**Práctica 2: Introducción**

**WireShark**



***Analizador de protocolos***

***Curso 2023 - 2024***

**Introducción a Redes de Computadoras**

Docente: Roberto Olea

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**OBJETIVOS DE LA PRÁCTICA**

Esta práctica tiene como objetivo que el alumnado conozca las funcionalidades básicas de un analizador de protocolos y sea capaz de utilizarlas para analizar algunos protocolos sencillos.

**Introducción**

Un analizador de protocolos es una herramienta que sirve para desarrollar y depurar protocolos y aplicaciones de red. Permite al ordenador capturar diversas tramas de red para analizarlas, ya sea en tiempo real o después de haberlas capturado. Por analizar se entiende que el programa puede reconocer que la trama capturada pertenece a un protocolo concreto (TCP, ICMP...) y muestra al usuario la información decodificada. El analizador de protocolos que utilizaremos en esta práctica es **Wireshark**.

Wireshark es utilizado para analizar y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver de la información capturada on-line o salvada en disco.

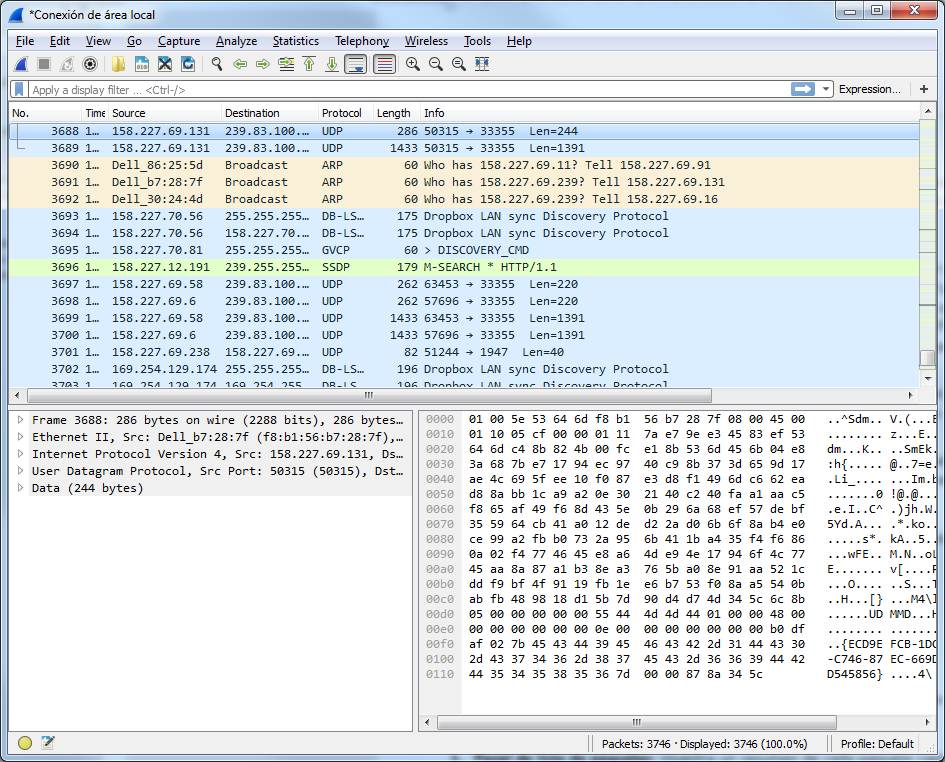
Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos Unix y compatibles, Mac OS X, así como en Microsoft Windows.

Descarga programa en: <http://www.wireshark.org/download.ht>ml

**PARTE 1: PRIMEROS PASOS CON WIRESHARK**

En esta primera figura 1 se muestra la pantalla principal del programa. Se pueden distinguir tres paneles diferentes:

1. **Panel de lista de paquetes**: muestra un resumen de cada paquete capturado. Pulsando sobre los paquetes de este panel se controla el contenido de los otros dos paneles inferiores.
2. **Panel de vista en árbol:** muestra el paquete seleccionado en el panel superior (1) con más detalle, permitiendo acceder a los distintos niveles de protocolos. Al pulsar sobre cada uno de los niveles se resaltan los datos del paquete correspondientes a dicho nivel en el panel inferior (3).
3. **Panel de detalle de los datos:** muestra el contenido del paquete seleccionado en el panel superior (1) en formato hexadecimal y ASCII.



**Panel de detalle de los datos**

**Panel de lista de paquetes**

**Panel de vista en árbol** **de paquetes**

*Figura 1 - Pantalla principal del programa WireShark*

Además de los tres paneles principales, se pueden destacar **la barra para trabajar con filtros**, disponibles en la parte superior de la pantalla de WireShark:



**Texto del filtro**

**Aplicar filtro**

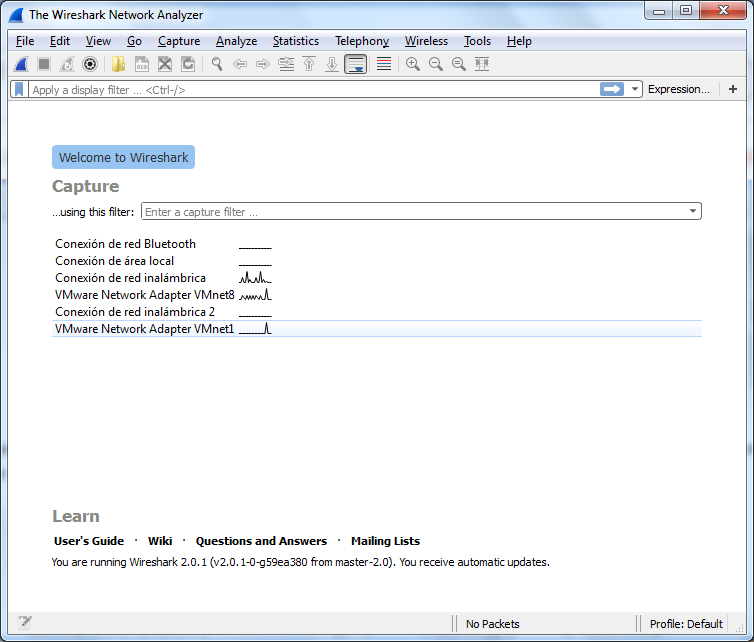
**Administrar filtros**

**Desactivar filtro**

*Figura 2. Barra para trabajar con filtros*

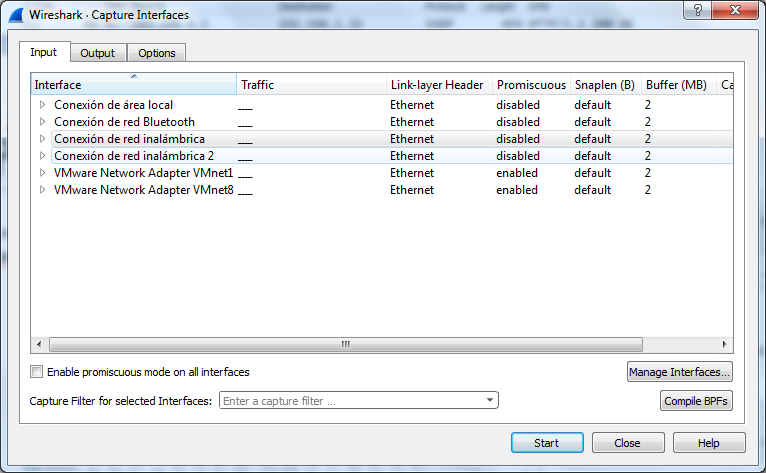
**Captura de datos**

Se comienza seleccionando la red en la que se quiere realizar la captura de datos. *Si el ordenador dispone de módem o tiene varias tarjetas de red, se tendrá que seleccionar la que se desea analizar. Veremos una pantalla similar a la siguiente*



*Figura 3. Pantalla de Inicio*

Una vez seleccionado la interface, se puede cambiar y modificar las opciones de la captura seleccionando del menú **Capture > Options**. Al hacerlo aparece el siguiente cuadro de diálogo.

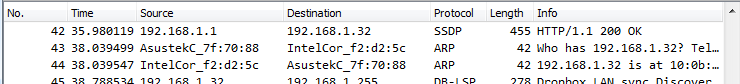


*Figura 4. Capture Options*

Para detener la captura de datos se usa la opción del menú **Capture > Stop.** Tanto para el inicio como la finalización de la captura se pueden utilizar los botones de la barra de menú.

Se muestran los paquetes según se van capturando. Se puede desactivar esta opción desactivando las “**Update list of packets in real time**", que se encuentra en la pestaña **Options** del cuadro de diálogo de opciones de captura.

**Información sobre paquetes capturados**

****

*Figura 5. Información de paquetes capturados en ventana principal*

En el panel de **lista de paquetes** la ventana principal, la información que aparece sobre los paquetes capturados es:

* **Nº** **-** El índice del paquete capturado
* **Time -** El tiempo que ha transcurrido desde el comienzo de la captura hasta que se ha capturado este paquete
* **Source -** La IP del origen del paquete
* **Destination -** La IP destino del paquete
* **Protocol** **-** El protocolo utilizado para mandar el paquete
* **Info** **-** Pequeña descripción del paquete

**Filtrado de Información Capturada**

Dada la enorme cantidad de información que circula por una tarjeta de red mientras está funcionando, resulta especialmente útil poder limitar de alguna manera dicha información y poder concentrarse en el análisis del tráfico de red concreto que interese estudiar.

WireShark ofrece varias posibilidades de **filtrado de información**: filtrado por protocolo, filtradode captura y **filtrado en la presentación de la información.** Nos ocuparemos de este último únicamente.

**Filtrado de Presentación**

Se puede definir un filtro mediante el cual se muestran en el panel principal (1) aquellos paquetes de datos que interesa analizar, ocultando el resto.

Para utilizar un filtro de presentación se puede escribir su expresión directamente en el recuadro de texto del filtro y aplicarlo (figura 2). Existe otra alternativa, que consiste en pulsar el botón de administrar filtro, entonces aparece el cuadro de diálogo con filtros ya definidos.

|  |
| --- |
| **ESTA PRÁCTICA SE DEBE REALIZAR EN LOS ORDENADORES DEL LABORATORIO**  **&&** : AND lógico (y)  **||** : OR lógico (ó)  **contains** : Contener  **==** : Igual  **!=** : Desigual  **Informe: Cuestión 1:** Experimenta con algunos filtros:  *dns: se muestran las azules*  *ip.src == XXX.XXX.XXX.XXX (pon tu IP) Saca los datos cuya fuente es tu IP*  *ip.dst == XXX.XXX.XXX.XXX (pon la IP del servidor de la página web de la EHU/UPV) \* Saca los datos cuyo destino sea esa IP*  *http.host ==* [*www.google.com*](http://www.google.com) *\**  *\* Seguramente, al aplicar esto filtros no se muestre ningún paquete. ¿Por qué? No olvides que para poder ver los paquetes de una conexión tiene que existir dicha conexión, es decir, se debe abrir la página web, por ejemplo.*  Utiliza el comando de red *ipconfig* para conocer la IP de tu PC y la puerta de enlace predeterminada. ¿Cómo puedes conocer la IP del servidor EHU/UPV? ¿Que nos permiten ver nuestro analizador con los filtros anteriores? nslookup www.ehu.eus  **Informe: Cuestión 2:** Define los filtros que:   * Presenten los paquetes cuya dirección IP de origen sea *host1* y su dirección IP de destino sea la *host2* (o viceversa, dos filtros diferentes)   ip.src == XXX (Nuestra IP) && ip.dst == YYY (Otra IP)  ip.src == YYY (Otra IP) && ip.dst == XXX (Nuestra IP)   * Capturar todo el tráfico cuyo origen y destino es el host *host1* y el host *host2*, ohost *host2* y el host *host1*, respectivamente (un único filtro)   (ip.src XXX (Nuestra IP) || ip.src YYY (Otra IP)) && (ip.dst YYY (Otra IP) || ip.src XXX (Nuestra IP))   * Visualiza todo el tráfico menos el *host1*   ip.src != XXX && ip.dst !=XXX  \*como *host1* y *host2* elige tu dirección y la de otro ordenador del aula. |
|  |

**PARTE 2: ANALIZANDO PROTOCOLOS CON WIRESHARK**

**1.-** Una vez conocido el entorno, se va a analizar el comando **ping** que vimos en la práctica anterior, en este caso vamos a analizar los paquetes reales no una simulación. Para ello

A.- Captura los paquetes de red cuando envías un ping a otro ordenador del aula.

B.- Filtra los paquetes para conseguir ver los paquetes del **protocolo ICMP** entre los ordenadores implicados (***filtro: ip.addr==host1 && ip.addr==host2 && icmp)*** (Igual debe ser modificado)

C.- **Informe: Cuestión 3:** Analiza la información y realiza una representación esquemática. ¿Es lo mismo que podíamos ver en la simulación con Packet Tracer?

Vemos que nos mandan 1 paquete, respondemos, mandan el 2º paquete, respondemos, mandan el 3º paquete, respondemos, mandan el 4º paquete, respondemos.

Vemos que mandamos 1 paquete, nos responden……

Vemos lo mismo que en Packet Tracer.

D.- **Informe: Cuestión 4:** Permite que se vean todos los paquetes. ¿Aparecen algún paquete del **protocolo ARP**? Analiza la información de este paquete e intentan explicar cuál es la funcionalidad de este protocolo.

No aparece ninguno entre nuestros dos ordenadores, pero sí si miramos sin filtros. El destino de los protocolos ARP siempre será broadcast.

Encontrar la dirección física a través de una dirección IP.

**2.- ¿Como funciona TRACEROUTE?**

**Traceroute** es una herramienta de diagnóstico de redes, presente en la mayoría de los sistemas operativos. Esta herramienta permite determinar la ruta efectuada por un paquete.

El comando **Traceroute** es algo diferente según el sistema operativo

* En los sistemas UNIX / Linux: **traceroute** nombre.destino
* En los sistema Windows : **tracert** nombre.destino

En general un **tracert** muestra las Trayectoria que le lleva a un paquete llegar hasta el destino que queremos comprobar.

Uso: tracert [-d] [-h saltos\_máximos] [-j lista\_de\_hosts] [-w tiempo\_de\_espera]

nombre\_destino

Opciones:

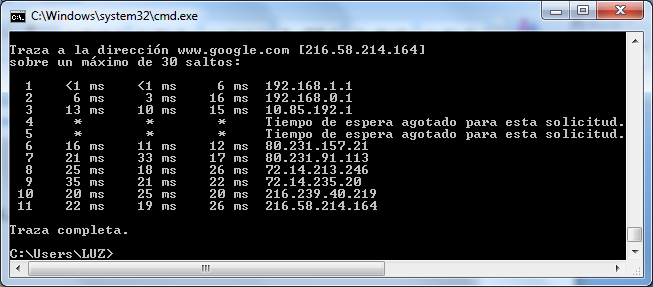
-d No convierte direcciones en nombres de hosts.

-h saltos\_máximos Máxima cantidad de saltos en la búsqueda del objetivo.

-j lista-de-host Enrutamiento relajado de origen a lo largo de la lista de hosts.

-w tiempo\_espera Cantidad de milisegundos entre intentos

Si ejecutamos este comando de red en nuestro ordenador obtenemos algo similar a esto:



*Figura 8.-Resultado ejecución Tracer-Route*

*(Si tienes curiosidad puedes consultar a quien pertenecen estas IPs, por ejemplo, en la web* [*https://bandaancha.eu/whois*](https://bandaancha.eu/whois) *)*

El número de la primera columna es el número de salto, las tres columnas posteriores son los tiempos de respuesta para los paquetes enviados (un asterisco indica que no se obtuvo respuesta) y finalmente el nombre o la dirección IP del nodo por el que pasa.

**Conceptos en los que se basa**:

**Tracert** utiliza el campo [**Time To Live**](http://es.wikipedia.org/wiki/Time_To_Live) (TTL) de la **cabecera** [**IP**](http://es.wikipedia.org/wiki/Protocolo_IP). Este campo sirve para que un paquete no permanezca en la red de forma indefinida. El campo **TTL** es un número entero al que se le resta 1 en cada nodo por el que pasa. De esta forma, cuando el campo TTL llega al valor 0 ya no se reenvía más, sino que el nodo que lo esté manejando en ese momento lo descarta.

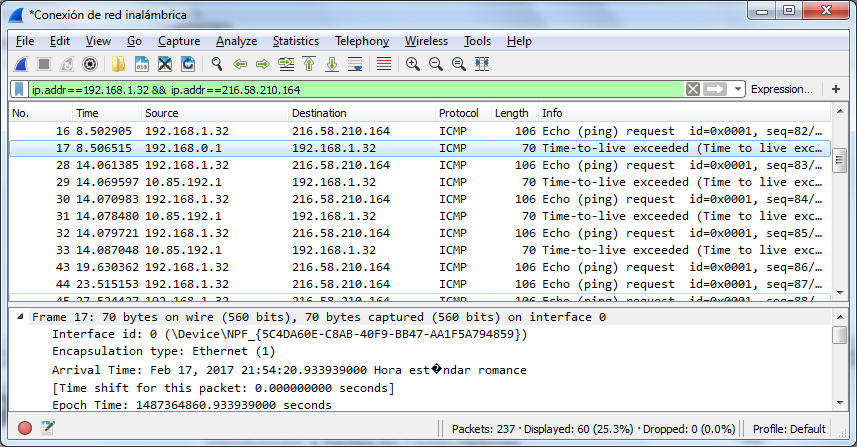
Cuando un nodo descarta un paquete, envía al emisor un mensaje de control especial **(ICMP-Internet Control Message Protocol)** indicando de la incidencia al nodo origen. El comando **Tracert usa esta respuesta para averiguar la dirección IP del nodo que desechó** el paquete.

1. Utiliza el comando **tracert** para conocer la ruta que tiene que seguir un paquete para alcanzar la dirección de [www.google.es](http://www.google.es) u otra dirección. Analiza los resultados obtenidos. ¿Por cuantos nodos intermedios pasa el mensaje?

Pasa por 10 nodos

1. Repite de nuevo el comando, pero ahora **captura con el analizador las tramas.**

**Lo hemos hecho**



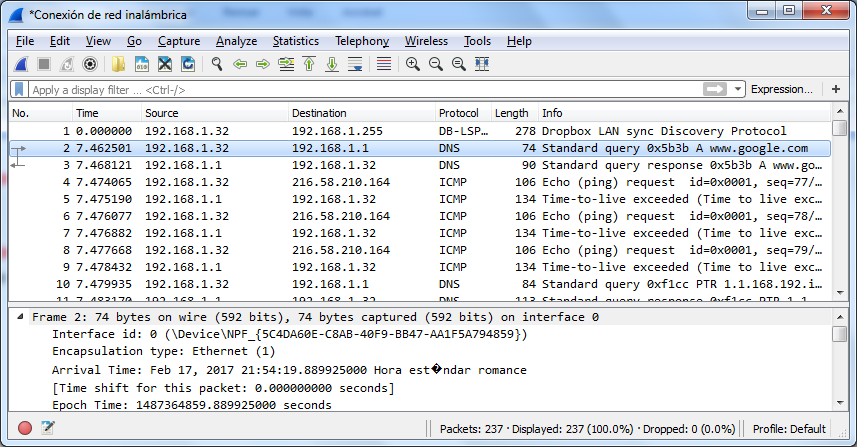
*Figura 8.-Captura Tracer-Router*

1. **Informe: Cuestión 5:** Analiza la captura realizada y explica cómo funciona el comando (recuerda utilizar filtros)

Siempre tendremos una de las dos direcciones IP en el Source o el Destination. De vez en cuando perderemos algún paquete, porque el tiempo de respuesta es muy largo.

Explicar el tracert

**Informe: Cuestión 6:** Analiza los paquetes del protocolo **DNS** que aparecen justo antes de los paquetes anteriores. ¿Qué función crees que tiene este protocolo? ¿Cuál es la dirección de tu servidor DNS? Describe un mensaje DNS (response), desglosando cada cabecera existente en él (y los campos más significativos de cada cabecera).



*Figura 8.- Paquetes DNS*