la operación se quita la referencia de esa trama. Tenga en cuenta que puede haber varias "referencias locales" en el listado de tramas.

18. Fíjese que en el listado de tramas hay varias tramas que contienen PDUs del protocolo HTTP (fíjese en el valor de la columna "Protocol"). Concretamente, debe encontrar una trama que muestre en la columna "Info" que contiene una petición GET del protocolo HTTP.
19. La petición GET la ha emitido el proceso cliente, que está asociado a un determinado puerto en el host origen ("source host"). Gracias a la columna SrcPort (puerto fuente o puerto origen) podemos averiguar con facilidad el número de puerto asociado al proceso cliente.
20. Compruebe que el valor numérico de la columna DstPort de la trama que encapsula el GET es el valor habitual usado por un proceso servidor del protocolo HTTP. Anótelo.

21. Compruebe también que en la columna “Destination” NO aparece como nombre del servidor www.redes.lab sino otro distinto (webserver.af.lab). Esto es debido a que www.redes.lab es un alias de webserver.af.lab, es decir, ambos son nombres asociados al mismo sistema final. El nombre principal se conoce como CNAME o nombre canónico en la terminología DNS (Canonical NAME) y es el que aparece en la columna “Destination”. Anote la correspondencia entre el nombre canónico y el alias del servidor web.

22. Ahora vamos a hacer que la trama con el GET pase a ser la "referencia temporal" con respecto a la cual medir los tiempos de captura de las tramas capturadas detrás de ella. Para ello seleccione dicha trama haciendo clic sobre ella con el botón derecho del ratón y seleccione en el menú contextual que le aparece la opción "Set Time Reference (Toggle)". Fíjese que ahora esa trama no tiene marca de tiempo en la columna "Time" sino que aparece el texto ***REF*.** Es posible anular esta operación repitiendo los mismos pasos que hemos dado sobre esa trama.
23. Localice, detrás de la trama del GET, una trama que encapsule la respuesta HTTP a dicho GET. En la columna "Info" de la respuesta debe aparecer la línea de estado "HTTP/1.1 200 OK" o bien la línea de estado "HTTP/1.1 304 Not modified", dependiendo de las circunstancias en que se generó el GET con Mozilla Firefox.
24. Compruebe que en la respuesta se usan los mismos puertos que en la solicitud, pero puerto origen es ahora puerto destino y viceversa.
25. Compruebe que ocurre lo mismo con los valores de las columnas "Source" y "Destination".
26. Como hemos hecho que la trama del GET sea la referencia temporal de todas las tramas que le siguen, es muy fácil medir el tiempo transcurrido entre la emisión del GET y la recepción de la respuesta.
27. ¿Cuánto vale el RTT entre su PC y el servidor web www.redes.lab? Avise al profesor para que compruebe el valor de RTT.

28. Wireshark es capaz de, a partir de una trama cualquiera que contenga una T_PDU del protocolo TCP (o UDP), localizar todas las demás T_PDU que se transmitieron en la misma conexión TCP que ella (o en el caso de UDP, las T_PDU entre el mismo cliente y servidor usando una determinada pareja de puertos UDP). Gracias a eso puede mostrarnos el flujo de bytes transmitidos a través de esa conexión TCP por los procesos cliente y servidor (o el intercambio de A_PDUs en el caso de UDP). Seleccione la trama del GET haciendo clic sobre ella con el botón derecho del ratón y seleccione en el menú contextual que le aparece la herramienta "Follow TCP Stream". (Nota: Para UDP sería "Follow UDP Stream")
29. En la ventana "Follow TCP Stream" podemos ver en color rosa los bytes enviados por el proceso cliente y de color morado los bytes enviado por el proceso servidor. Si el cliente y el servidor mantienen un diálogo "largo" a través de una misma conexión (como en las conexiones HTTP persistentes) podría verse como se van alternando los mensajes del cliente y del servidor.

## Segunda parte: Filtros de visualización

En el cuadro de texto etiquetado "Filter:" que aparece bajo la barra de iconos de Wireshark puede escribir una expresión lógica con una determinada sintaxis, que hace que se muestren en el listado de tramas sólo aquellas que hacen que la expresión lógica sea cierta. Esta expresión lógica recibe el nombre de "filtro" y sólo afecta a lo que se muestra en el listado de trama, es decir, ni elimina una trama ya capturada ni tampoco es tenido en cuenta a la hora de decidir si una trama debe o no capturase. Se trata sólo de un filtro de visualización ("display filter") y no de un filtro de captura ("capture filter").

Hay muchas expresiones de filtrado que podemos introducir para cada protocolo reconocido por Wireshark. Puede ver todas las expresiones que hay para cada protocolo, haciendo clic en el botón "Expression...", pero si conoce la expresión puede escribirla directamente en el cuadro de texto "Filter:". Wireshark va mostrando una ayuda con las expresiones compatibles con lo que ya lleva escrito e incluso las autocompleta a medida que las escribe. Si la expresión es correcta aparecerá con fondo VERDE y si es incorrecta el fondo será ROJO. Para que Wireshark utilice el filtro que acabamos de escribir, es necesario pulsar sobre "Apply", opción que podemos ver a la derecha del filtro. Si deseamos volver a ver todas las tramas basta con pulsar sobre "Clear". Pulse ahora sobre "Clear" si el cuadro de texto "Filter:" mostrase algún filtro. Algunas expresiones básicas son:

- "ip.addr == 193.1.10.1", que muestra tramas que contienen R_PDUs cuya dirección IP origen o destino sea la 193.1.10.1.
- "tcp.port == 80", para mostrar el tráfico con número de puerto origen o destino el 80.
- "udp.port == 53", que hace lo mismo pero con UDP.
- "ip.src" y "ip.dst" son parecidos a "ip.addr" pero sólo se fijan en que la IP especificada esté en el origen o en el destino.
- "tcp.dstport" y "tcp.srcport" se fijan solamente en el puerto destino o el origen.
- "http", "dns", "tcp", "udp" y "icmp" son expresiones sencillas que hacen que sólo se muestren tramas que encapsulen PDUs de esos protocolos.

Es posible construir expresiones lógicas complejas combinando expresiones sencillas con los operadores lógicos "and", "or" y "not". Por ejemplo "http or dns" captura tramas que hacen que la expresión lógica "combinada" sea cierta. Es decir, aquellas que hacen que "http" sea cierta y también aquellas que hacen que "dns" sea cierta. También es posible utilizar el operador "contains" que permite buscar cadenas dentro de la PDU de un protocolo, por ejemplo, "http contains GET" permitiría ver todas las tramas que encapsulan PDUs HTTP que contienen la palabra "GET". Otro ejemplo con este operador sería "dns contains www" que nos dejaría ver solo las tramas con PDUs DNS en las que aparezca la cadena "www".

Fíjese que si marca una trama como referencia temporal con ***REF*** (recuerde el apartado 22), esa trama siempre se visualiza en el listado de tramas, sea cual sea el filtro de visualización aplicado.

30. Si no tiene una captura hecha que le llene completamente el listado de tramas, haga una nueva generando el tráfico de red que sea necesario y luego detenga la captura.
31. Escriba un filtro sencillo, como "http", "tcp", "udp", etc. y aplíquelo.
32. Observe cómo en el borde inferior de la ventana de Wireshark aparece el texto "Packets:" indicando el número total de tramas capturadas y el texto "Displayed:" indicando el número de tramas que han pasado el filtro de visualización y pueden verse en el listado de tramas.
33. Haga clic en "Clear" para borrar el filtro de visualización y volver a ver todas las tramas capturadas.
34. Aplique un filtro que muestre sólo el tráfico de nivel de red con origen o destino su propia IP. Avise al profesor para que compruebe el resultado del filtrado.

## Tercera parte: Programación de sockets[1]

Un *socket* es una interfaz de programación de aplicaciones (API) creada por una aplicación y controlada por el sistema operativo. A través de esta interfaz el proceso de aplicación puede enviar/recibir mensajes a/desde otros procesos remotos (o incluso locales). Esta interfaz es distinta dependiendo si se quiere una transferencia fiable (TCP) o no (UDP). A la hora de crear el *socket* se decide el tipo de transferencia que se desea realizar.

Si la transferencia es fiable, el socket se trata como un *buffer* (*stream*) del que se puede leer y escribir, ofreciendo por tanto un servicio de entrega fiable de un flujo de bytes/caracteres (y no de A_PDUs). Un aspecto importante en este tipo de transferencias es que debe establecerse una conexión entre el cliente y el servidor como paso previo a la comunicación de los datos. Para ello, el servidor debe crear un tipo especial de *socket* (*socket* de escucha) que acepte las solicitudes de conexión de los clientes. Cada vez que el servidor acepta una conexión se crea un nuevo *socket* (*socket* de conexión) que permite atender única y exclusivamente al cliente que realizó dicha solicitud.

Si la transferencia es no fiable, el *socket* proporciona una transferencia no fiable de datagramas (grupo de bytes) entre el cliente y el servidor. En este tipo de transferencias no se establece ningún tipo de conexión previa a la comunicación de los datos. Todo esto provoca que los datos transmitidos puedan recibirse de manera desordenada o incluso que algunos puedan perderse sin que los procesos de aplicación se den cuenta de ello.

Para ilustrar el funcionamiento de estos dos tipos de transferencias, se plantea el desarrollo de una pequeña aplicación, que consiste en un conversor de minúsculas a mayúsculas y que se basa en el modelo cliente-servidor. La operación de esta aplicación se resume a continuación: 1) el cliente envía al servidor un mensaje de texto que el usuario ha introducido por teclado, 2) el servidor recibe el mensaje, lo convierte a mayúsculas y responde al cliente, y 3) el cliente recibe la respuesta y muestra el mensaje de texto en mayúsculas por pantalla.

35. Para realizar las pruebas se proporcionan todos los ficheros fuentes (en lenguaje Java) necesarios. El profesor indicará la localización desde la que podrá descargarlos.
36. Una vez descargado el fichero comprimido que contiene los fuentes, debe extraerlos en la carpeta c:\RC.
37. Como paso previo a compilar y ejecutar las aplicaciones, debe abrir una ventana de Símbolo del sistema y situarse en la carpeta anterior ejecutando el comando **cd c:\RC**. Para consultar la lista de archivos que hay dentro de la carpeta puede usar el comando **dir**. Como resultado debe visualizar una serie de archivos con extensión .java, correspondientes a los clientes y servidores TCP y UDP. No siga adelante hasta que visualice estos archivos.
38. Para compilar las aplicaciones debe usar el comando **javac**. Como parámetro obligatorio de este comando debe indicar el fichero *.java* que se quiere compilar. Como resultado de la compilación se obtiene un fichero ejecutable con extensión *.class.* El nombre de este fichero corresponde al nombre de la clase principal contenida en el fichero *.java*.
39. Realice la compilación del cliente y del servidor TCP. Para comprobar que la compilación se ha realizado correctamente ejecute de nuevo el comando **dir**. Debe observar que aparecen dos archivos *.class* correspondientes al cliente y al servidor TCP.
40. Para ejecutar las aplicaciones debe usar el comando **java**. Como parámetro obligatorio debe indicar el nombre del archivo *.class*, es decir, el nombre de la clase. Para que funcione, en ningún caso debe incluir la extensión *.class*.
41. Ejecute en el Símbolo del sistema la aplicación servidor TCP. Debe tener en cuenta que si ejecuta primero la aplicación cliente recibirá un mensaje de error, ya que no habría ningún servidor aceptando las solicitudes de conexión de los clientes.
42. Abra una nueva ventana de Símbolo del sistema y sitúese en la carpeta c:\RC (si no recuerda como hacerlo revise el apartado 37). Ejecute en el Símbolo del sistema la aplicación cliente TCP e interactúe con la misma siguiendo los pasos que le vaya indicando.
43. Puede observar que tanto el cliente como el servidor proporcionan información de todas las tareas que van realizando. Revise estos mensajes y compruebe si todo el proceso se ha realizado correctamente.

[1] Si no le diera tiempo a realizar la tercera parte del estudio experimental en el laboratorio puede hacerla en casa. Para ello, sólo es necesario descargar el código fuente de las aplicaciones y seguir detenidamente los puntos descritos. Para realizar los apartados 45 y 54 debe usar otro sistema final (accesible a través de la conexión de red) y ejecutar ambos servidores en él. Para editar el código fuente puede utilizar Notepad++ o cualquier editor de texto similar. Puede venir a las tutorías de su profesor/a de prácticas si tiene algún problema a la hora de realizar esta parte.

44. Edite con Notepad++ el código fuente del cliente TCP y busque la línea de código donde se crea el *socket.* ¿Qué dirección IP o nombre de host se está usando identificar al servidor? ¿A qué sistema final identifica esta dirección IP o nombre especial? ¿Qué tipo de pruebas permite realizar?
45. Modifique el código del cliente para poder conectarse al servidor TCP del PC del profesor. ¿Qué parámetro es necesario modificar en el código del cliente?
46. Inicie Wireshark y comience a capturar tráfico.
47. Realice una prueba con el servidor del PC del profesor y compruebe su correcto funcionamiento.
48. Detenga las aplicaciones cliente y servidor TCP.
49. En la ventana de Wireshark aplique un filtro que permita visualizar la comunicación entre el cliente y el servidor TCP. ¿Qué filtro ha aplicado?

50. Seleccione una de las tramas de la captura y utilice la herramienta "Follow TCP Stream".
51. Realice la compilación del cliente y del servidor UDP. Revise los apartados 38 y 39 si no recuerda el procedimiento.
52. Ejecute en el Símbolo del sistema la aplicación servidor UDP. En este caso si ejecuta primero la aplicación cliente no recibirá ningún mensaje de error, a diferencia de TCP. Esto se debe a que UDP es un protocolo no orientado a la conexión, es decir, un cliente no necesita establecer una conexión con el servidor como paso previo a la comunicación de los datos. No obstante, hasta que el servidor no se ponga en funcionamiento todos los datagramas enviados por el cliente no serían procesados por ninguna aplicación.
53. En la otra ventana de Símbolo del sistema ejecute la aplicación cliente UDP. Interactúe nuevamente con la aplicación y compruebe el correcto funcionamiento de la misma.
54. Edite con Notepad++ el código fuente del cliente UDP y modifíquelo para permitir la comunicación con el servidor UDP del PC del profesor.
55. Realice una nueva captura de Wireshark.
56. Inicie de nuevo el cliente UDP y realice una prueba con el servidor UDP del profesor.
57. En la ventana de Wireshark aplique un filtro que permita visualizar la comunicación entre el cliente y el servidor UDP. ¿Qué filtro ha aplicado?

58. Seleccione una de las tramas de la captura y utilice la herramienta "Follow UDP Stream".
59. Detenga las aplicaciones cliente y servidor UDP.
60. Cierre todas las ventanas abiertas en su PC.
    Desconecte su PC de la Intranet del laboratorio.
    Conecte su PC a la red ETSII.
    Apague el PC.
    Vuelva a dejar en su sitio el latiguillo que conectó al **SWITCH EUROPA** o al **SWITCH_SUDAMÉRICA**.