

# Practica 1. Redes 1

## VM y Wireshark (wireshark\_captura.pcap se corresponde con practipa1.pcap)

### Ejercicio 1:

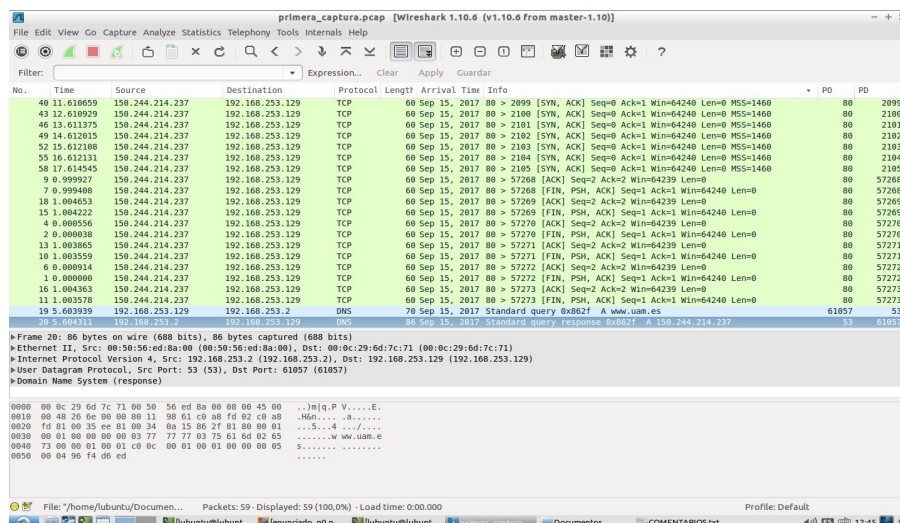
El proceso que hemos seguido es el siguiente:

- 1) Hemos ejecutado Wireshark y lo hemos configurado para se capture el tráfico de "eth0".
- 2) Iniciamos la captura de tráfico mientras que comenzamos a navegar para que pueda captar los paquetes que se intercambian y podamos analizar los resultados obtenidos.
- 3) En este caso introducimos por consola el comando "sudo hping3 -S -p 80 www.uam.es" por el cual accedemos a la página de la uam.
- 4) Paramos el tráfico y lo guardamos en un archivo .pcap.
- 5) Al volver a abrir la captura almacenada lo que observamos son los paquetes que hemos obtenido de la captura realizada; más concretamente nos vamos a fijar en las dos columnas que previamente hemos añadido:  
-PO: puerto de origen.  
-PD: puerto de destino.

En estas columnas observamos que el puerto número 53 aparece una única vez en cada columna (PD y PO). Este puerto se corresponde con la página que hemos buscado por terminal. Esto quiere decir que lo que estamos preguntando por debajo al ordenador es la dirección IP de 150.244.214.237; es más, si buscamos en la columna de puerto de destino el número 53 vemos que se corresponde con la dirección de [www.uam.es](http://www.uam.es).

Los problemas que nos hemos encontrado a lo largo de la realización del ejercicio no han sido muchos ni importantes.

Simplemente lo que nos costó al principio fue familiarizarnos con el programa y posteriormente entender lo que significaban cada uno de los datos que nos proporcionaba Wireshark.



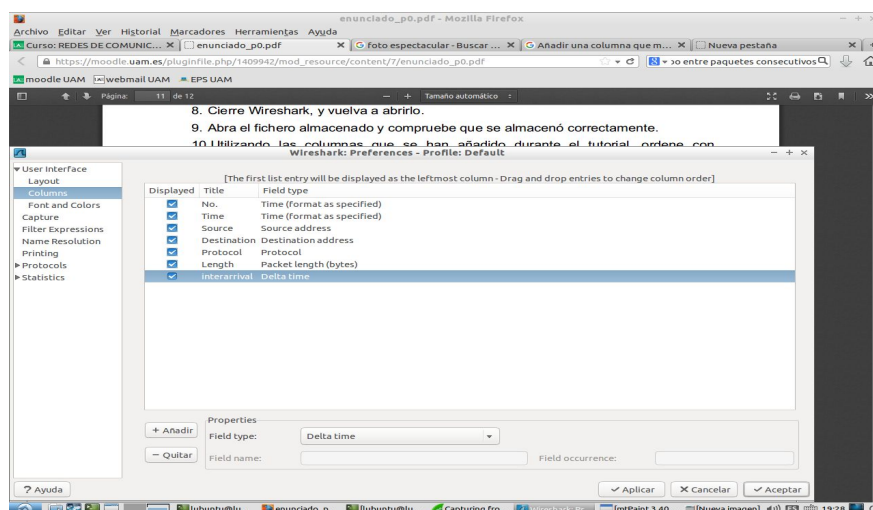
## Ejercicio 2:

En este ejercicio lo primero que hicimos fue generar una captura de tráfico mediante la visualización de páginas web. Posteriormente cuando nos pidieron que aplicásemos ciertos filtros a la captura realizada vimos que teníamos dos opciones. La primera de ellas la encontrábamos a la hora previa de hacer la captura (más concretamente en su configuración), donde se nos daba la oportunidad de aplicar directamente un filtro (por ejemplo ip), para que todo el tráfico capturado tuviera que cumplir la condición de ser una dirección IP y ya después una vez finalizada la captura escribir en el apartado de “Filter” lo siguiente: “frame.cap\_len>1000” de tal forma que sólo nos aparecieran los paquetes IP mayores de 1000 bytes (dato que apreciamos en la columna de “Length”). La segunda opción que vimos, y por la cual nos decantamos, es la de aplicar los dos filtros a la vez, escribiendo en el apartado de “Filter” lo siguiente: “ip && frame.cap\_len>1000”. A la hora de almacenar la captura con los filtros que nos pedían no habíamos la solución para que aparecieran los dos a la vez. Por lo que la solución que vimos era hacer una captura de tráfico normal y una vez hecha, al abrirla introducir los dos filtros. El tamaño de los paquetes IP en comparación con el tamaño del protocolo IP es una diferencia de 14 bytes en cada uno de los casos (eso sí sólo entre los que son direcciones IP). Esta diferencia se debe a que la cabecera de una dirección IP es siempre de 14 bytes independientemente del paquete.

## Ejercicio 3:

Para añadir una columna en la que se especifique el tiempo entre paquetes consecutivos cliqueamos al menú 'Edit'-'>'Preferences'. Aparecerá la ventana de edición de preferencias, en la que podremos configurar distintos aspectos de Wireshark. Entramos en el apartado 'User Interface'—>'Columns' nos aparecerá la opción de añadir unas columnas a las ya existentes. Elegimos esta opción y designamos a esta columna con el nombre de “interarrival”. A continuación, elegimos en “Field Time” la opción de “Delta Time” que es la que nos ofrece el valor requerido.

En esta imagen se ve cómo añadimos la columna de “interarrival”:



En esta imagen vemos cómo se ven los valores del interarrival:

No.	Time	Source	Destination	Protocol	Length	interarrival
0.000000	0.000000	192.168.203.130	216.58.204.131	TCP	74	0.000000
0.020077	0.020077	00:50:56:e2:ca:b6	ff:ff:ff:ff:ff:ff	ARP	60	0.020077
0.020096	0.020096	00:0c:29:5d:56:e6	00:50:56:e2:ca:b6	ARP	42	0.000019
0.020188	0.020188	216.58.204.131	192.168.203.130	TCP	60	0.000092
0.020203	0.020203	192.168.203.130	216.58.204.131	TCP	54	0.000015
0.020523	0.020523	192.168.203.130	216.58.204.131	TLSv1.2	571	0.000320
0.020756	0.020756	216.58.204.131	192.168.203.130	TCP	60	0.000233
0.030805	0.030805	216.58.204.131	192.168.203.130	TLSv1.2	222	0.019049
0.030824	0.030824	192.168.203.130	216.58.204.131	TCP	54	0.000019
0.040252	0.040252	192.168.203.130	216.58.204.131	TLSv1.2	141	0.000428
0.040406	0.040406	216.58.204.131	192.168.203.130	TCP	60	0.000154
0.040583	0.040583	192.168.203.130	216.58.204.131	TLSv1.2	998	0.000177
0.040683	0.040683	216.58.204.131	192.168.203.130	TCP	60	0.000108
0.219413	0.219413	216.58.204.131	192.168.203.130	TLSv1.2	1484	0.178730
0.219764	0.219764	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000351
0.219785	0.219785	192.168.203.130	216.58.204.131	TCP	54	0.000021
0.219867	0.219867	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000082
0.219876	0.219876	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000009
0.219882	0.219882	192.168.203.130	216.58.204.131	TCP	54	0.000006
0.219930	0.219930	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000048
0.219937	0.219937	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000007
0.219942	0.219942	192.168.203.130	216.58.204.131	TCP	54	0.000005
0.219986	0.219986	216.58.204.131	192.168.203.130	TLSv1.2	1334	0.000044
0.220365	0.220365	216.58.204.131	192.168.203.130	TLSv1.2	1514	0.000379
0.220378	0.220378	192.168.203.130	216.58.204.131	TCP	54	0.000013

## Ejercicio 4

Modificamos la columna del tiempo para que este se muestre en formato humano y formato UNIX. Para el formato humano establecemos el tiempo como UTC date and time, mostrándose así la fecha y hora actual. Para establecer el tiempo UNIX, segundos transcurridos desde el 1 de Enero de 1970, cambiamos la configuración de medida del tiempo. Para ello, vamos a View → Time Display Format → Seconds since 1970-01-01

En esta imagen vemos cómo añadimos el tiempo UTC:

Wireshark: Preferences - Profile: Default

User Interface

Layout

Columns

Font and Colors

Capture

Filter Expressions

Name Resolution

Printing

Protocols

Statistics

Properties

Field type: UTC date and time

Field name:

Field occurrence:

Apply Cancel Accept

En esta imagen vemos el tiempo unix de cada paquete en la columna:

No.	Time	Source	Destination	Protocol	Length	Info
402	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TLSv1.2	571	Client Hello
403	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	60	443 → 51296 [ACK] Seq=1 Ack=518 Win=64240 Len=0
404	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TLSv1.2	222	Server Hello, Change Cipher Spec, Hello Request, Hello
405	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	54	51296 → 443 [ACK] Seq=518 Ack=169 Win=38016 Len=0
406	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	141	Change Cipher Spec, Hello Request, Hello Request
407	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	60	443 → 51296 [ACK] Seq=169 Ack=605 Win=64240 Len=0
408	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	1514	TCP segment of a reassembled PDU
409	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TLSv1.2	579	Application Data
410	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	60	443 → 51296 [ACK] Seq=169 Ack=2598 Win=31088 Len=0
411	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	60	443 → 51296 [ACK] Seq=169 Ack=2598 Win=31088 Len=0
412	2017-09-26 19:33:19.908937000	216.58.211.238	192.168.203.130	TCP	918	Application Data, Application Data
413	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	54	51296 → 443 [ACK] Seq=2598 Ack=1033 Win=31088 Len=0
414	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	54	36276 → 443 [ACK] Seq=1 Ack=135 Win=31056 Len=0
415	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	119	Application Data
416	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
417	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36288 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
418	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36288 → 443 [ACK] Seq=1 Ack=172 Win=31056 Len=0
419	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	144	Application Data
420	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=259 Win=31088 Len=0
421	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	60	443 → 36276 [ACK] Seq=259 Ack=32 Win=64240 Len=0
422	2017-09-26 19:33:19.908937000	192.168.203.130	216.58.211.238	TCP	60	443 → 36276 [ACK] Seq=259 Ack=33 Win=64239 Len=0
423	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
424	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
425	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
426	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
427	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
428	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
429	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
430	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
431	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
432	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0
433	2017-09-26 19:33:19.908937000	64.233.166.189	192.168.203.130	TCP	54	36276 → 443 [ACK] Seq=1 Ack=169 Win=31088 Len=0

## Ejercicio 5

En este ejercicio lo que hicimos fue en la configuración previa a la captura de paquetes establecer como filtro que sólo fueran paquetes UDP los paquetes capturados. De esta forma al ejecutar el comando “sudo hping3 -S -p 80 [www.uam.es](http://www.uam.es).” en la terminal, todos los paquetes capturados por el programa Wireshark se ven que son UDP. Esto lo podemos ver en la ventana del medio, clickeando sobre “Internal Protocol” vemos que todos son del tipo especificado.

Vemos que todos los paquetes son DNS, cuyo protocolo interno es UDP.

The screenshot shows the Wireshark network protocol analyzer interface. At the top, the packet capture filter is set to 'udp'. The main pane displays a list of 125 captured packets, all of which are DNS queries. The details pane for the selected packet (No. 125) shows the following structure:

- Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
- Ethernet II, Src: VMware, 72:52:9d:00:0c:29, Dst: VMware, ed:8a:00:00:50:56:ed:8a:00
- Internet Protocol Version 4, Src: 192.168.253.229, Dst: 192.168.253.2
- User Datagram Protocol, Src Port: 28359, Dst Port: 53
- Domain Name System (query)
  - Standard query query 6x77f2 A encrypted-tbni.gstatic.com
  - Standard query response 6x77f2 A encrypted-tbni.gstatic.com A 216.58.210.174 NS ns2.google.com NS ns1.google.com NS ns4.google.com NS ns3.google.com
  - Standard query response 6x77f2 A encrypted-tbni.gstatic.com AAAA 2a00:1450:4003:800::200e NS ns1.google.com NS ns3.google.com NS ns4.google.com
  - Standard query response 6x77f2 A www.google-analytics.com CNAME www.google-analytics.l.google.com A 216.58.210.174 NS ns2.google.com NS ns1.google.com NS ns4.google.com
  - Standard query response 6x77f2 A www.google-analytics.com CNAME www.google-analytics.l.google.com AAAA 2a00:1450:4003:800::200e NS ns1.google.com NS ns3.google.com NS ns4.google.com
  - Standard query response 6x77f2 A www.youtube.com CNAME youtube-ui.l.google.com A 216.58.210.174 NS ns4.google.com NS ns3.google.com NS ns2.google.com
  - Standard query response 6x77f2 A www.youtube.com CNAME youtube-ui.l.google.com AAAA 2a00:1450:4003:800::200e NS ns3.google.com NS ns2.google.com NS ns4.google.com
  - Standard query response 6x77f2 A campusescencia.uam-csic.es A 150.244.214.206 NS ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A campusescencia.uam-csic.es SOA ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A www.edx.org A 150.244.214.206 NS ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A biblioteca.uam.es A 150.244.214.206 NS ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A biblioteca.uam.es SOA ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A posgrado.uam.es A 150.244.214.206 NS ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es
  - Standard query response 6x77f2 A posgrado.uam.es SOA ns1.chico.rediris.es NS ns2.chico.rediris.es NS ns3.chico.rediris.es NS ns4.chico.rediris.es

At the bottom, the packet bytes pane shows the raw data in hexadecimal and ASCII format.