

INFORME PRÁCTICA 0

Alejandro Santorum Varela - alejandro.santorum@estudiante.uam.es

David Cabornero Pascual - david.cabornero@estudiante.uam.es

Pr. REDES1 - Pareja 4

Universidad Autónoma de Madrid

15-10-2018

Contents

1	Introducción	2
2	Ejercicio 1	2
3	Ejercicio 2	4
4	Ejercicio 3	6
5	Ejercicio 4	6
6	Ejercicio 5	8
7	Comentarios finales	8

1 Introducción

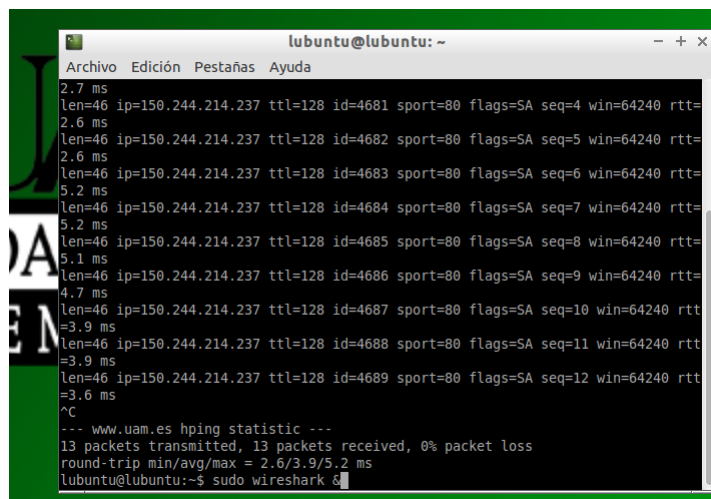
Nos encontramos en la práctica 1 de Redes de Comunicaciones I. En este documento se recogen los ejercicios de captura de tráfico con el programa Wireshark.

2 Ejercicio 1

En el primer ejercicio se nos pide abrir una terminal y Wireshark para capturar el tráfico. Una vez iniciada la captura, escribiremos en la terminal el comando especificado. Podemos observar el tráfico en la siguiente captura. v Podemos percibir que nuestro dispositivo le envía un paquete al servidor, y este nos devuelve otro. De esta forma podemos calcular el ping que existe entre ambos. Nuestro dispositivo tiene un identificador de puerto igual a 80, podemos observarlo en la columna PD cuando recibimos el paquete, o en la columna PO cuando lo enviamos. En cada envío, el servidor cambia de identificador, sumándole uno al anterior. En el pantallazo comienza en 2643, avanzando con los sucesivos envíos hasta 2648. Por otro lado también nos podemos fijar en la columna de "Source" y "Destination", donde nos muestran las direcciones de origen y destino de los paquetes, es decir, la dirección de nuestro equipo y la del servidor.

A continuación se nos pide guardar la traza de la captura y cerrar Wireshark, para volver abrirlo y comprobar que la traza se ha guardado correctamente:

Cerramos Wireshark y lo abrimos de nuevo:



```
lubuntu@lubuntu: ~  
Archivo Edición Pestañas Ayuda  
2.7 ms  
len=46 ip=150.244.214.237 ttl=128 id=4681 sport=80 flags=SA seq=4 win=64240 rtt=  
2.6 ms  
len=46 ip=150.244.214.237 ttl=128 id=4682 sport=80 flags=SA seq=5 win=64240 rtt=  
2.6 ms  
len=46 ip=150.244.214.237 ttl=128 id=4683 sport=80 flags=SA seq=6 win=64240 rtt=  
5.2 ms  
len=46 ip=150.244.214.237 ttl=128 id=4684 sport=80 flags=SA seq=7 win=64240 rtt=  
5.2 ms  
len=46 ip=150.244.214.237 ttl=128 id=4685 sport=80 flags=SA seq=8 win=64240 rtt=  
5.1 ms  
len=46 ip=150.244.214.237 ttl=128 id=4686 sport=80 flags=SA seq=9 win=64240 rtt=  
4.7 ms  
len=46 ip=150.244.214.237 ttl=128 id=4687 sport=80 flags=SA seq=10 win=64240 rtt=  
=3.9 ms  
len=46 ip=150.244.214.237 ttl=128 id=4688 sport=80 flags=SA seq=11 win=64240 rtt=  
=3.9 ms  
len=46 ip=150.244.214.237 ttl=128 id=4689 sport=80 flags=SA seq=12 win=64240 rtt=  
=3.6 ms  
^C  
--- www.uam.es hping statistic ---  
13 packets transmitted, 13 packets received, 0% packet loss  
round-trip min/avg/max = 2.6/3.9/5.2 ms  
lubuntu@lubuntu:~$ sudo wireshark &
```

Comprobamos que la traza de la captura anterior se ha guardado correctamente:

tutorial_traza1.pcap [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Guard

No.	Time	Source	Destination	Protocol	Length	Info	PO	PO
17	0.057926	150.244.214.237	192.168.182.128	TCP	60	88 > 2640 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2640
29	7.057957	150.244.214.237	192.168.182.128	TCP	60	88 > 2647 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2647
26	6.057329	150.244.214.237	192.168.182.128	TCP	60	88 > 2646 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2645
21	5.056748	150.244.214.237	192.168.182.128	TCP	60	88 > 2645 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2645
18	4.056518	150.244.214.237	192.168.182.128	TCP	60	88 > 2644 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2644
15	3.056314	150.244.214.237	192.168.182.128	TCP	60	88 > 2643 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2643
12	2.055699	150.244.214.237	192.168.182.128	TCP	60	88 > 2642 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2642
9	1.055026	150.244.214.237	192.168.182.128	TCP	60	88 > 2641 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2641
6	0.055095	150.244.214.237	192.168.182.128	TCP	60	88 > 2640 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	88	2640
4	0.090634	192.168.182.2	192.168.182.128	DNS	86	Standard query response 0xd919 A 156.244.214.237	53	37007
1	0.090690	192.168.182.128	192.168.182.2	DNS	70	Standard query 0xc919 A www.uan.es	37007	53
33	0.057994	192.168.182.128	156.244.214.237	TCP	54	2648 > 80 [RST] Seq=1 Win=0 Len=0	2648	80
31	0.056042	192.168.182.128	156.244.214.237	TCP	54	2648 > 80 [SYN] Seq=0 Win=512 Len=0	2648	80
30	7.058036	192.168.182.128	156.244.214.237	TCP	54	2647 > 80 [RST] Seq=1 Win=0 Len=0	2647	80
28	7.056124	192.168.182.128	156.244.214.237	TCP	54	2647 > 80 [SYN] Seq=0 Win=512 Len=0	2647	80
27	6.057395	192.168.182.128	156.244.214.237	TCP	54	2646 > 80 [RST] Seq=1 Win=0 Len=0	2646	80
25	6.055956	192.168.182.128	156.244.214.237	TCP	54	2646 > 80 [SYN] Seq=0 Win=512 Len=0	2646	80
22	5.056798	192.168.182.128	156.244.214.237	TCP	54	2645 > 80 [RST] Seq=1 Win=0 Len=0	2645	80
20	5.055026	192.168.182.128	156.244.214.237	TCP	54	2645 > 80 [SYN] Seq=0 Win=512 Len=0	2645	80
19	4.056593	192.168.182.128	156.244.214.237	TCP	54	2644 > 80 [RST] Seq=1 Win=0 Len=0	2644	80

Frame 1: 79 bytes on wire (596 bits), 79 bytes captured (560 bits) on interface II, src: 00:0c:29:cae:b:99 (00:0c:29:cae:b:99), dst: 00:50:56:f9:78:11 (00:50:56:f9:78:11)

Ethernet II, Src: 00:0c:29:cae:b:99 (00:0c:29:cae:b:99), Dst: 00:50:56:f9:78:11 (00:50:56:f9:78:11)

Internet Protocol Version 4, Src: 192.168.182.128 (192.168.182.128), Dst: 192.168.182.2 (192.168.182.2)

User Datagram Protocol, Src Port: 37007 (37007), Dst Port: 53 (53)

Domain Name System (query)

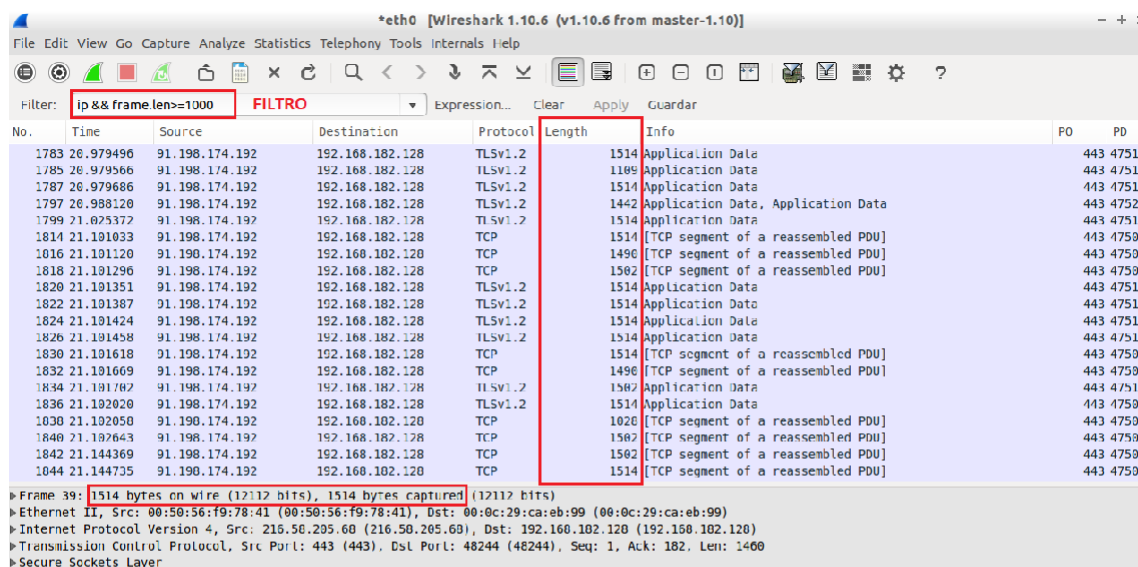
File: /home/tubuntu/Desktop/... Packets: 33 - Displayed: 33 (100.0%) - Load Time: 0:00.000 Profile: Default

Aprovechamos para ordenar los paquetes con respecto al campo 'PO' en sentido descendente y contabilizar el número de paquetes con el valor 53. En la imagen anterior es fácil comprobar que solo se contabiliza un paquete con 'PO' = 53.

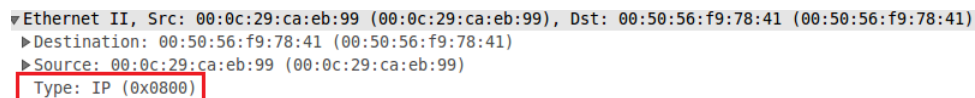
Por último, se pide comentar las dificultades que hemos tenido en la realización de este ejercicio, que no han sido muy grandes, solo hemos tenido que leer la documentación aportada de Wireshark para saber hacer este ejercicio.

3 Ejercicio 2

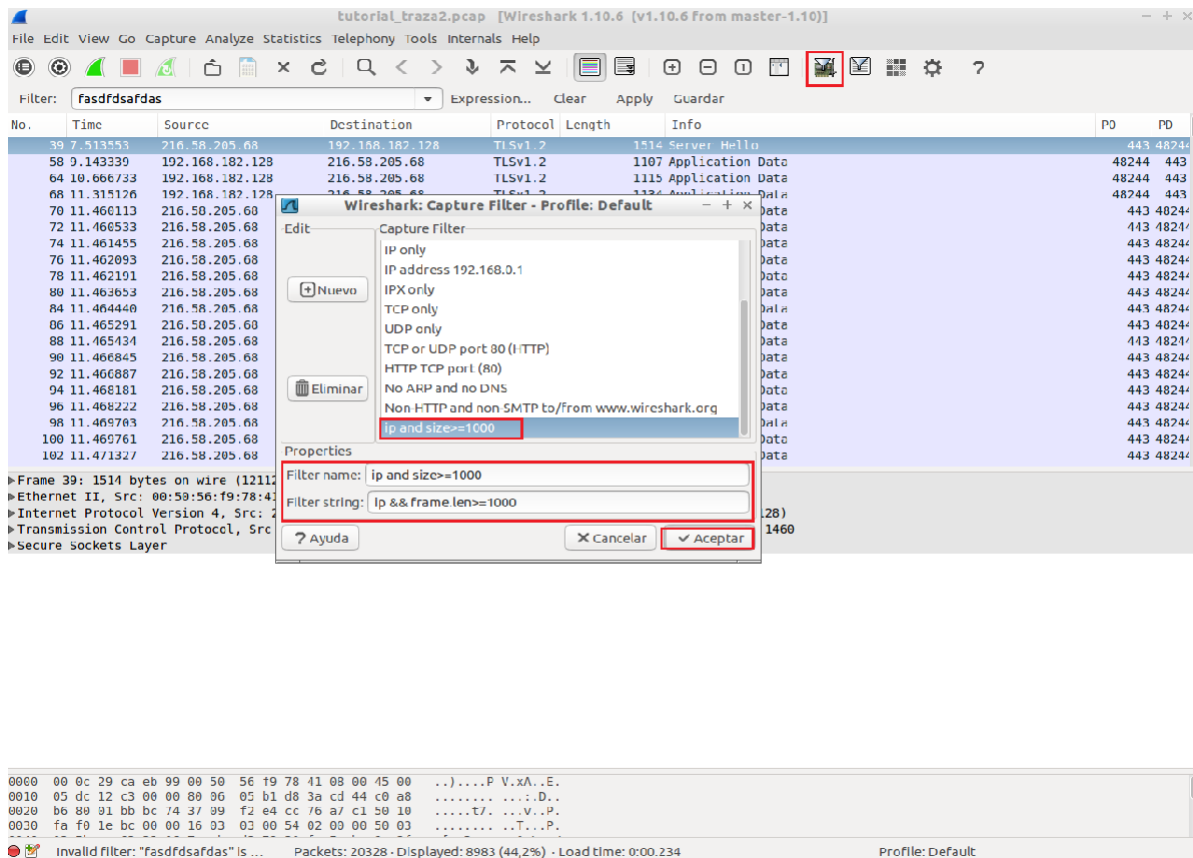
Se pide realizar una captura de tráfico entrando en varias páginas aleatorias en Internet y, después de detener la captura, añadir un filtro de modo que solo se visualicen los paquetes que sean de tipo IP y que tengan un tamaño de paquete mayor a 1000 Bytes.



Podemos ver en la barra del filtro que hemos seleccionado solo los paquetes de tipo IP y los paquetes con un tamaño mayor o igual que 1000 Bytes. Sabemos que el filtro selecciona los paquetes con un tamaño mayor a 1000 bytes porque para cada paquete que aparece en la captura, su columna 'Length' es mayor o igual a 1000, cosa que es ratificada más abajo en 'Frame ... X bytes on wire ... X bytes captured'. Por otro lado, sabemos que todos los paquetes son de tipo IP porque, extendiendo en panel 'Ethernet II' para cada paquete, podemos observar que son de tipo IP:



A continuación se pregunta como almacenar en una captura solo los paquetes seleccionados. Si esta pregunta se refiere a como guardar los paquetes capturados en concreto, podemos guardar la traza de la captura tal y como se ha hecho en el ejercicio 1. Si por el contrario se pregunta como hacer una captura con este filtro ya seleccionado, solo tenemos que guardar el filtro en los filtros de captura:



Por último, se pregunta que relación observamos entre el tamaño de paquete IP y el campo 'length' del protocolo IP del mismo. Después de ver varios ejemplos, no fue muy difícil ver que la diferencia entre el tamaño del paquete y el campo 'Length' del mismo es de 14 para todos los paquetes.

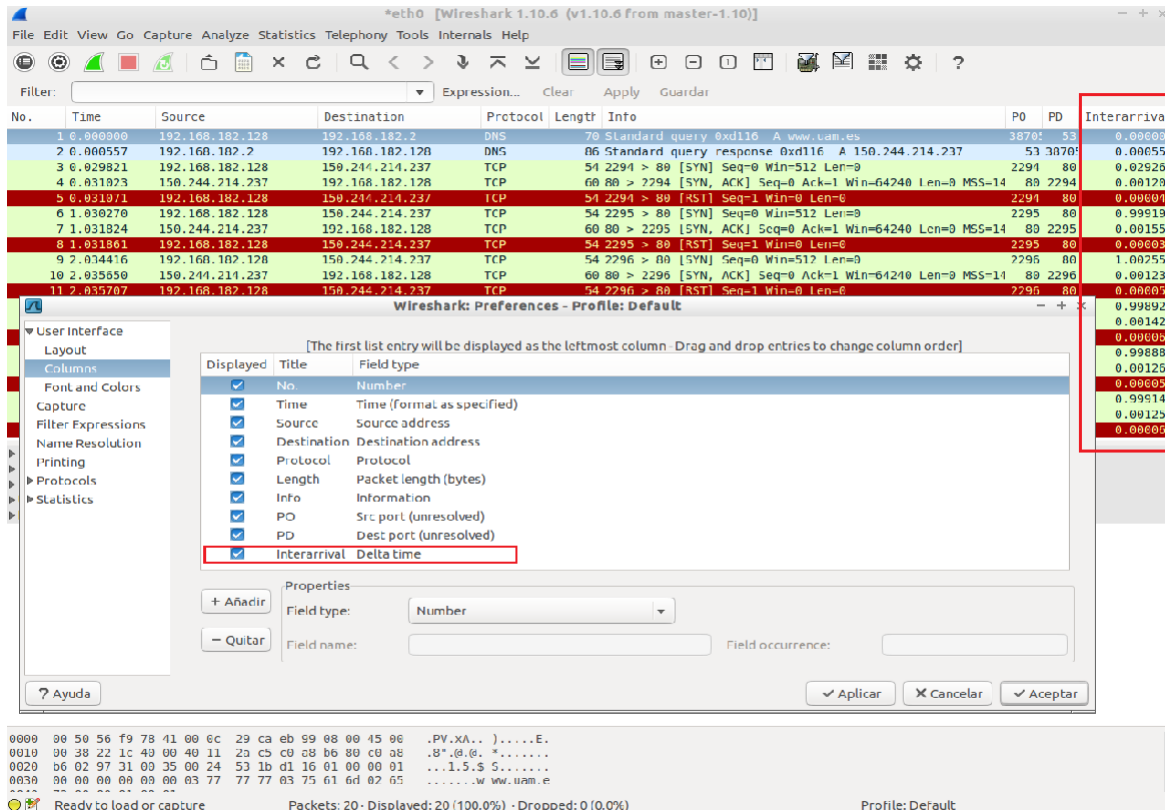
No.	Time	Source	Destination	Protocol	Length	Info
9	1.443233	150.244.214.5	192.168.59.128	TLSv1.2	1464	Server Hello
11	1.445657	150.244.214.5	192.168.59.128	TLSv1.2	1514	Certificate
34	1.692023	150.244.214.5	192.168.59.128	TCP	1464	[TCP segment of a reassembled PDU]
35	1.693229	150.244.214.5	192.168.59.128	TCP	1464	[TCP segment of a reassembled PDU]
38	1.693834	150.244.214.5	192.168.59.128	TCP	1464	[TCP segment of a reassembled PDU]

Frame 38: 1464 bytes on wire (11712 bits), 1464 bytes captured (11712 bits) on interface 0 Ethernet II, Src: VMware_f0:50:56:00:12:00, Dst: VMware_ba:2f:19:00:0c:29:ba:2f:19 Internet Protocol Version 4, Src: 150.244.214.5 [150.244.214.5], Dst: 192.168.59.128 [192.168.59.128] Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1450 Identification: 0x0cb7 (3255) Flags: 0x00 Fragment offset: 0 Time to live: 128 Protocol: TCP (6) Header checksum: 0xb774 [validation disabled] Source: 150.244.214.5 [150.244.214.5] Destination: 192.168.59.128 [192.168.59.128] [Source GeoIP: Unknown] [Destination GeoIP: Unknown] Transmission Control Protocol, Src Port: https (443), Dst Port: 49438 (49438), Seq: 6785, Ack: 688, Len: 1410						
--	--	--	--	--	--	--

La causa de esta diferencia la desconocemos, pero suponemos que puede ser a que ha sido suprimida una cabecera, ya que según avanzamos en los niveles o capas, se van eliminando las cabeceras ya leídas.

4 Ejercicio 3

Se pide añadir una columna *interarrival* que muestre el tiempo entre paquetes consecutivos.



En Edit-Preferences-Columns hemos añadido una columna tal y como hemos hecho antes con 'PD' y 'PO', con el nombre 'Interarrival'. Para que esta columna realizase la función requerida le hemos asignado un *Field type* de tipos *Delta time*. Podemos ver en la imagen anterior el tiempo entre paquetes consecutivos en la columna más a la derecha.

5 Ejercicio 4

Modificamos la forma en que Wireshark muestra la información en la columna 'Time' para que muestre los tiempos en formato para humanos.

*eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Guardar

No.	Time	Source	Destination	Protocol	Length	Info	PO	PD	Interarrival
15	2018-09-25 20:47:51.265865	150.244.214.237	192.168.182.128	TCP	60	80 > 2299 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2299	0.001259
16	2018-09-25 20:47:50.265602	150.244.214.237	192.168.182.128	TCP	60	80 > 2298 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2298	0.001263
13	2018-09-25 20:47:49.265391	150.244.214.237	192.168.182.128	TCP	60	80 > 2297 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2297	0.001428
10	2018-09-25 20:47:48.264985	150.244.214.237	192.168.182.128	TCP	60	80 > 2296 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2296	0.001234
7	2018-09-25 20:47:47.261160	150.244.214.237	192.168.182.128	TCP	60	80 > 2295 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2295	0.001554
4	2018-09-25 20:47:46.260359	150.244.214.237	192.168.182.128	TCP	60	80 > 2294 [SYN, ACK] Seq=0 Ack=1 Win=64 Len=0	86	2294	0.001202
2	2018-09-25 20:47:46.229893	192.168.182.2	192.168.182.128	DNS	86	Standard query response 0xd116 A 150.244.214.237	53	3870	0.009557
20	2018-09-25 20:47:51.265812	192.168.182.128	150.244.214.237	TCP	54	2299 > 80 [RST] Seq=1 Win=0 Len=0	2299	80	0.003065
18	2018-09-25 20:47:51.264007	192.168.182.128	150.244.214.237	TCP	54	2299 > 80 [SYN] Seq=0 Win=512 Len=0	2299	80	0.999140
17	2018-09-25 20:47:50.265659	192.168.182.128	150.244.214.237	TCP	54	2298 > 80 [RST] Seq=1 Win=0 Len=0	2298	80	0.003657
15	2018-09-25 20:47:50.264339	192.168.182.128	150.244.214.237	TCP	54	2298 > 80 [SYN] Seq=0 Win=512 Len=0	2298	80	0.998883
14	2018-09-25 20:47:49.265556	192.168.182.128	150.244.214.237	TCP	54	2297 > 80 [RST] Seq=1 Win=0 Len=0	2297	80	0.003665
12	2018-09-25 20:47:49.263963	192.168.182.128	150.244.214.237	TCP	54	2297 > 80 [SYN] Seq=0 Win=512 Len=0	2297	80	0.998928
11	2018-09-25 20:47:48.265043	192.168.182.128	150.244.214.237	TCP	54	2296 > 80 [RST] Seq=1 Win=0 Len=0	2296	80	0.003657
9	2018-09-25 20:47:48.263752	192.168.182.128	150.244.214.237	TCP	54	2296 > 80 [SYN] Seq=0 Win=512 Len=0	2296	80	1.002555
6	2018-09-25 20:47:47.261197	192.168.182.128	150.244.214.237	TCP	54	2295 > 80 [RST] Seq=1 Win=0 Len=0	2295	80	0.003637
3	2018-09-25 20:47:47.259605	192.168.182.128	150.244.214.237	TCP	54	2295 > 80 [SYN] Seq=0 Win=512 Len=0	2295	80	0.999199
5	2018-09-25 20:47:46.260407	192.168.182.128	150.244.214.237	TCP	54	2294 > 80 [RST] Seq=1 Win=0 Len=0	2294	80	0.003648
3	2018-09-25 20:47:46.259157	192.168.182.128	150.244.214.237	TCP	54	2294 > 80 [SYN] Seq=0 Win=512 Len=0	2294	80	0.029284

>Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
 >Ethernet II, Src: 08:00:29:ca:eb:99 (08:00:29:ca:eb:99), Dst: 08:00:56:f9:78:41 (08:00:56:f9:78:41)
 >Internet Protocol Version 4, Src: 192.168.182.128 (192.168.182.128), Dst: 192.168.182.2 (192.168.182.2)
 >User Datagram Protocol, Src Port: 38795 (38795), Dst Port: 53 (53)
 >Domain Name System (query)

0000 00 56 f9 78 41 00 0c 29 ca eb 99 08 00 56 f9 78 41 .PV.xA..).....E.
 0010 00 3e 22 1c 49 00 40 11 2a c5 c0 e8 b6 80 c0 e8 .8*.@.@.*.....
 0020 b6 02 97 31 09 35 00 24 53 1b d1 16 61 00 96 01 ...1.5.5 S.....
 0030 00 00 00 00 00 03 77 77 77 03 75 61 6d 02 65w ww.uan.e

Ready to load or capture Packets: 20 · Displayed: 20 (100,0%) · Dropped: 0 (0,0%) Profile: Default

Esto se hace en View/Time Display Format/Date and Time of Day. Seleccionando esta última opción en el menú de 'Time Display Format' obtenemos el formato para humanos. Del mismo modo hacemos para ver el tiempo Unix del paquete: View/Time Display Format/Seconds since Epoch. Se muestra en la imagen de abajo:

*eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Guardar

No.	Time	Source	Destination	Protocol	Length	Info	PO	PD	Interarrival
15	1537968471.266866	150.244.214.237	192.168.182.128	TCP	60	80 > 2299 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2299	0.001259
16	1537968476.265602	150.244.214.237	192.168.182.128	TCP	60	80 > 2298 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2298	0.001263
13	1537968469.265391	150.244.214.237	192.168.182.128	TCP	60	80 > 2297 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2297	0.001420
10	1537968468.264986	150.244.214.237	192.168.182.128	TCP	60	80 > 2296 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2296	0.001234
7	1537968467.261160	150.244.214.237	192.168.182.128	TCP	60	80 > 2295 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2295	0.001554
4	1537968466.260359	150.244.214.237	192.168.182.128	TCP	60	80 > 2294 [SYN, ACK] Seq=0 Ack=1 Win=64246 Len=0	89	2294	0.001202
2	1537968466.229893	192.168.182.2	192.168.182.128	DNS	86	Standard query response 0xd116 A 150.244.214.237	53	3870	0.009557
20	1537968471.266132	192.168.182.128	150.244.214.237	TCP	54	2299 > 80 [RST] Seq=1 Win=0 Len=0	2299	89	0.003065
18	1537968471.264807	192.168.182.128	150.244.214.237	TCP	54	2299 > 80 [SYN] Seq=0 Win=512 Len=0	2299	89	0.999148
17	1537968476.265659	192.168.182.128	150.244.214.237	TCP	54	2298 > 80 [RST] Seq=1 Win=0 Len=0	2298	89	0.003657
15	1537968476.264339	192.168.182.128	150.244.214.237	TCP	54	2298 > 80 [SYN] Seq=0 Win=512 Len=0	2298	89	0.998883
14	1537968469.265556	192.168.182.128	150.244.214.237	TCP	54	2297 > 80 [RST] Seq=1 Win=0 Len=0	2297	89	0.003665
12	1537968468.263963	192.168.182.128	150.244.214.237	TCP	54	2297 > 80 [SYN] Seq=0 Win=512 Len=0	2297	89	0.998928
11	1537968468.263752	192.168.182.128	150.244.214.237	TCP	54	2296 > 80 [RST] Seq=1 Win=0 Len=0	2296	89	0.003657
9	1537968468.263552	192.168.182.128	150.244.214.237	TCP	54	2296 > 80 [SYN] Seq=0 Win=512 Len=0	2296	89	1.002555
6	1537968467.261197	192.168.182.128	150.244.214.237	TCP	54	2295 > 80 [RST] Seq=1 Win=0 Len=0	2295	89	0.003637
3	1537968467.259606	192.168.182.128	150.244.214.237	TCP	54	2295 > 80 [SYN] Seq=0 Win=512 Len=0	2295	89	0.999199
5	1537968466.260407	192.168.182.128	150.244.214.237	TCP	54	2294 > 80 [RST] Seq=1 Win=0 Len=0	2294	89	0.003648
3	1537968466.259157	192.168.182.128	150.244.214.237	TCP	54	2294 > 80 [SYN] Seq=0 Win=512 Len=0	2294	89	0.029284

>Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
 >Ethernet II, Src: 08:00:29:ca:eb:99 (08:00:29:ca:eb:99), Dst: 08:00:56:f9:78:41 (08:00:56:f9:78:41)
 >Internet Protocol Version 4, Src: 192.168.182.128 (192.168.182.128), Dst: 192.168.182.2 (192.168.182.2)
 >User Datagram Protocol, Src Port: 38705 (38705), Dst Port: 53 (53)
 >Domain Name System (query)

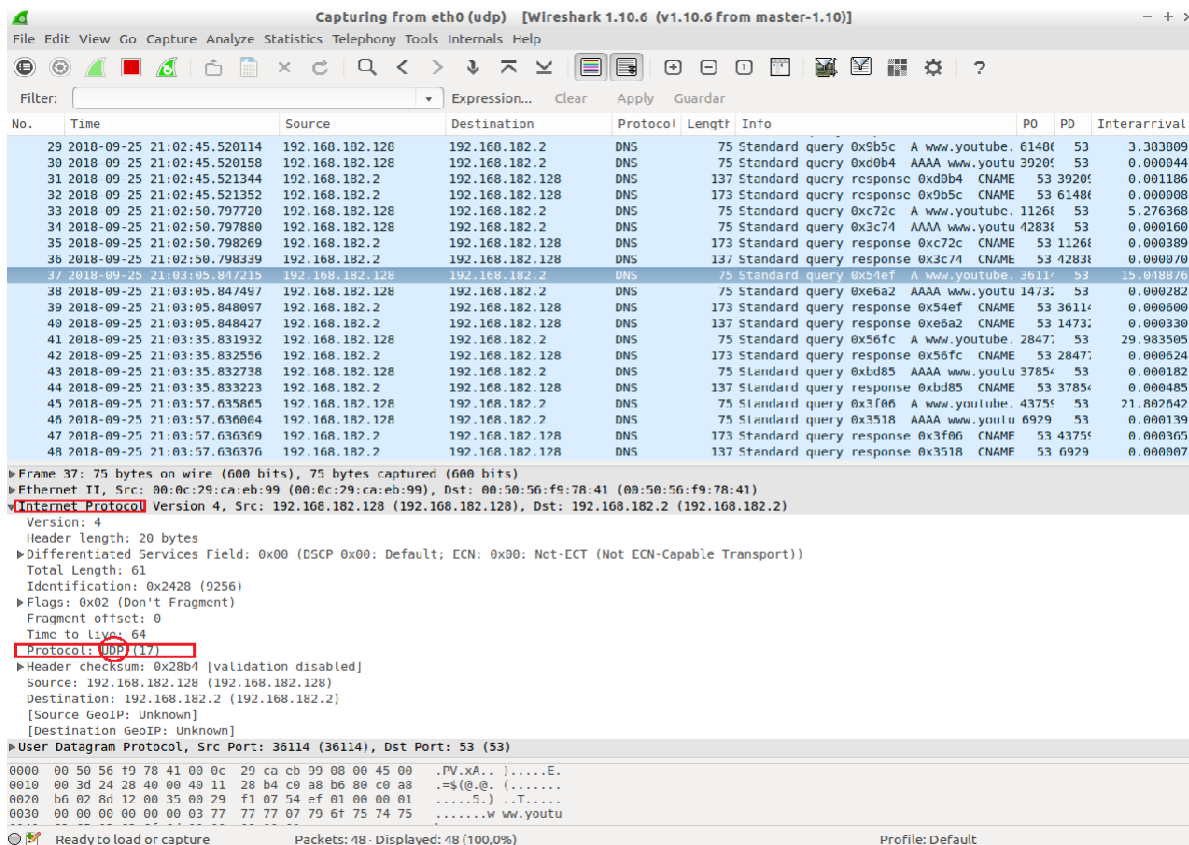
0000 00 56 f9 78 41 00 0c 29 ca eb 99 08 00 56 f9 78 41 .PV.xA..).....F.
 0010 00 3e 22 1c 49 00 40 11 2a c5 c0 e8 b6 80 c0 e8 .8*.@.@.*.....
 0020 b6 02 97 31 09 35 00 24 53 1b d1 16 61 00 96 01 ...1.5.5 S.....
 0030 00 00 00 00 00 03 77 77 77 03 75 61 6d 02 65w ww.uan.e

Ready to load or capture Packets: 20 · Displayed: 20 (100,0%) · Dropped: 0 (0,0%) Profile: Default

6 Ejercicio 5

Último ejercicio de esta práctica de iniciación. Se pide aplicar otro filtro de captura para que solo se capturen los paquetes de tipo UDP. Para ello solo tenemos que aplicar el filtro de captura 'UDP only' cuando estemos especificando las opciones de la captura.

Cuando tengamos la captura de tráfico lista, ejecutamos en la terminal el comando requerido y, al mismo tiempo, realizamos varias búsquedas en Internet. Obtenemos un tráfico tal que así:



7 Comentarios finales

Se adjunta una captura de tráfico con la que hemos hecho los ejercicios 2, 3 y 4 llamada **practica1.pcap**. En la documentación de la práctica solo se pedía guardar una captura de tráfico, por lo que hemos supuesto que era la que más ejercicios abarcara. Adicionalmente comentar que los pantallazos enseñados en este documento no se han extraído a partir de la captura que se adjunta, sino que proceden de la captura de tráfico en vivo mientras realizábamos el ejercicio por primera vez la primera semana de prácticas.