Lucía Colmenarejo Pérez y Marta García Marín G1302 P01

Memoria Práctica 3

Redes de Comunicaciones 1

1.Introducción:

En esta práctica, profundizaremos en el uso del tshark, shell y el awk, así como la utilización de estos en scripts. Para ello, crearemos una traza de wireshark personalizada, a partir de la cual tendremos que calcular diferentes valores a partir de los datos de los niveles de cada paquete.

Utilizando filtros sobre la traza trabajaremos con paquetes IP, puertos y crearemos ECDFs de los diferentes tamaños de los paquetes según niveles, así como de los tiempos entre las llegadas de flujo del nivel 4.

Todo esto lo realizaremos con el fin de desempeñar un buen papel como gestores de red para así abarcar tanto problemas de la propia red como problemas de los equipos y aplicaciones que la componen.

Para la realización de esta práctica necesitamos ciertos valores que obtenemos al ejecutar el generador de trazas que nos proporcionan como material. Estos valores son los siguientes:

- La dirección MAC(nivel 2) : 00:11:88:CC:33:E1
- La dirección IP (origen o destino) del flujo TCP: 98.64.49.36
- El puerto (origen o destino) del flujo UDP: 25256

2.Análisis de datos:

2.1.Porcentajes de los paquetes capturados en la traza:

```
Porcentaje IP: 99.0515 %
Porcentaje NO IP: 0.948461 %
Paquetes IP:
        Porcentaje TCP: 84.6891 %
        Porcentaje UDP: 13.0486 %
        Porcentaje OTROS: 2.26224 %
```

Ante estos resultados podemos llegar a la conclusión de que la gran mayoría de los paquetes usan el protocolo IP (donde están incluidos los paquetes VLAN). El protocolo IP en un comienzo se creó con la intención de realizar la entrega de paquetes de una form no fiable. Lo más importante es que IP trata de encontrar la mejor ruta posible para el transporte de los paquetes desde un host origen a un host destino. IP lo único que nos garantiza es la seguridad de las cabeceras de los paquetes transportados, pero no de los datos que estos portan. Para hablar de fiabilidad tendremos que fijarnos en los protocolos de la capa de transporte, es decir, TCP y UDP. El resto de paquetes que no siguen el protocolo IP pueden seguir otros protocolos como son OSPF, IS-IS, IGMP..., pero son muy pocas debido a que el protocolo IP ya que es el elemento común de Internet hoy en día.

Dentro del protocolo IP hemos obtenido los porcentajes de paquetes TCP,UDP y otros(son aquellos que no son ni TCP ni UDP). Los dos protocolos más importantes en la

capa de transporte son TCP y UDP, que fueron los primeros en definirse. TCP es el más utilizado hoy en día ya que ofrece un servicio fiable independientemente de las capas inferiores. En cambio UDP no te garantiza de ninguna manera si los paquetes llegan a su destino ni si llegan de la forma correcta, este tipo de protocolo es utilizado sobre todo para la retransmisión de audio y de vídeo en streaming. El resto de protocolos en la capa de transporte son menos utilizados como por ejemplo DNS, los cuales asignan un nombre a una dirección IP de tal modo que se simplifica el modo de acceder a un servidor; o HTTP, que se encarga de establecer la conexión entre el servidor y el navegador.

Observamos, como estudiamos en clase de teoría, que el protocolo IP es el más abundante.

Para el cálculo de porcentajes de paquetes IP y no IP:

1) Volcamos en un .txt el tipo de ethernet de todos los paquetes de la traza:
   tshark -r traza.pcap -T fields -e eth.type>eth_type.txt
2) Calculamos el porcentaje de paquete IP, tomando aquellos valores del .txt creado anteriormente cuyo valor sea 0x0800 o 0x8100:
   awk -v var1=`grep '0800' eth_type.txt | wc -l` -v var2=`grep '8100' eth_type.txt | wc -l` -v var3=`wc -l <eth_type.txt` 'BEGIN{print "Porcentaje IP: " (var1+var2)/var3*100, "%" }'. Restando a 100 el resultado anterior, obtenemos el porcentaje de paquetes no IP: **awk -v var1=`grep '0800' eth_type.txt | wc -l` -v var2=`grep '8100' eth_type.txt | wc -l` -v var3=`wc -l <eth_type.txt` 'BEGIN{print "Porcentaje NO IP: "100- (var1+var2)/var3*100, "%" }'**

Dentro de los paquetes IP, distinguimos entre TCP, UDP y otros según su protocolo. Para ello:

1) Introducimos en un .txt todos los protocolos de los paquetes IP de la traza:
   tshark -r traza.pcap -T fields -e ip.proto>protocol.txt
2) Para UDP, buscamos que el protocolo sea 6:
   awk -v var1=`grep '6' protocol.txt | wc -l` -v var2=`wc -l <protocol.txt` '
   BEGIN{print "\tPorcentaje TCP: " (var1/var2)*100, "%" }'.
   Para UDP, buscamos que el protocolo sea 17:

awk -v var1=`grep '17' protocol.txt | wc -l` -v var2=`wc -l <protocol.txt` 'BEGIN{print "\tPorcentaje UDP: " (var1/var2)*100, "%" }'.

   Para otros protocolos, calculamos aquellos que no tengan ni protocolo 6, ni 17:
   awk -v var1=`grep '6' protocol.txt | wc -l` -v var2=`grep '17' protocol.txt | wc -l` -v var3=`wc -l <protocol.txt` 'BEGIN{print "\tPorcentaje OTROS: " 100-((var1+var2)/var3)*100, "%" }'

2.2.Cálculo de los top 10 de direcciones IP activas y los top 10 de puertos:

Para el cálculo del Top 10 de las direcciones IP destino en paquetes , introducimos todas estas en un .txt y lo ordenamos, de forma que la que aparezca más veces sea la primera y la que aparezca menos sea la última, tomando después las 10 primeras.
Para el cálculo en bytes, tomamos el tamaño de estas direcciones, ya convertido a bytes, y sumamos todos aquellos cuyas direcciones coincidan (**{bytessrc[$1] += $2;}**), obteniendo así el tamaño total en bytes de una dirección. Ordenamos estos tamaños de mayor a menor y tomamos los 10 primeros, los cuales introducimos en el fichero .txt.
Realizaremos este mismo proceso para el el cálculo del Top 10 de las direcciones IP origen, así como para el resto de TOPs a calcular.

Top 10 de direcciones IP activas origen:

Paquetes:

```
15454 127.240.154.197
11463 9.27.79.108
 6423 21.234.85.63
 4657 73.156.119.101
 2906 56.36.157.186
 2188 59.74.162.98
 2161 91.154.43.179
 2048 95.63.219.68
 2020 56.105.211.86
 1883 98.64.49.36
```

Bytes:

```
127.240.154.197 23098523
73.156.119.101 6918040
56.36.157.186 4344112
59.74.162.98 3245100
91.154.43.179 3193577
95.63.219.68 3009353
98.64.49.36 2730262
19.249.95.42 2473818
21.234.85.63 1400214
9.27.79.108 1025537
```

Podemos ver que en comparación del top 10 de paquetes de IP destino y del top 10 de bytes por IP hay algunas diferencias, aunque no muchas; esto es dependiendo de la traza que ha sido generada en nuestro caso.

Top 10 de las direcciones IP activas destino:

Paquetes:

```
34986 9.27.79.108
 6423 65.120.61.242
 3881 127.240.154.197
 2857 56.105.211.86
 1273 73.156.119.101
 1046 98.64.49.36
  983 56.36.157.186
  666 91.154.43.179
  664 57.254.24.120
  619 19.249.95.42
```

Bytes:

```
9.27.79.108 50345203
56.105.211.86 2853122
65.120.61.242 1400214
127.240.154.197 249160
57.254.24.120 115206
73.156.119.101 79229
34.119.213.61 76301
98.64.49.36 70017
56.36.157.186 59576
91.154.43.179 47886
```

Podemos ver que en comparación del top 10 de paquetes de IP destino y del top 10 de bytes por IP hay algunas diferencias, aunque no muchas; esto es dependiendo de la traza que ha sido generada en nuestro caso.

Top 10 de los puertos TCP origen:

Paquetes:

```
36640 80
 1423 55934
 1096 55860
 1046 54615
  617 55865
  607 43585
  603 33896
  471 55173
  418 55848
  380 33903
```

Bytes:

```
80 52857665
443 217800
55934 88065
54615 70017
55860 67367
55865 40574
43585 36512
33896 35533
55173 28338
46832 26382
```

La PC de cada uno ocupa un puerto aleatorio, al momento de originar una petición al servidor, en el caso del HTTP siempre será, indistintamente el puerto 80, el que escuche o envía la solicitud de servicio hecha por la PC cliente, de esta forma el paquete con el puerto 80 (en TCP) será el más enviado. En el caso en que la TCP de origen es el puerto 80 es también el que, en general, ocupa más espacio debido a que envía toda la información o datos que la PC ha solicitado anteriormente.

Podemos destacar que en el top 10 de los paquetes TCP de origen está como segundo el puerto 443 que es un puerto que se emplea para establecer conexiones web cifradas para proteger los datos y las credenciales.

Top 10 de los puertos TCP destino:

Paquetes:

```
12342 80
 5486 55934
 4313 55860
 3204 55865
 2188 43585
 1883 54615
 1813 33896
 1717 55173
 1396 55848
 1174 46371
```

En los puertos de destino de TCP también el paquete con el puerto 80 es el más abundante debido a que es el que recibe la solicitud de servicio que ha hecho la PC.

Bytes:

```
55934 8236507
55860 6437994
55865 4808618
43585 3245100
54615 2730262
33896 2707440
55173 2566453
55848 2072650
46371 1756652
57063 1690967
```

Si nos damos cuenta a nivel de byte ya no está el puerto 80 debido a que al enviar la solicitud el paquete no contiene excesiva información,es el momento en el este puerto establece la conexión cuando los paquetes contienen muchas información (explicado en el caso de TCP de origen).

Top 10 de los puertos UDP origen:

Paquetes:

```
6423 24704
 592 53
 124 546
  95 5353
  12 1900
   6 63423
   6 58532
   6 55421
   6 49169
   3 61153
```

Bytes:

```
24704 1400214
53 85720
5353 23317
546 18337
1900 6447
63423 1080
58532 1080
55421 1080
49169 1080
61153 624
```

Top 10 de los puertos UDP destino:

Paquetes:

```
6423 25256
 591 53
 134 5355
 124 547
  95 5353
  42 1900
   2 5035
   2 12013
   1 9920
   1 9800
```

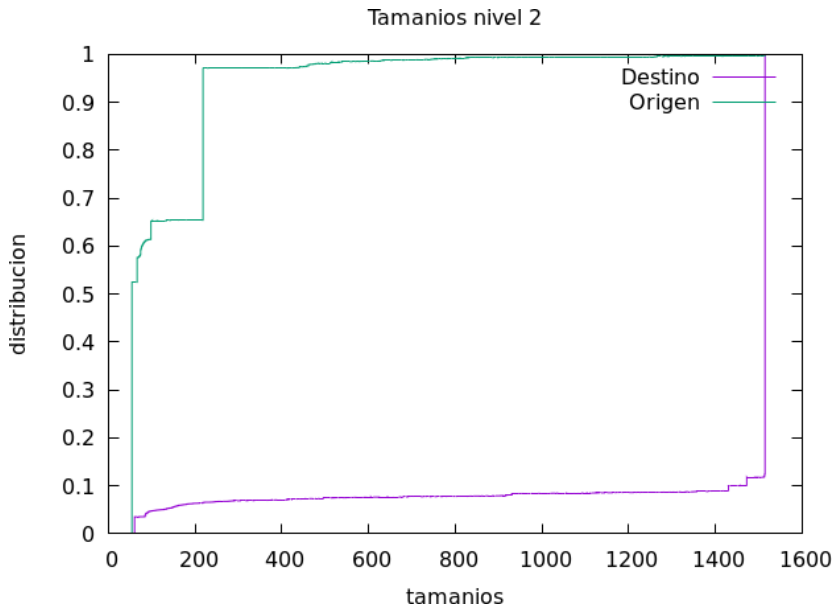
Bytes:

```
25256 1400214
53 46391
5353 23317
547 18337
1900 12015
5355 11460
12013 533
5035 461
64925 394
23710 318
```

Vemos que el puerto 53 es uno de los más usados y es porque se utiliza para la resolución de nombres de dominio (consulta de direcciones). Aunque su tamaño total parece elevado, cada uno de los paquetes individualmente no ocupa demasiado espacio, la cuestión es que son un elevado número.

2.3.ECDFs (funciones de distribución):

ECDF de los tamaños a nivel 2:



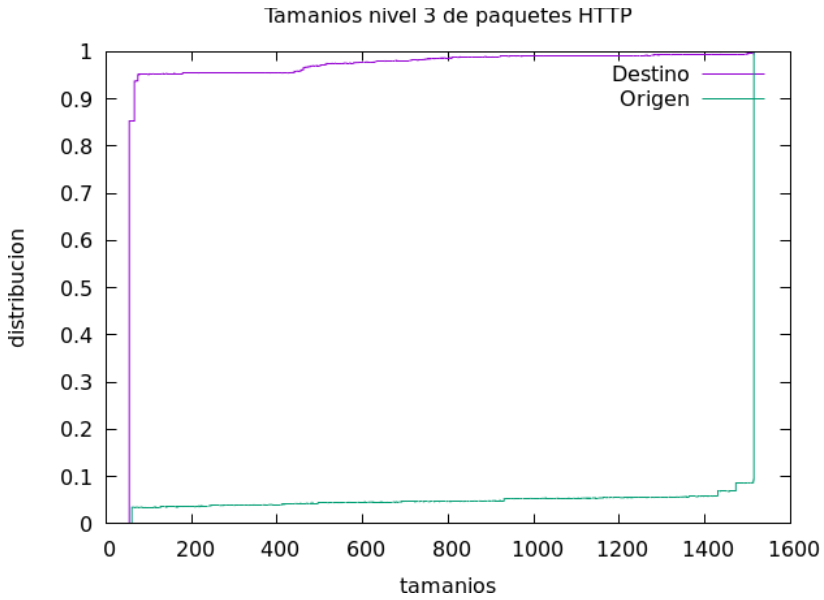
En esta gráfica se puede apreciar el tráfico de paquetes de nuestra traza que salen o llegan a nuestro dispositivo (identificado con una dirección MAC que obtenemos de la información dada al generar la traza por terminal, que en nuestro caso es 00:11:88:CC:33:E1).

La distribución de los paquetes que salen de nuestro dispositivo(origen) podemos concluir que es exponencial. Este comportamiento se debe a que nuestro dispositivo prácticamente todos los paquetes que va a enviar van a ser de solicitud de conexión o de acceso a una página web, que serán primeros enviados al DNS para que nos devuelva la dirección ip a la que nos queremos conectar, y que serán de un tamaño relativamente pequeño (ya que es prácticamente una cadena de caracteres). El crecimiento exponencial se debe a que todos los paquetes van a ser prácticamente del mismo tamaño, que como vemos en la gráfica estará entre 50 y 200 bytes.

La distribución de los paquetes que entran a nuestro dispositivo (destino) podemos deducir que es logarítmica. Este comportamiento se debe a que una vez realizada la solicitud de conexión a una página web el paquete de vuelta a tu dispositivo (nuestra mac como dirección destino) contendrán toda la información de dicha página. Por ello el crecimiento es de forma logarítmica ya que la mayoría de los paquetes tienen el máximo tamaño que pueden tener, que es de 1514 bytes.

La zona de la gráfica de origen que tiene pendiente 0 es debido a cómo hemos decidido representar la gráfica, esto es “with steps”, que remarca los cambios para que sea más claro ver el crecimiento.

ECDF de los tamaños a nivel 3 de los paquetes HTTP:

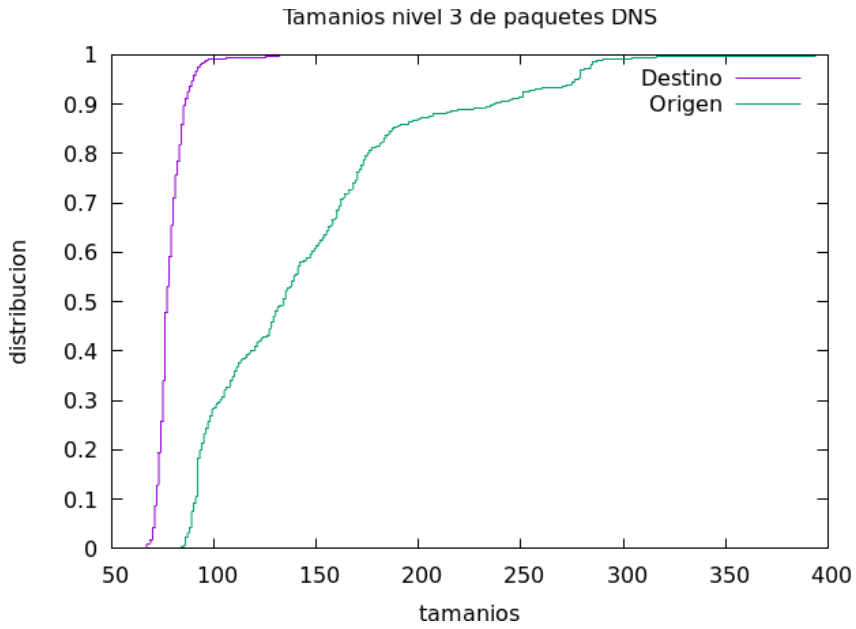


En esta gráfica podemos apreciar el tamaño (a nivel 3) de paquetes HTTP para el puerto 80, tanto como puerto de destino como de origen.

La distribución de los paquetes que tienen como puerto origen el puerto 80 tiene un crecimiento logarítmico, esto se debe a que el servidor al que hicimos la solicitud devuelve toda la información de la página web que se ha pedido, entonces tendrá el mayor tamaño que puede tener un paquete que es de 1514 bytes.

Por lo contrario la distribución de los paquetes que tienen como puerto destino el puerto 80 tiene un crecimiento exponencial debido a que el navegador realiza una solicitud HTTP para que el servidor al que la manda (puerto destino 80) procese la solicitud y posteriormente envíe una respuesta HTTP. Este paquete contiene una serie de líneas que contienen información como: línea de solicitud (especifica el tipo de documento solicitado, el método que se va a aplicar y la versión del protocolo), los campos del encabezado de solicitud y el cuerpo de solicitud (información opcional). Por lo tanto una solicitud HTTP al fin y al cabo no es más que una cadena de caracteres, por lo que su tamaño no será muy elevado; como apreciamos en la gráfica se encuentra entre unos 50-70 bytes.

ECDF de los tamaños a nivel 3 de los paquetes DNS:



En esta gráfica podemos apreciar el tamaño (a nivel 3) de paquetes DNS para el puerto 53, tanto como puerto de destino como de origen.

La distribución de los paquetes que tienen como puerto destino el destino el 53 son los paquetes generados por lo que se conoce con el nombre de Cliente DNS . Estos paquetes contienen peticiones de consulta para resolver nombres y direcciones IP (por lo general contienen la URL de la dirección que están solicitando). Básicamente preguntan por la dirección IP que corresponde a un nombre determinado, por eso el tamaño del paquete es pequeño (en torno a unos 50-80 bytes, que es lo que podemos apreciar en la gráfica).

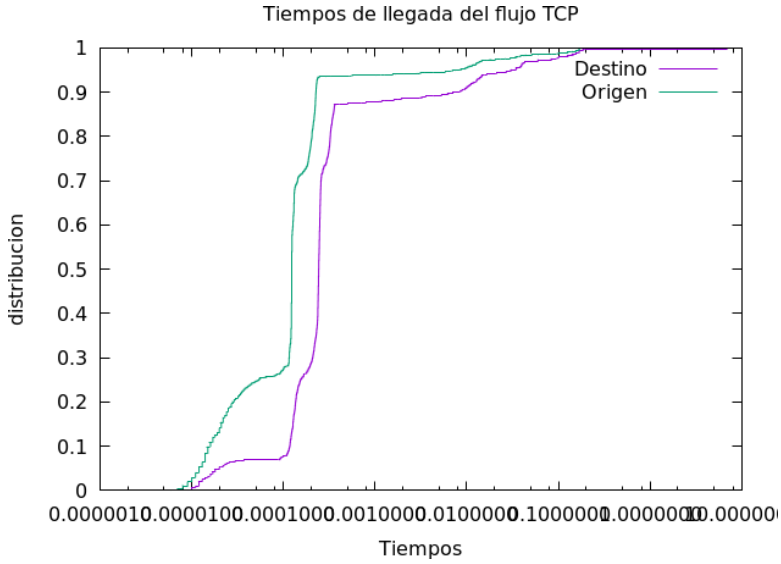
Por lo contrario la distribución de los paquetes que tienen como puerto origen el puerto 53 son los paquetes que se forman en los Servidores DNS. Estos paquetes por lo general contienen la dirección IP solicitada por el Cliente DNS. Es por ello que el tamaño de estos paquetes tampoco va a diferir excesivamente de los paquetes de solicitud, ya que por lo general contienen una dirección. Sin embargo, una dirección suele ocupar algo más de espacio que una URL o un nombre, por eso en la gráfica vemos que tiene un tamaño en torno a 80-300 bytes.

Casi todas las consultas consisten de una sola solicitud UDP desde un Cliente DNS, seguida por una sola respuesta UDP del Servidor DNS. Se realiza una conexión TCP cuando el tamaño de los datos de la respuesta exceden los 512 bytes, tal como ocurre con tareas como transferencia de zonas.

Por estas razones podemos concluir mirando la gráfica que la distribución en ambos casos tiende a ser exponencial. Aunque en el caso de la distribución con puerto de origen 53

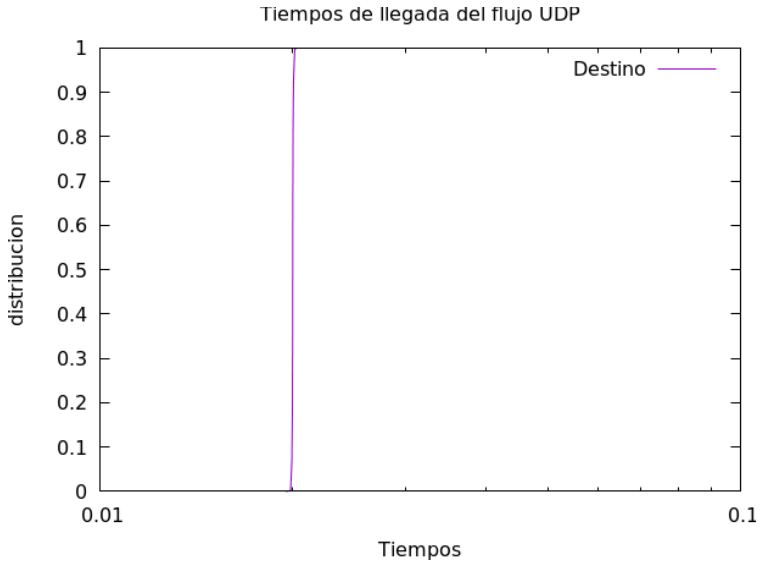
es menos significativa dado que el tamaño de los paquetes varía en un rango algo mayor que en la de destino.

ECDF de los tiempo entre llegadas del flujo TCP:



En esta gráfica podemos apreciar que las dos distribuciones se parecen bastante debido a que se está enviando y recibiendo paquetes de forma simultánea por lo que los paquetes están siendo afectados por la misma congestión y el mismo retardo (influye en ambos el jitter). Un hecho que cabe destacar es que hay paquetes entre los que se puede apreciar que hay un mayor tiempo de diferencia (tanto en origen como en destino) uno de los motivos por los que puede ser es, por ejemplo, que en ese momento no hay flujo de paquetes con el protocolo TCP; ambiñe puede ser que el flujo haya disminuido.

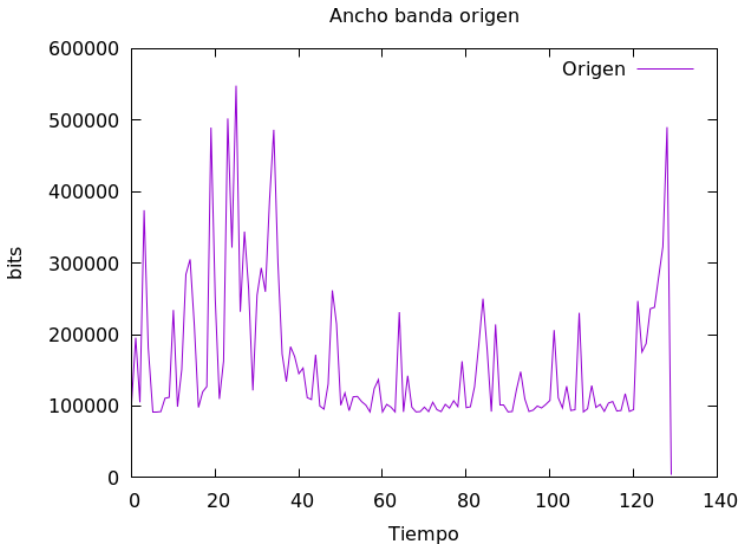
ECDF de los tiempo entre llegadas del flujo UDP:



Observamos que no hay flujo UDP de origen. Esto se debe a que en UDP hay un flujo unidireccional, donde no se establece ninguna sincronización entre el origen y el destino.

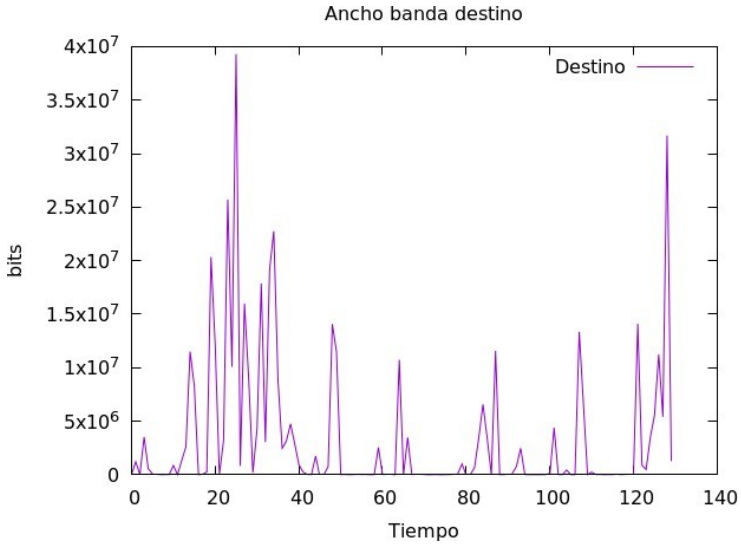
Como es un protocolo no fiable no hace comprobaciones de cabecera, ni suma de bits de control, ni manda paquetes ACK,... , por lo que no va a generar ningún tipo de retardo. De esta manera no hay apenas diferencia de tiempo entre los paquetes, de ahí que veamos en la gráfica una recta totalmente vertical, ya que debido a la escala no se puede apreciar con claridad. Si la pusiéramos en la misma gráfica que TCP y con la misma escala apreciaríamos una leve inclinación con respecto a la vertical.

Ancho de banda a nivel 2 en b/s de la traza origen:



Esta gráfica representa los paquetes que está generando la IP de origen, que por lo general, observando la gráfica, es en los primeros 40 segundos donde se produce el mayor caudal y luego al final de la conexión (esto se debe a que por lo general, al comienzo es cuando más se envían paquetes de solicitud a acceso y otro tipo de paquetes)

Ancho de banda a nivel 2 en b/s de la traza destino:



En esta gráfica podemos ver que hay una serie de picos muy pronunciados. Esto es debido a que como es de dirección destino se van a recibir paquetes cada vez que

interactuamos con la página (clickear en la página, abrir una pestaña…), mientras que no se interactúe con la página no se producirán picos en la gráfica debido a que no se transmiten paquetes y, por tanto, bytes.

En cuanto a la implementación de nuestro código para la obtención de los datos que nos han ayudado a realizar su representación gráfica, vamos a dar una breve explicación de ella.

Por una parte, abordaremos el código de las ECDFs de los tamaños de nivel (código que tenemos dentro de bash3). En este bash, como se puede observar, la estructura seguida para calcular los valores de la función de distribución de cada uno de los niveles (tanto en origen como destino), tomando las direcciones/puertos que nos especifican en el enunciado y que se nos dan en la traza, es la misma. Hemos optado por realizar una llamada a tshark para cada uno de ellos. Es cierto que, desde el punto de vista de optimización, es un tanto más lenta su ejecución; pero, desde el punto de vista visual tanto como desde el punto de vista lectivo, nos ha parecido mucho más conveniente realizarlo de esta manera. En cada una de las llamadas realizadas a tshark lo que hacemos es obtener una columna con todos los tamaños de los paquetes (mediante el comando **-e frame.len** para nivel 2 y **-e ip.len** para el nivel 3, ya que no queremos los datos de la cabecera de nivel 2) e introducimos el filtro de visualización específico, a través del comando **-Y '<filtro de visualización>'** (-Y 'campo a comparar *eq* valor buscado'*)* en cada uno de los casos (en el nivel 2 la MAC de destino/origen; en el nivel 3 para los paquetes HTTP el puerto de destino/origen TCP; y en el nivel 3 para los paquetes DNS el puerto de destino/origen UDP). Estos datos los volcaremos a un fichero .txt para que en la línea siguiente se pueda utilizar para hallar la probabilidad de cada tamaño de cara a la función de distribución.

Y por otra, abordaremos el código de las ECDFs de los tiempos entre llegadas de paquetes y el caudal de ancho de banda (código que tenemos dentro de bash4). En el caso de los tiempos entre llegadas de paquetes, usaremos la misma estructura que hemos empleado para calcular los tamaños, pero con la excepción de que ahora no queremos obtener e tamañao de los paquetes, sino el tiempo transcurrido desde el anterior paquete mostrado (mediante el comando **-e frame.time_delta_displayed).** Por el contrario en el caso del caudal de ancho de banda lo que haremos será obtener el tiempo de captura del paquete desde el comienzo (mediante el comando **– e frame.time_relative**) y lo que hacemos en los dos siguientes awk es obtener por cada segundo el número de bytes por cada sentido (usando como dirección Ethernet la indicada en el generador de traza).

En cada uno de los scripts al comienzo borramos todos los .txt que usamos como "intermediarios" para la obtención de los datos. Esto lo hacemos para evitar que al volver a ejecutar se sobrerrescriban los datos dentro de un mismo archivo y, por tanto, se tomen valores incorrectos.

3.Conclusiones:

La realización de esta práctica nos ha servido para poder aplicar y poner en práctica gran parte de los conocimientos adquiridos durante el curso de redes de comunicación I. Al haber hecho el análisis de rendimiento y estado de la red (monitorización de una red),hemos podido ver y entender el comportamiento de los paquetes a través de la Inthernet, así como hemos podido observar cúal es la distribución que siguen estos paquetes en distintos niveles y en función de distintos parámetros (tamaños y tiempos). De igual forma, hemos podido comprobar cuáles eran las direcciones y puertos más utilizados en cuanto a paquetes y bytes, pudiendo identificar algunos de ellos como los más usados hoy en día (como son el puerto 53 en los protocolos UDP para la consulta de la direcciones o nombres de dominio; o el uerto 80 en los protocolos TCP que es el que escucha o envía la solicitud de servicio hecha por la PC cliente).

Esta práctica nos ha permitido realizar la monitorización desde dos puntos de vista funcional distintos. Por un lado, hemos llevado a cabo la monitorazación activa de la red para la medición de ciertos parámetros, como son el ancho de banda, retardos..., que afectan al comportamiento de dicha red. Por otro lado, la monitorización pasiva de la red que mide parámetros que no afectan al estado de la red, tales como los tamaños de los paquetes, los tiempos entre llegadas..., es decir, información de flujo.

Hemos conseguido desempeñar el papel de un gestor de redes al conseguir analizar el tráfico de la red de forma satisfactoria.