Práctica 3

Observando la red

1. Objetivos

El objetivo principal que se persigue en esta práctica es ser capaz de observar el tráfico de red mediante un analizador de protocolos como Wireshark y comprender los conceptos básicos de uso de unos protocolos sobre otros.

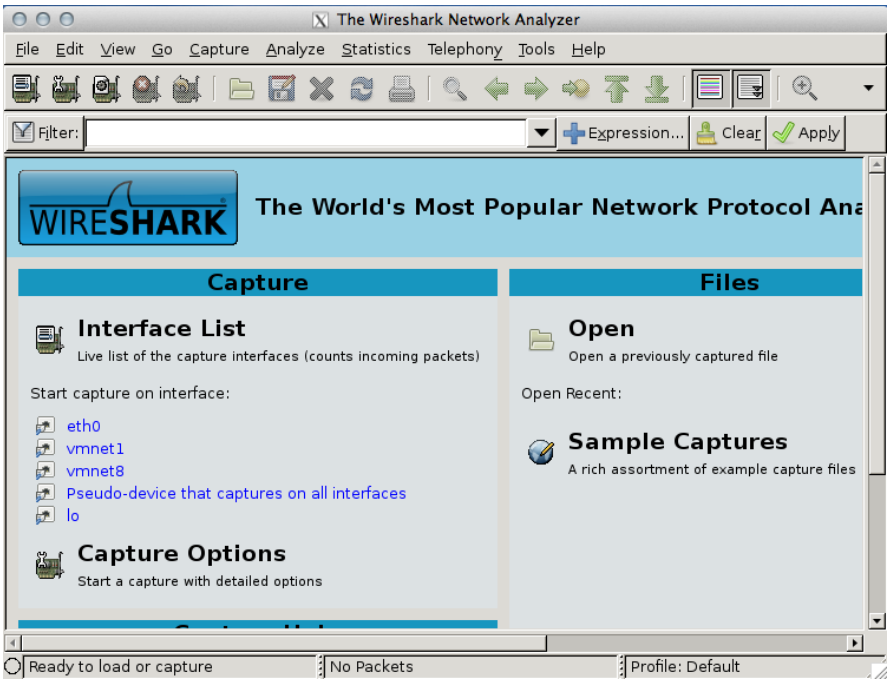
2. Usando Wireshark

Lanza el programa `wireshark` desde el menú aplicaciones o con el comando:

```
$ wireshark &
```

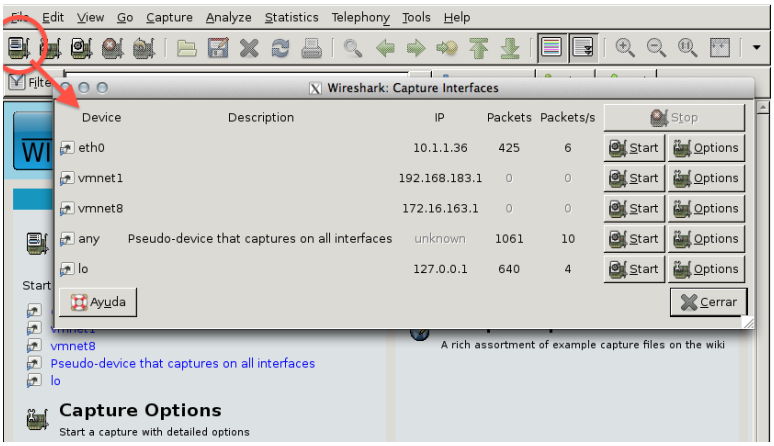
El programa se lanzará y abrirá una ventana. Wireshark es un analizador de protocolos. Su función principal es mostrar los paquetes observados en un interfaz y ayudar a diseccionarlos identificando cada cabecera de protocolo y los campos con información incluidos en la cabecera. Puede hacer esto directamente observando la red (o las redes) a las que está conectado el ordenador pero también permite grabar los paquetes que ha visto en un fichero para su posterior análisis. A estas grabaciones las llamamos normalmente ficheros de captura o trazas. Wireshark también es capaz de abrir un fichero de traza previamente grabado aunque sea en otra máquina y hacer su análisis sobre los paquetes observados en la grabación.

La pantalla inicial de Wireshark es muy similar a la que puedes ver en la siguiente imagen.



Lo primero que vamos a hacer va a ser realizar una nueva captura de la red. Para ello debes elegir en cuál de los interfaces que unen un ordenador a la red quieres capturar. Observa la lista de interfaces disponibles eligiendo el primer icono de arriba. ¿Tu ordenador tiene varios interfaces de red? Aunque es posible tener varias tarjetas de red y estar conectado a varias redes, en el caso del laboratorio la mayoría de los interfaces que ve son virtuales y no representan en realidad una salida a una red física.

El interfaz `eth0` es el correspondiente a la tarjeta Ethernet de tu ordenador. Puedes observar el cable por detrás que lo une al punto de red de la mesa. Eso es `eth0`. Del resto de los interfaces que ves, el indicado como `lo` es el llamado interfaz de *loopback* que sirve para que programas de este ordenador puedan hablarse entre sí usando protocolos de red aunque el ordenador esté desconectado o no posea un interfaz físico. El interfaz `any` no es un interfaz sino la manera en que Wireshark permite observar todos a la vez. En está práctica queremos observar la red Ethernet del laboratorio así que elige siempre `eth0`.



Una vez elegido el interfaz a usar, podemos capturar directamente sin pensar mucho (pulsando *Start*) o bien configurar algunas opciones en la captura (pulsando *Options*). Esto mismo se podía hacer desde las opciones de la pantalla inicial o con los iconos de arriba que dejan iniciar captura rápido desde el ultimo interfaz seleccionado o ir directamente a las opciones.

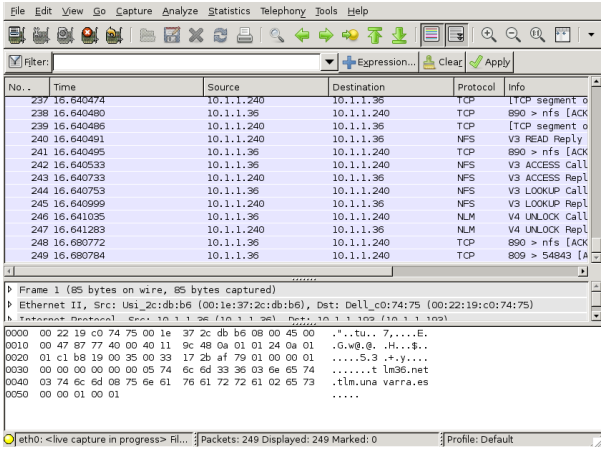
Elige *Options* en el interfaz `eth0` (o el icono *Capture Options*). Aquí puedes configurar algunas cosas de interés. Las que no se explican puedes dejarlas como están.

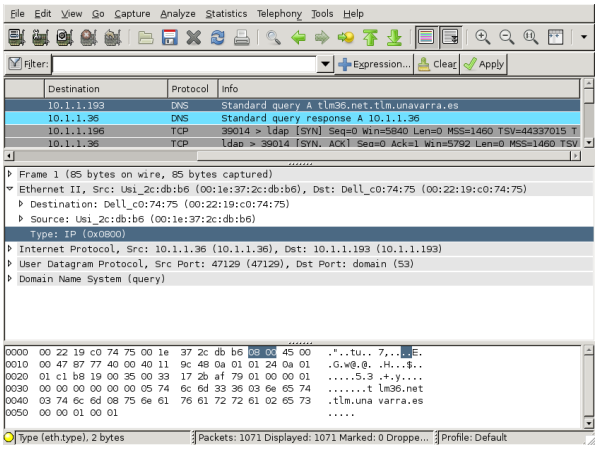
- *Interface*: permite elegir el interfaz. Disponemos de los mismos que hemos visto anteriormente. Elige `eth0` si no está ya seleccionado
- *Link-layer header type*: permite elegir el tipo de trama que se envía en ese interfaz. En nuestro caso es *Ethernet*
- *Capture packets in promiscuous mode*: el modo promiscuo indica si queremos capturar todas las tramas que se vean en ese interfaz o solo las que haya enviado o vayan dirigidas a este ordenador. De momento vamos a ver todas

las disponibles, con lo que activaremos esta casilla
- *Limit each packet to xxx bytes*: se puede limitar que no se capturen paquetes enteros sino sólo el principio de cada uno para ahorrar memoria y espacio en disco ya que es posible que lo que queramos ver son sólo las cabeceras, siendo un tanto indiferente el contenido. De momento no capturaremos muchos paquetes así que no indicaremos límite
- *Capture filter*: permite decidir de una manera flexible qué paquetes queremos capturar y cuáles no. Es una expresión de texto en un lenguaje de reglas que pone condiciones a todos los paquetes se reciben. Si un paquete cumple la regla se captura y se muestra o se guarda y si no la cumple se descarta. En esta primera prueba asegúrate de que el campo está vacío, lo que significa que queremos capturarlos todos
- Deja el resto de opciones como están y pulsa *Start*

Wireshark empezará a capturar paquetes y mostrar una lista de los que ha capturado. En cuanto haya unos cuantos detén la captura pulsando *Stop*.

Si seleccionas un paquete de la lista puedes ver detalles sobre ese paquete en los paneles inferiores. Para ver la información un poco más clara, desactiva los colores pulsando el botón *colorize packets*. Abajo del todo se ve el contenido del paquete completo y en el medio el análisis que hace Wireshark del contenido del mismo. Podemos desplegar cada una de las cabeceras y seleccionar los campos de cada cabecera de modo que en el panel inferior se mostrará dónde está situado ese campo.





El nivel inferior del análisis, *Frame*, muestra los datos de la captura del paquete y no es propiamente una cabecera. Dentro de *Frame* verás tramas *Ethernet* y dentro de las tramas *Ethernet* veras paquetes IP o ARP. IP a su vez puede transportar paquetes TCP o UDP. Por ejemplo elige un paquete cualquiera y observa con ayuda del análisis dónde están situadas las direcciones origen y destino de *Ethernet*. Observa también cómo los paquetes de diferentes tipos IP o ARP tienen diferente valor en el campo *Type*.

Busca en tu captura paquetes que transporten protocolos TCP o UDP sobre IP y observa el encapsulado de unos paquetes dentro de otros.

Utiliza el comando `ifconfig` para averiguar la dirección de tu interfaz *Ethernet*. Puedes observar la dirección MAC (*direcciónHW*) y la dirección IP (*Direc. Inet*).

```
$ ifconfig eth0
eth0      Link encap:Ethernet  direcciónHW 00:1e:37:2c:db:b2
          Direc. inet:10.1.1.26  Difus.:10.1.255.255  Másc:255.255.0.0
            ...
```

Utiliza la dirección IP para localizar un paquete enviado por tu ordenador. Fíjate que en la lista de paquetes puedes aplicar un filtro para seleccionar algunos. Esto se llama *display filter* y se escribe en un lenguaje de reglas. Puedes editar reglas y añadirlas pulsando en el botón *expression* (tendrás que elegir en la lista los protocolos sobre los que aplicar las reglas, busca *Ethernet* por ahí). Por ejemplo puedes ver solo los paquetes que tengan una dirección IP o una dirección MAC concreta eligiendo reglas como estas (averigua qué hacen). Pulsa en el botón *apply* para aplicar el filtro.

```
eth.addr==00:1e:37:2c:db:b2   (pon tu dirección MAC)

eth.src==00:1e:37:2c:db:b2    (pon tu dirección MAC)

ip.src==10.1.1.26             (pon tu dirección IP)
```

En este paquete examina la cabecera de *Ethernet* para comprobar las direcciones MAC origen y destino de la trama *Ethernet*. Comprueba que la dirección MAC de origen coincide con la de tu ordenador (que puedes ver también haciendo `ifconfig` en un terminal). Observa que las direcciones IP origen y destino del paquete aparecen en la cabecera de IP que va dentro de la trama *Ethernet*. Borra el filtro de *display* con *clear* para volver a observar todos los paquetes.

Guarda la traza de paquetes capturados en disco para su posterior análisis, usa el formato *wireshark/tcpdump libpcap*. Observa que puedes guardar solo los paquetes seleccionados o los que cumplen el *display filter* si es necesario. Cierra Wireshark y prueba a volver a abrirlo y abrir el fichero para volver a examinar la traza.

Por ultimo, en esta visión general del uso de Wireshark, vuelve a iniciar el diálogo de opciones para capturar. Elige el interfaz `eth0` y fíjate en el filtro de captura. Puedes indicar a Wireshark que no capture todos los paquetes que vea sino que sólo elija algunos. Esto se llama *capture filter*. Consiste también en una serie de condiciones en un lenguaje de reglas. Por razones históricas este lenguaje no es el mismo que el del *display filter*. Por ejemplo las reglas de antes en *capture filter* son:

```
ether host 00:1e:37:2c:db:b2

ether src 00:1e:37:2c:db:b2

ip src 10.1.1.26
```

Pulsando el botón *capture filter* dispones de algunos ejemplos más de reglas pregrabadas.

3. Observando y localizando el tráfico

Una vez se tiene un control básico de Wireshark observemos el tráfico en la red. La idea es poner a Wireshark a capturar los paquetes que intercambia tu maquina con otro ordenador concreto utilizando un *capture filter*. No deberías observar demasiados paquetes de forma que puedes dejar la captura en tiempo real y observar lo que va apareciendo. Para ello obtén la dirección MAC de otro ordenador de tu mesa del laboratorio y realiza una captura que vea solo el tráfico entre estos ordenadores con un *capture filter*. Seguidamente tienes dos opciones, prueba ambas y decide cuál es la correcta. Las dos significan cosas diferentes y deberías ser capaz de interpretar la diferencia.

```
ether src midireccionMAC and ether dst direccionMACdelOtro
ether host midireccionMAC and ether host direccionMACdelOtro
```

Una vez que estés capturando tráfico, haz un `ping` al otro ordenador, lo que le enviará paquetes al otro para obtener respuestas, verificando que está funcionando como se vio en la práctica anterior.

```
$ ping –c 2 ordenadorvecino     (con –c sólo se envían 2 paquetes)
```

Observa los paquetes que aparecen en la red como resultado del `ping`. Seguidamente haz algo que genere intercambio de información entre esos dos ordenadores como utilizar un `ssh` para obtener un acceso remoto de uno en el otro y observe el trafico que genera.

```
$ ssh ordenadorvecino
```

Prueba en el menú S*tatistics* la opción *Summary* que resume los parámetros principales de una captura. Prueba también la descomposición del tráfico en protocolos (*Protocol Hierarchy*) para ver cuántos paquetes y qué porcentaje de la captura corresponde a cada protocolo.

Prueba también que puedes realizar gráficos de los parámetros en función del tiempo usando *IO Graphs* y utilízalo para ver cuántos paquetes por segundo y bits por segundo está generando. Observa que en la gráfica puedes elegir el intervalo de tiempo en el que van a promediarse las medidas (*tick interval*). Con eso puedes calcular por ejemplo el *throughput* (Mbps) en intervalos de 1s o de 10s.

Puedes usar estas estadísticas sobre una captura en tiempo real o sobre un fichero ya capturado.

Puedes probar esto también con un ordenador externo. Para ello elige una pagina web que no sea google. Utiliza el comando `host` para obtener la dirección IP del servidor. Por ejemplo:

```
$ host www.tlm.unavarra.es
www.tlm.unavarra.es is an alias for pluto.tlm.unavarra.es.
pluto.tlm.unavarra.es has address 130.206.164.68
```

Lanza Wireshark capturando sólo los paquetes que vayan entre tu maquina y el servidor. Para ello también puedes poner condiciones sobre protocolos que no sean Ethernet por ejemplo la dirección IP del servidor

```
ether src midireccionMAC and ip dst direccionIPservidor
ether host midireccionMAC and ip host direccionIPservidor
```

Una vez esté Wireshark capturando pide una pagina web al servidor y observa los paquetes que aparecen debido a la petición web. Prueba a buscar el contenido de la petición y de la pagina pedida en los paquetes capturados.

4. Analizando una captura en fichero

Descarga el fichero `p3_capture.cap` de la página web de la asignatura. Contiene una captura realizada con Wireshark realizada en otro momento. La captura se ha realizado en la maquina con dirección MAC `00:1e:37:2c:db:b2`. Analízala para responder a las siguientes preguntas

- ¿Cuántos paquetes capturados hay en el fichero?
- ¿Durante cuánto tiempo se ha estado capturando esa traza?
- ¿Cuál es la velocidad media de captura? ¿Cuál es el tamaño medio del paquete capturado?
- ¿En que instante de tiempo aparece por primera vez un tráfico sostenido de mas de 5Mbps?
- ¿Durante cuánto tiempo se mantiene un tráfico sostenido de mas de 7Mbps?
- ¿El primer paquete de la traza ha sido enviado o recibido por la maquina en la que se ha obtenido la captura?
- ¿En el paquete número 2000 cuál es la dirección MAC de origen y destino?
- ¿Cuántos paquetes IP hay en la captura? ¿Cuántos paquetes TCP? ¿Cuántos UDP?
- ¿Cuál es la velocidad media en paquetes por segundo de la traza?
- ¿A qué velocidad aproximada se han enviado datos hacia la dirección MAC `00:1e:37:2c:dd:04`?
- ¿Durante cuánto tiempo ha recibido datos a esa velocidad la dirección MAC 00:1e:37:2c:dd:04?
- ¿Las direcciones IP `10.1.1.24`, `10.1.1.25`, `10.1.1.26` aparecen en la traza?

5. Analizadores de tráfico de línea de comandos [opcional]

Wireshark es fácil y cómodo de utilizar pero en ocasiones queremos capturar rápidamente en línea de comandos simplemente para ver el contenido de un paquete o si hay determinado tipo de tráfico. Para ello existen programas capaces de capturar trafico en línea de comandos.

Prueba los programas `tcpdump` y `tshark` para mostrar los paquetes que se vean en el

interfaz `eth0` de tu maquina. `tcpdump` es el clásico, `tshark` es la herramienta de línea de comandos de `wireshark`. Los dos son similares y se basan en la librería de captura de tráfico llamada `libpcap`. Prueba al menos estas opciones:

```
tcpdump –i eth0        # capturar paquetes en el interfaz eth0
tshark –i eth0         # capturar paquetes en el interfaz eth0
tshark –V –i eth0      # ídem y mostrar los detalles de cada paquete
```

También puedes usar un *capture filter* indicándolo al final de la línea de comandos:

```
tshark –i eth0 ether host 00:1e:37:2c:db:b2
tcpdump –i eth0 ether host 00:1e:37:2c:db:b2
```

Mostrar el contenido de los paquetes:

```
tcpdump –i eth0 –XX ether host 00:1e:37:2c:db:b2
tshark –i eht0 –x ether host 00:1e:37:2c:db:b2
```

Grabar y cargar ficheros:

```
tcpdump –i eht0 –w captura.cap   # guarda los paquetes en captura.cap
tcpdump –r captura.cap    # lee de captura.cap en lugar del capturar
```