Introducción al analizador de protocolos *WireShark*

Los analizadores de protocolos o de red, también conocidos vulgarmente como "*sniffers*" son herramientas de gran ayuda para los administradores de las redes de computadores, ya que permiten el análisis detallado de muchos factores del comportamiento de las mismas. Estas aplicaciones permiten capturar una copia de los paquetes que circulan por la red para su análisis posterior. Los más avanzados incluyen una interfaz gráfica capaz de mostrar los campos de los protocolos de comunicación de los distintos niveles, obtener estadísticas de utilización y facilitar considerablemente el posterior análisis de los datos capturados. De este modo se facilita la detección de problemas, así como la depuración del software de red durante su fase de elaboración. Por ejemplo, un administrador de red que detecte que las prestaciones de la red son bajas puede utilizar uno de estos analizadores para detectar qué segmentos de la red, protocolos y máquinas están generando más tráfico, y de esa forma llevar a cabo las acciones necesarias, o bien verificar el correcto funcionamiento de los diferentes dispositivos de red (*hosts*, servidores, *routers*, cortafuegos, NAT, etc).

Desde el punto de vista docente, los analizadores de protocolos permiten ver de forma práctica los protocolos de comunicación ya presentados en las clases de teoría, así como las relaciones entre los protocolos de distinto nivel. Por todo ello, intentaremos familiarizar al alumno con el uso de una de estas herramientas.

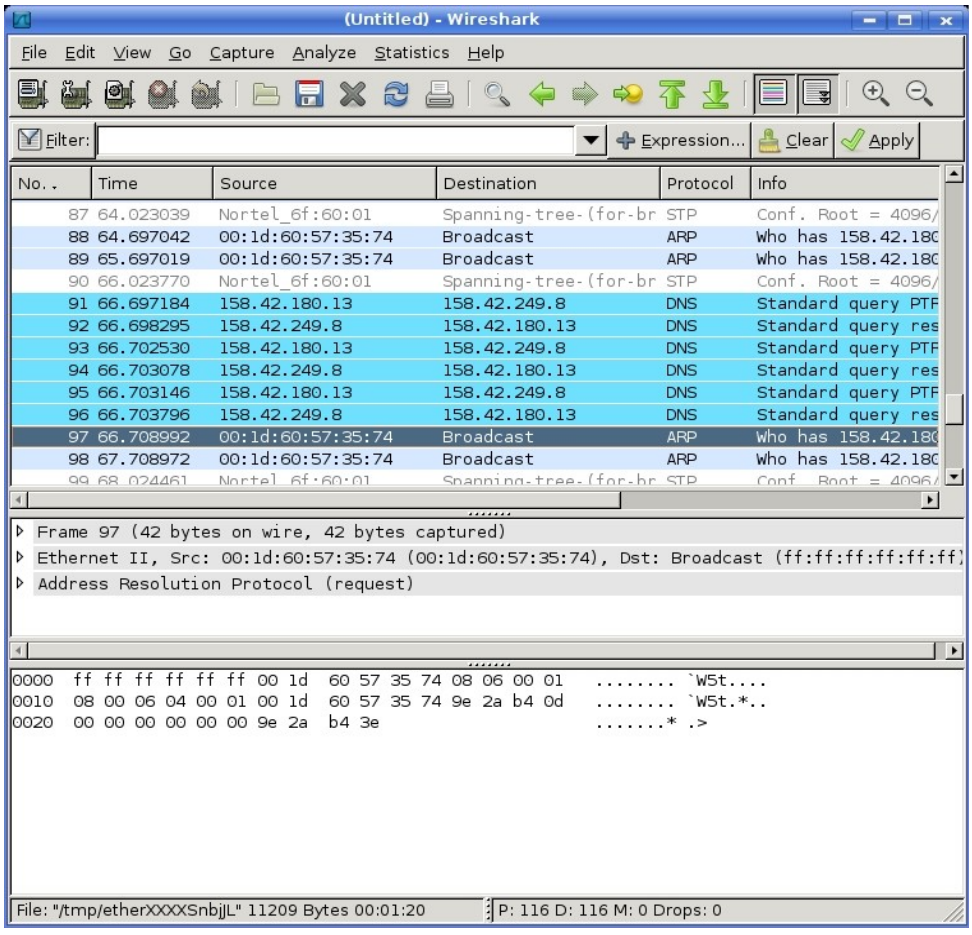
En esta introducción se pretenden adquirir las capacidades necesarias para capturar paquetes usando la herramienta *WireShark*. Dado que por la red viajan multitud de paquetes, será necesario seleccionar aquellos que nos resulten de interés. Por ello vamos a aprender a capturar paquetes utilizando los filtros que nos proporciona *WireShark*, de manera que aceptaremos unos paquetes y desecharemos otros. También, se pretende introducir al alumno en la interpretación del contenido de los paquetes capturados para afianzar los conceptos relativos a los protocolos estudiados en clase. Todo ello permitirá poner en práctica los conocimientos adquiridos a lo largo de varios de los temas vistos en las clases de teoría, adquiriendo una mayor comprensión de los procesos que ocurren en la red cuando se llevan a cabo diversas acciones a nivel de usuario.

1. El analizador de protocolos: *WireShark*

A la hora de elegir un analizador de protocolos nos encontramos con una abundante oferta, tanto de productos comerciales como de software de libre distribución. Uno de los más populares, y el elegido para la práctica de hoy, es *WireShark*. Se trata de un producto gratuito y muy versátil, que puede descargarse desde http://www.wireshark.org. Está disponible tanto para sistemas Windows como Unix, y permite no sólo capturar tráfico de una red, sino también filtrarlo y analizarlo. Además, permite leer ficheros de datos recogidos con otros analizadores de protocolos como *tcpdump* o *NetXRay*, con lo que pueden aprovecharse otras capturas previamente realizadas.

Por seguridad, los analizadores de protocolos requieren permisos de administrador del sistema para poder realizar capturas del tráfico de la red.

Figura 1. Descripción de WireShark



Comenzaremos comentando el aspecto habitual del programa, que se muestra en la figura 1. *WireShark* comprende tres ventanas o áreas principales.

1) La ventana superior es la lista de los paquetes. Muestra una breve descripción de cada paquete capturado. Pulsando en alguno de los paquetes de esta lista podemos controlar lo que se visualiza en las dos ventanas restantes.
2) La ventana intermedia muestra con mayor detalle el paquete seleccionado en la primera ventana. Indica los protocolos empleados en los distintos niveles de la arquitectura, así como los valores de cada uno de los campos de cada protocolo.
3) Por último, la ventana inferior muestra el valor de los datos, en hexadecimal y en ASCII, del paquete seleccionado en la ventana superior, y marca en negro los datos seleccionados en la ventana intermedia.

Además de estas tres ventanas, la ventana principal de *WireShark* ofrece (en la parte inferior) el filtrado en pantalla de los paquetes capturados en función del tipo de paquetes y/o contenido de sus campos. Este filtrado *a posteriori* es complementario al filtrado de paquetes en el momento de la captura, tal y como se explicará más adelante.

2. Cómo capturar paquetes

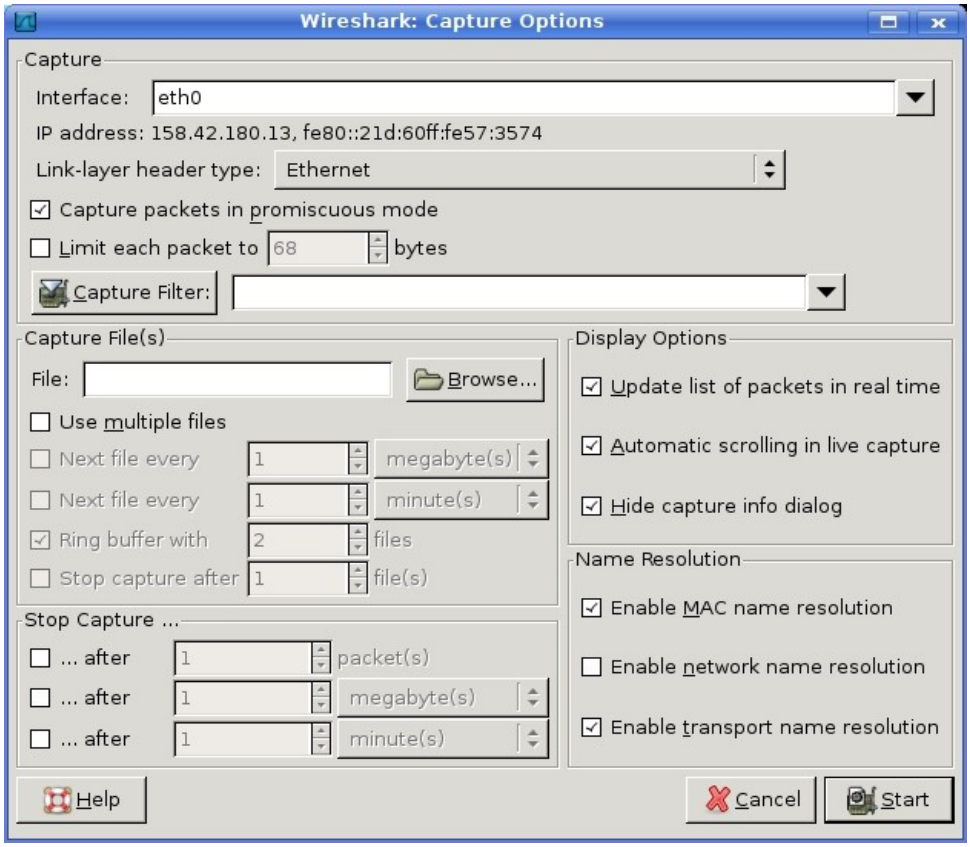


Figura 2. Ventana de captura de paquetes

Realizar una captura de paquetes es sencillo. Primero accederemos al menú *Capture* y allí seleccionaremos la opción *Options*. Esto nos lleva a una nueva ventana

similar a la mostrada en la figura 2, llamada *Capture Options*. En esta ventana podremos especificar los diversos parámetros relacionados con la captura.

El primer parámetro que podemos especificar es el interfaz, es decir, la tarjeta de red sobre la que queremos realizar la captura. Esta opción sólo tiene sentido si disponemos de varias tarjetas de red (posiblemente conectadas a diferentes redes), como es el caso de los computadores del laboratorio. En este caso, el interfaz a seleccionar es *eth0*.

Con el fin de evitar un consumo excesivo de memoria también podemos indicar la cantidad de bytes que vamos a guardar de cada paquete capturado. No obstante, para capturas pequeñas como las que vamos a realizar, esta opción no es necesaria.

Otra opción que podemos detallar es si deseamos realizar una captura en modo promiscuo. En este modo el programa capturará cualquier paquete que sea visible a la tarjeta de red, independientemente de si está destinado a ella o no. Por el contrario, podemos seleccionar la captura de únicamente aquellos paquetes que van destinados a, o que provienen de, nuestra tarjeta de red.

En esta ventana también podemos introducir un **filtro de captura**, con el fin de procesar después más fácilmente la información obtenida. El uso de este tipo de filtros lo describiremos más adelante.

Otra posibilidad que nos ofrece esta ventana es la de volcar los paquetes capturados a un fichero. Esto puede resultar interesante para guardar un registro del tráfico capturado. No obstante, dado que también podemos almacenar las capturas después de realizarlas, desde el menú *File/Save*, no vamos a hacer uso de esta posibilidad por el momento.

Otras opciones que esta ventana nos permite detallar están relacionadas con la visualización en pantalla de la captura. Podemos optar por ver en tiempo real los paquetes que se van capturando y también podemos elegir que se realice un desplazamiento vertical automático de la pantalla (*scrolling*).

Finalmente, con el fin de terminar la captura, podemos indicar que ésta termine automáticamente cuando se hayan capturado cierto número de paquetes, o se haya capturado una cantidad determinada de Kbytes, o bien cuando haya transcurrido cierta cantidad de tiempo. En caso de no seleccionar ninguna de las opciones, tendremos que finalizar la captura de forma manual. Para iniciar la captura presionaremos el botón *Start*.

3. Filtros de captura y de pantalla

Al intentar analizar el tráfico de cualquier red, en particular el de la red de la UPV, resulta habitual encontrarse gran cantidad de paquetes que emplean protocolos en los que no estamos interesados. Tal cantidad de tráfico dificulta el análisis de los paquetes

capturados y aumenta innecesariamente el tamaño de los ficheros de captura, por lo que se hace indispensable filtrar toda esa información.

Para filtrar los paquetes y facilitar el análisis del tráfico capturado podemos usar dos alternativas, no excluyentes entre ellas. La primera es definir un **filtro de captura**, de modo que el propio *WireShark*, cuando llega un nuevo paquete, decide si ese paquete se ajusta o no a los criterios establecidos por el filtro. En caso de que se ajuste, el paquete es aceptado y mostrado en pantalla, mientras que en caso contrario el paquete se descarta.

La otra alternativa para filtrar la información de los paquetes es establecer un **filtro de pantalla**. En este caso lo habitual es capturar todos los paquetes que circulan por la red, sin restricción alguna, y especificar un filtro para poder extraer entre todo ese tráfico capturado aquellos paquetes que nos interesan. En cualquier caso, es posible usar un filtro de captura y posteriormente, sobre los paquetes capturados, usar un filtro de pantalla para ver mejor los detalles que estemos buscando en cada momento.

Los filtros de captura se pueden especificar desde la ventana de captura (*Capture Options*). Podemos especificar un filtro escribiendo la expresión correspondiente en la caja de texto situada junto al botón *Filter*. La sintaxis de estas expresiones es la misma que la usada en la orden **tcpdump**, disponible habitualmente en los sistemas Unix y Linux. En el apartado siguiente se mostrará un resumen de esta sintaxis.

Por otra parte, los filtros de pantalla se pueden especificar desde la parte inferior de la ventana principal de *WireShark*. La sintaxis para este tipo de filtros es ligeramente diferente. Es posible disponer de un asistente para especificar estos filtros.

Es conveniente tener cuidado con estos filtros de visualización, ya que pueden, en ocasiones, llevar a error. Por ejemplo, estos filtros se emplearán más adelante en TCP de forma automática para seguir un flujo TCP. Por tanto, para volver a visualizar todos los paquetes capturados es necesario limpiar este filtro de visualización, bien con la opción correspondiente (*clear filter*), o bien empleando un filtro vacío (línea en blanco).

3.1 Expresiones de filtros de captura

Para seleccionar qué paquetes serán capturados se emplea una expresión de filtro, de forma que el paquete será almacenado si cumple con los criterios del filtro (la expresión se evalúa a ***true***). Las expresiones de filtro se emplean siempre en minúsculas, y consisten en una o varias primitivas conectadas mediante operadores lógicos (and, or, not). Cada una de estas primitivas constan de un identificador precedido, al menos, de alguno de los siguientes tres tipos de calificadores:

De tipo: Indican a qué hace referencia el identificador

1. **Computadores concretos (*host*)**. Por ejemplo, el filtro:

```
host 158.42.180.62
```

captura todas las tramas dirigidas o procedentes de esa dirección IP.

2. **Redes concretas (*net*).** Por ejemplo, el filtro:

```
net 158.42
```

captura todas las tramas dirigidas o procedentes de dicha red IP.

3. **Puertos determinados (*port*)**. Así, el filtro:

```
port 7
```

captura todas las tramas dirigidas o procedentes del puerto 7 de cualquier computador, tanto TCP como UDP.

Si no se especifica el tipo se asume el tipo *host*. También se encuentra especificado el identificador *broadcast*, que permite hacer referencia a las direcciones de difusión.

De dirección: Especifican si el identificador debe entenderse sobre el origen, el destino o ambos. Los calificadores de dirección posibles son `src`, `dst`, `src or dst`, y `src and dst`. Se aplicarán sobre alguno de los calificadores de tipo. Ejemplos:

```
src 158.42.181.18
src port 25
src or dst net 158.42
```

De protocolo: Indican un tipo de protocolo. En nuestro caso resultan de interés los protocolos `arp`, `ip`, `icmp`, `tcp`, o `udp`. Como norma general, no existen calificadores de nivel de aplicación, debiendo emplearse el número de puerto y protocolo de transporte para capturar el tráfico correspondiente a un protocolo de aplicación concreto.

Se pueden combinar varias primitivas mediante los conectores `and` y *`or`*. También es posible usar el operador `not`.

Ejemplo: `udp and dst 158.42.148.3`

Esto mismo es aplicable tambiém a los identificadores.

Ejemplo: `dst 158.42.181.4 or 158.42.181.15`

También se puede hacer uso de paréntesis para indicar las precedencias deseadas.

Ejemplo: `dst 158.42.181.4 and (udp or icmp)`

De momento, para la práctica 1 sólo se requiere el uso de los filtros port y host. Posteriormente, a lo largo del curso ya iremos profundizando en el uso de filtros más complejos. Para acceder a la documentación detallada acerca de los filtros y sus posibilidades se puede acudir al manual en línea de la orden **tcpdump**, tecleando en una consola de texto la orden `man tcpdump`.

Tras completar esta lectura y hacer algunas pruebas debes enviar la captura solicitada como ejercicio al profesor.

Ejercicio:

- Aplica el filtro de captura “port 53”. Con este filtro deberían capturarse los accesos que realices al DNS para realizar la traducción de los nombres de dominio de los servidores a sus direcciones IP.
- Activa el inicio de una captura.
- Para generar tráfico accede a una página web (que no tengas en la caché del navegador). Ten en cuenta que podrías tener el nombre ya resuelto en tu caché local DNS y entonces no aparecería nada en la captura, porque no se consultaría al servidor DNS. Si ocurre esto prueba a acceder a otra página web en otro servidor distinto.
- Una vez hayas conseguido los resultados, para de capturar y analiza el encapsulamiento de un paquete. Comprueba el direccionamiento a distintos niveles (hardware, IP, números de puerto).
- Debes enviar como resultado del ejercicio la captura realizada, indicando las direcciones IP fuente y destino y los números de puerto fuente y destino.