Redes de Comunicaciones I – Prácticas 2020
Práctica 3: Análisis de tráfico

Turno y pareja: 1322 | 15

Integrantes:

Federico Perez Fernandez

Alejandro Tejada Paredes

Fecha de entrega:

Contenido

1 Introducción

En esta tercera y última práctica de REDES I nos centramos en la monitorización de la red. Este concepto consiste en analizar el rendimiento y el estado de la red mediante los datos que generamos gracias a una serie de instrucciones que ejecutamos en Wireshark vía tshark.

Para llevar a cabo este estudio pasivo de la red, nos proporcionan un archivo con una captura previa p3.pcap, un manual de instrucciones para el uso de tshark y una serie de datos como la dirección MAC que se generan gracias al ejecutable generadorPCAPx64 el cual asegura que usemos unos datos exclusivos para nuestra pareja.

Los ejercicios a realizar son un ejemplo de lo que un gestor de red debería de hacer para buscar en la red posibles ataques, ineficiencias o problemas. Estos inconvenientes se encuentran gracias a examinar mediante su representación gráfica dichos datos como los puertos UDP o TCP y direcciones IP o ETH más populares y el ancho de banda que se consume en un periodo de tiempo entre otros.

2 Realización de la práctica

1. Análisis de protocolos.

   Obtener los porcentajes de paquetes IP y NO IP (entendemos como **NO-IP** aquellos paquetes que no son ni **ETH|IP** ni **ETH|VLAN|IP**)

% Paquetes IP	% Paquetes NO-IP
99.3%	0.7%

Para este apartado hemos utilizado los dos siguientes filtros:
- tshark -r p3.pcap -T fields -e frame.number -Y 'ip'
- tshark -r p3.pcap -T fields -e frame.number -Y '!ip'

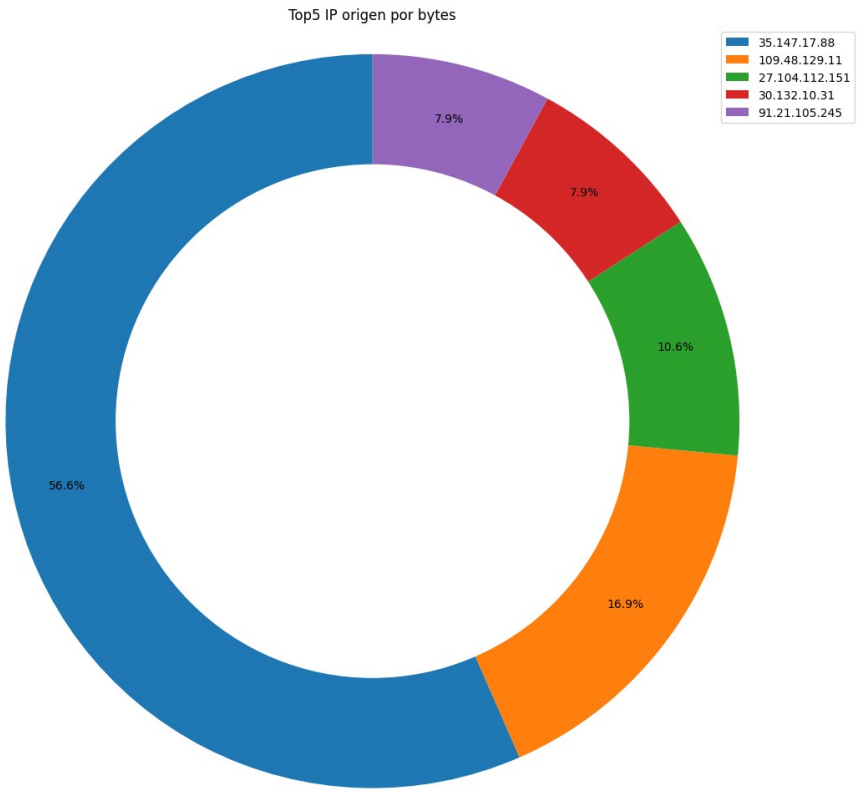
El primero obtiene el número de paquetes que son IP y el segundo obtiene los que no son IP, a continuación, calculamos sus porcentajes y se lo pasamos como dato al método que nos pintara la gráfica deseada. Como podemos observar la gran mayoría de paquetes que han sido capturados son de tipo IP y solo una pequeña minoría son NO-IP.

   Obtener los porcentajes de paquetes UDP, TCP y OTROS sobre los que son IP (igualmente entienda, un paquete IP como aquel que cumpla la pila **ETH|IP** o **ETH|VLAN|IP**).

Paquetes TCP	% Paquetes UDP	% Paquetes OTROS
58,4%	5,4%	34,2%

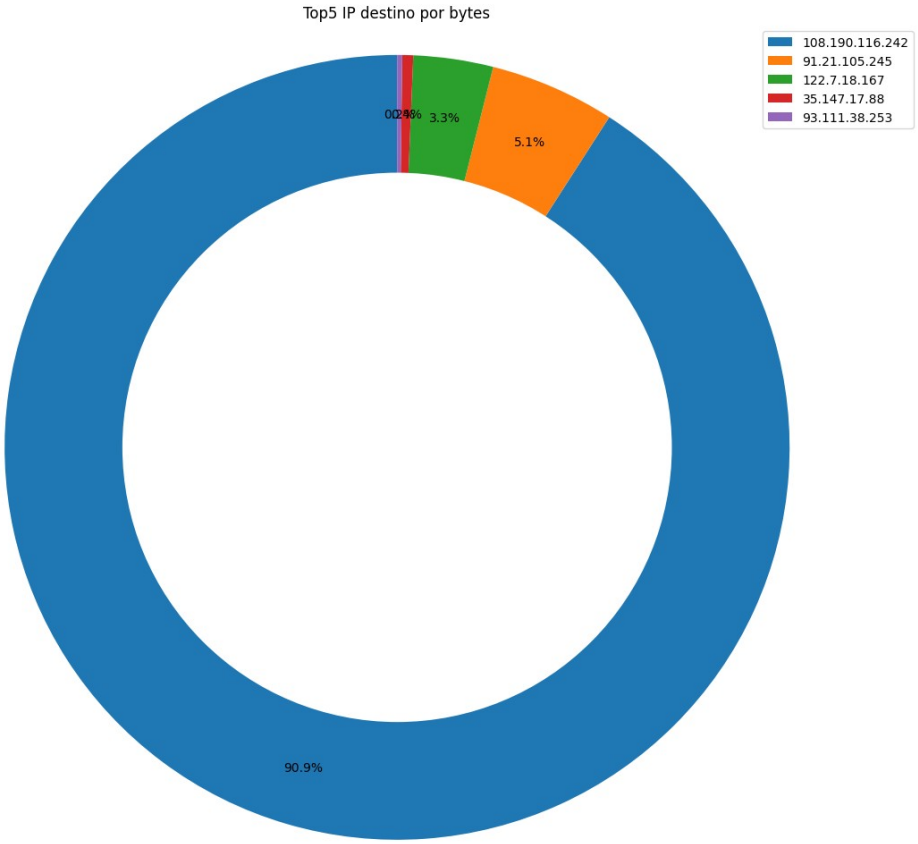
Para este apartado hemos utilizado el siguiente filtro:
- tshark -r p3.pcap -T fields -e frame.protocols
Este filtro se encarga de obtener el protocolo de cada uno de los paquetes de la captura, el código luego se encarga de hacer la cuenta del número de paquetes que son de tipo TCP, UDP u otros.

2. Obtención de top 5 de direcciones IP



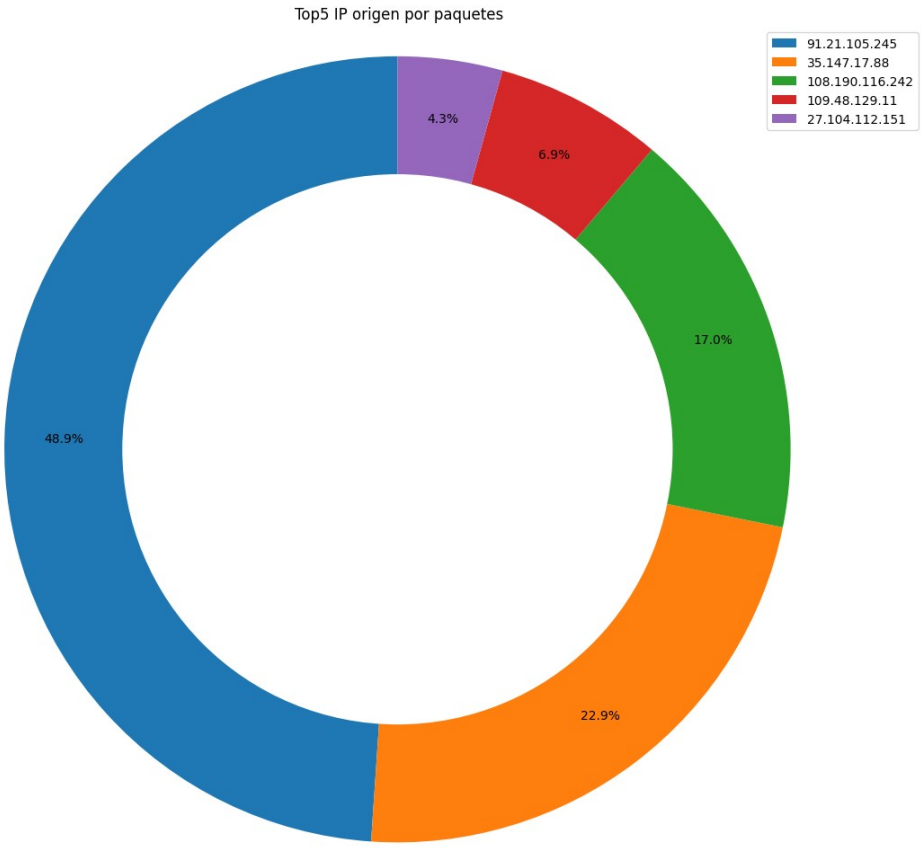
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e ip.src -e frame.len -Y 'ip'
Con este filtro obtenemos la dirección ip origen y el tamaño de cada paquete, que iremos almacenando en un diccionario, en caso de que la ip que es la clave se encuentre ya en el diccionario, el programa actualiza el valor que le corresponde a esa clave sumando lo el contenido nuevo al contenido antiguo, finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Como podemos observar el 56.6% de los bytes de todos los paquetes que tienen como dirección ip origen alguna de las direcciones del top5 le corresponde a la ip 35.147.17.88 esto quiere decir que esta ip para esta captura de tráfico es la genera paquetes en su conjunto más ‘’pesados’’ en la red. La segunda ip es la 109.48.129.11 con un 16.9%, la tercera 27.104.112.151 con un 10.6%, la cuarta 30.132.10.31 con un 7.9% y la quinta 91.21.105.245 con un 7.9%.



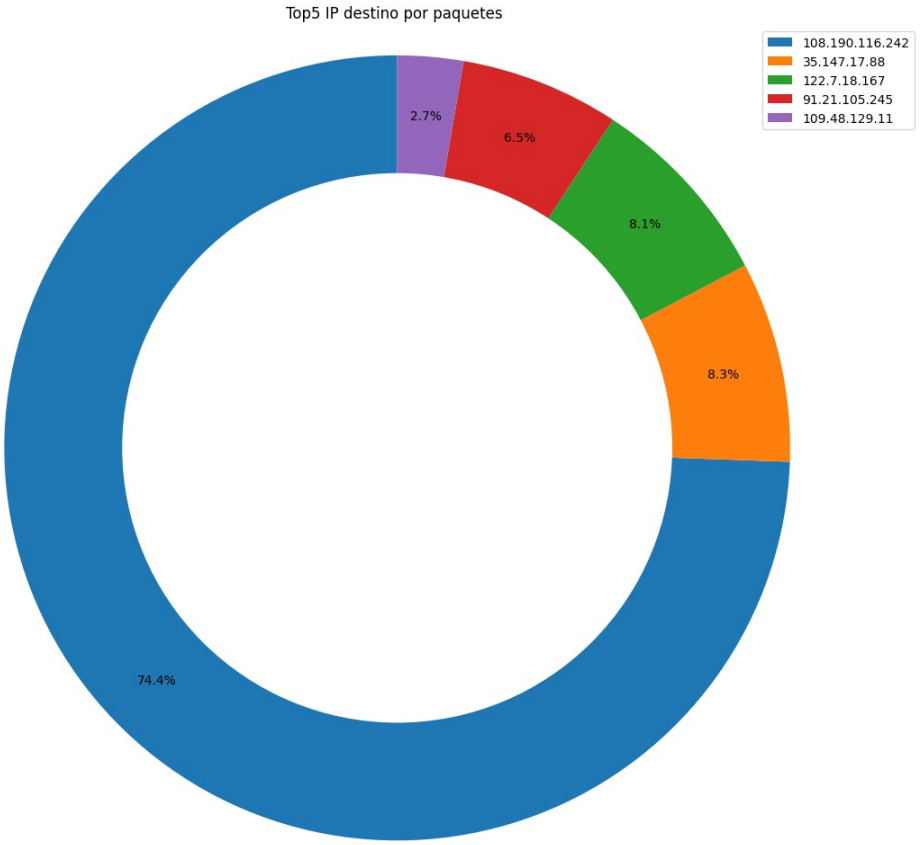
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e ip.dst -e frame.len -Y 'ip'
Con este filtro obtenemos la dirección ip destino y el tamaño de cada paquete, que iremos almacenando en un diccionario, en caso de que la ip que es la clave se encuentre ya en el diccionario, el programa actualiza el valor que le corresponde a esa clave sumando lo el contenido nuevo al contenido antiguo, finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Como podemos observar el 90.9% de los bytes de todos los paquetes que tienen como dirección ip destino alguna de las direcciones del top5 le corresponde a la ip 108.190.116.242 esto quiere decir que esta ip para esta captura de tráfico es la que recibe los paquetes en su conjunto más ‘’pesados’’ en la red.La segunda ip es la 91.21.105.245 con un 5.1%, la tercera 127.7.18.167 con un 3.3%, la cuarta 35.147.17.88 y la quinta 93.111.38.253.



El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e ip.src -Y 'ip'
Con este filtro obtenemos la dirección ip origen de cada paquete, que iremos almacenando en un diccionario, donde la clave será la ip y el valor el número de paquetes que tienen como ip origen esa clave, un contador, básicamente. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Como podemos ver la ip que genera más paquetes para enviar en la captura es la ip 91.21.105.245 con un 48.9% de los paquetes del total de paquetes del top5 de paquetes que tienen como ip origen alguna ip del top5.La segunda ip es la 35.147.17.88 con un 22.9%, la tercera 108.190.116.242 con un 17.0%, la cuarta 109.48.129.11 con un 6.9% y la quinta 27.104.112.151 con un 4.3%.

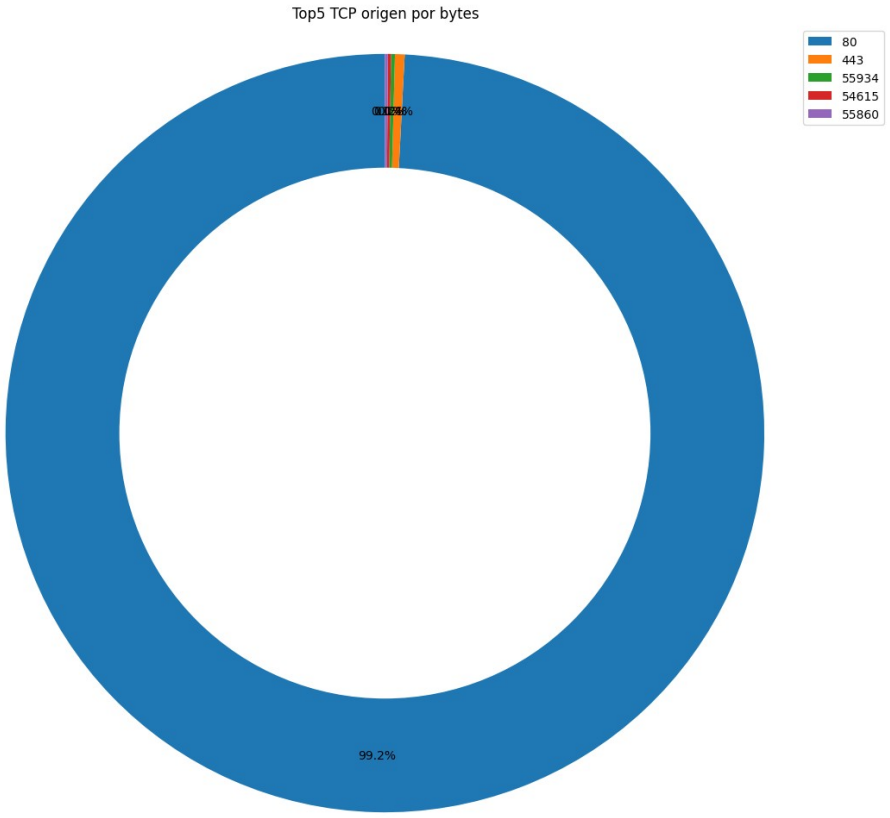


El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e ip.dst -Y 'ip'
Con este filtro obtenemos la dirección ip destino de cada paquete, que iremos almacenando en un diccionario, donde la clave será la ip y el valor el número de paquetes que tienen como ip destino esa clave, un contador, básicamente. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Como podemos ver la ip que genera más paquetes para enviar en la captura es la ip 108.190.116.242 con un 74.4% de los paquetes del total de paquetes del top5 de paquetes que tienen como ip origen alguna ip del top5. Vemos ademas que esta ip es la misma ip que tiene en conjunto le mayor peso de paquetes de la captura. La segunda ip es la 35.147.17.88 con un 8.3%, la tercera 122.7.18.167 con un 8.1%, la cuarta 91.21.105.245 con un 6.5% y la quinta 109.48.129.11 con un 2.7%.

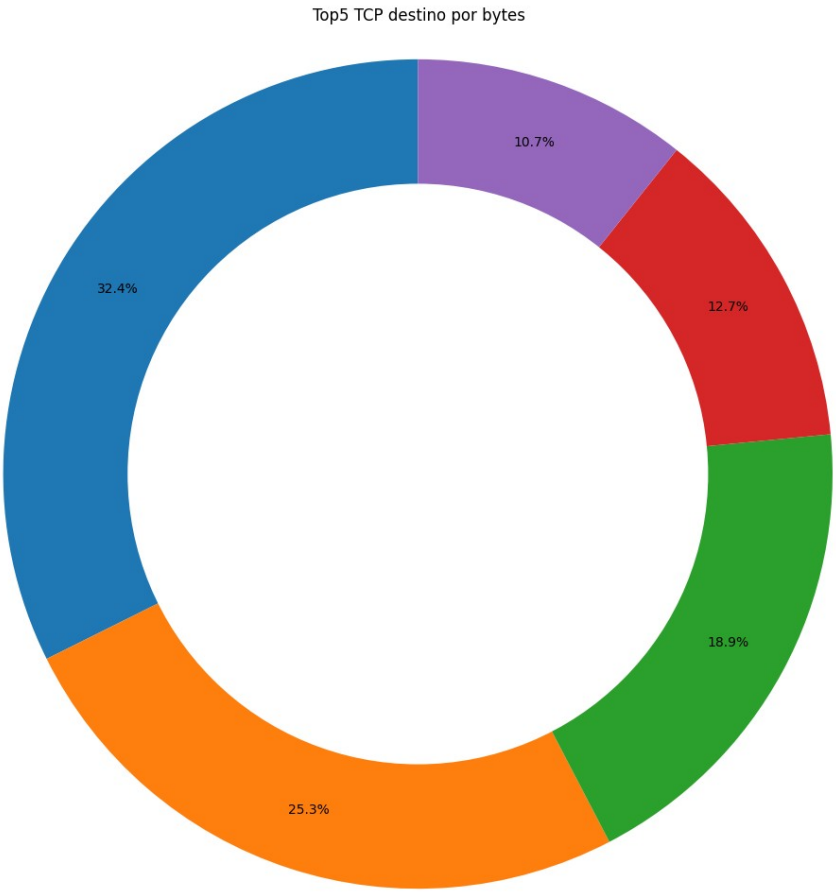
3. Obtención de top 5 de puertos:

TCP:



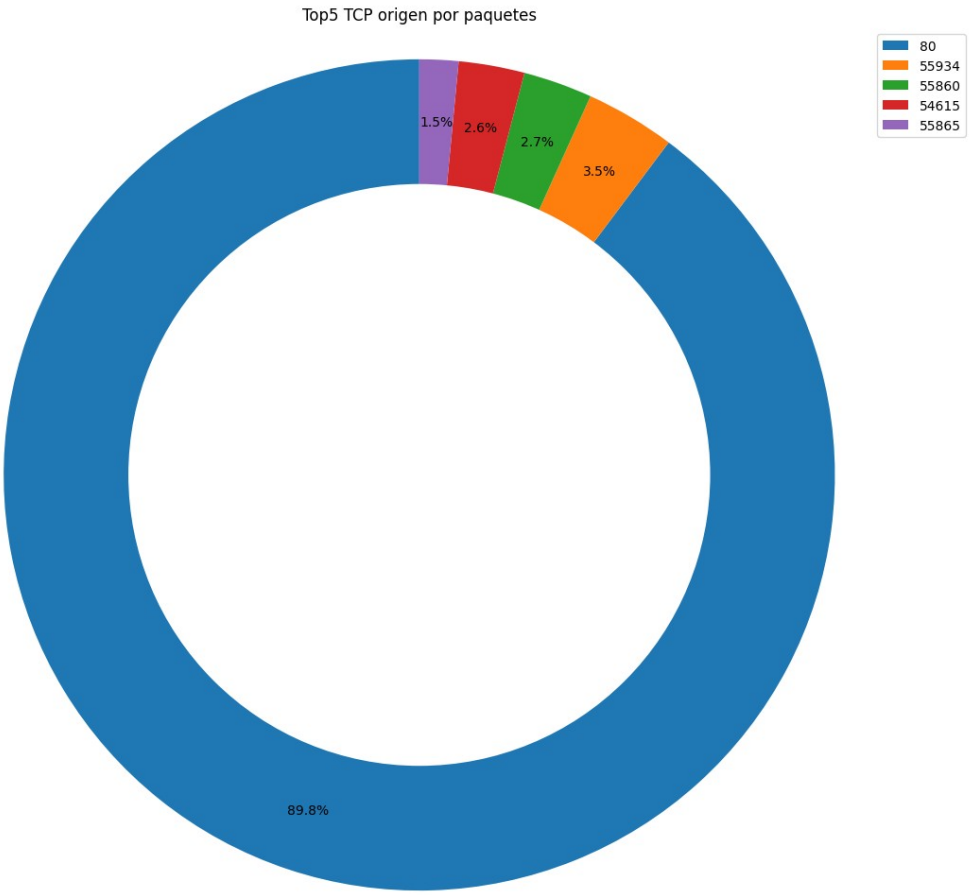
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e tcp.srcport -e frame.len -Y 'tcp'
Con este filtro obtenemos el puerto TCP origen y el tamaño de cada paquete, que iremos almacenando en un diccionario, donde la clave será el puerto y el valor la cantidad de bytes del total de paquetes de ese puerto. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Como podemos ver en la grafica, del puerto 80 sale la mayor cantidad de bytes de todos los paquetes de la captura con un 99.2%, el segundo es el 433, el tercero es el 55934, el cuarto es el 54615 y el quinto es el 55860.



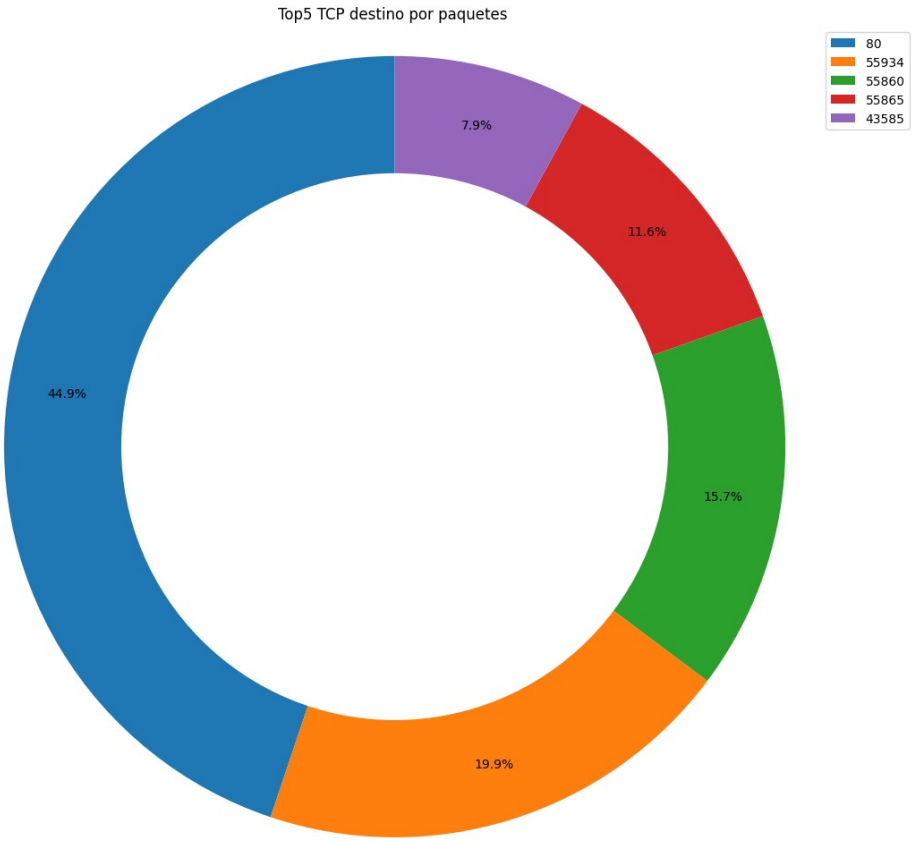
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e tcp.dstport -e frame.len -Y 'tcp'
Con este filtro obtenemos el puerto TCP destino y el tamaño de cada paquete, que iremos almacenando en un diccionario, donde la clave será el puerto y el valor la cantidad de bytes del total de paquetes de ese puerto. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Para la llegada de paquetes TCP en medidas de bytes como podemos ver en la gráfica se ve una mayor repartición entre los puertos destino, el puerto 55934 es en medidas de bytes más cantidad tiene con un 32.4%, el segundo es el 55860 con un 25,3%, el tercero 55865 con un 18.9%, el cuarto el 43585 con un 12.7% y el quinto es el 54615 con un 10.7%.



El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e tcp.srcport -Y 'tcp'
Con este filtro obtenemos el puerto TCP origen y vamos incrementando el valor de la clave puerto TCP cada vez que leemos un paquete que tenga como puerto origen esa clave. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

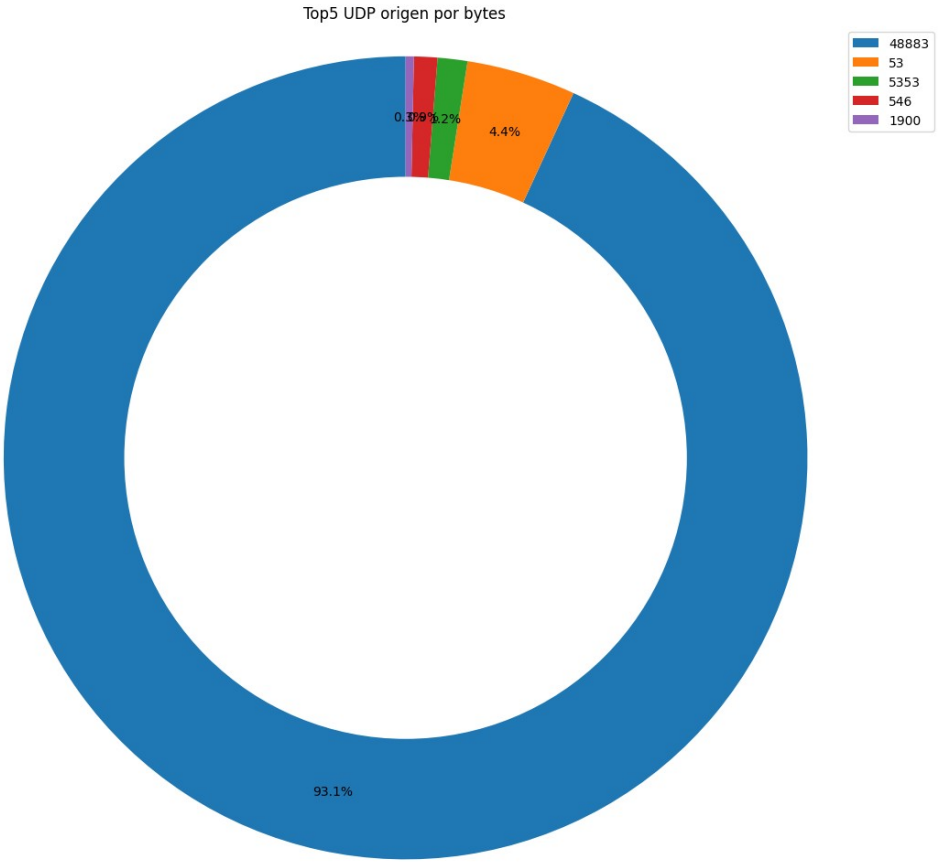
Vemos en la gráfica que el puerto 80 es el puerto que más envía paquetes en la captura, además, como ya hemos visto antes, este mismo puerto es el que más carga de bytes tiene. El segundo es el puerto 55934 con un 3.3%, el tercero el 55860 con un 2.7%, el cuarto el 54615 con un 2.6% y el quinto el 55865 con un 1.5%.



El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e tcp.dstport -Y 'tcp'
Con este filtro obtenemos el puerto TCP destino y vamos incrementando el valor de la clave puerto TCP cada vez que leemos un paquete que tenga como puerto destino esa clave. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

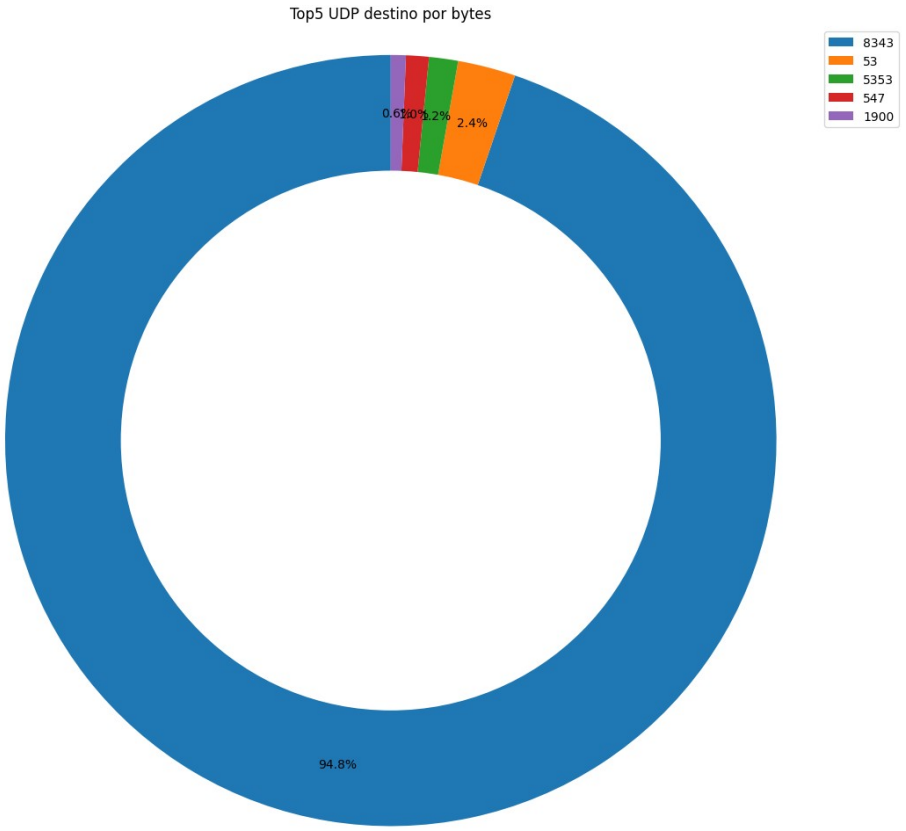
En esta gráfica vemos que el puerto que recibe más paquetes es el puerto 80 con un 44.9%, el segundo es el puerto 55934 con un 19.9%, que como ya se ha mencionado era el puerto destino TCP que tenia más carga de bytes, el tercero es el puerto 55860 con un 15.7%, el cuarto el 55865 copn un 11.8% y el quinto el 43585 con un 7.9%.

UDP:



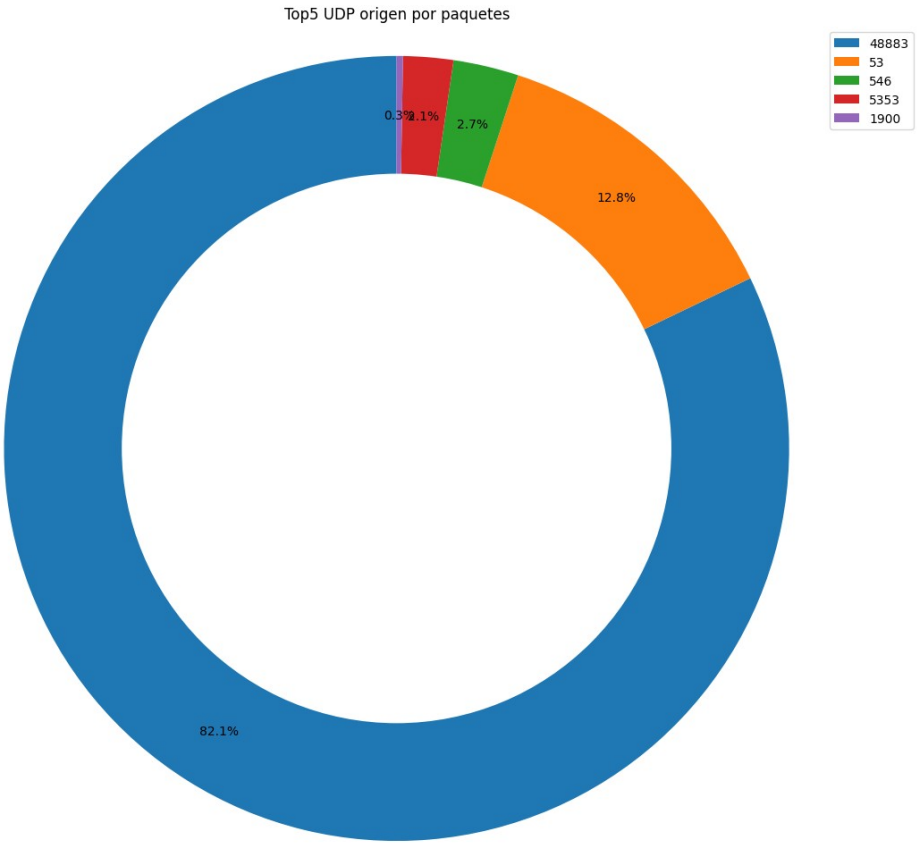
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e udp.srcport -e frame.len -Y 'udp'
Con este filtro obtenemos el puerto UDP origen y el tamaño de cada paquete, que iremos almacenando en un diccionario, donde la clave será el puerto y el valor la cantidad de bytes del total de paquetes de ese puerto. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Para la salida de paquetes UDP en medidas de bytes como podemos ver en la gráfica se ve que el puerto 48883 es el que acapara los bytes con un 93,1%, el segundo es el 53 con un 4,4%, el tercero 5353, el cuarto el 546 y el quinto es el 1900.



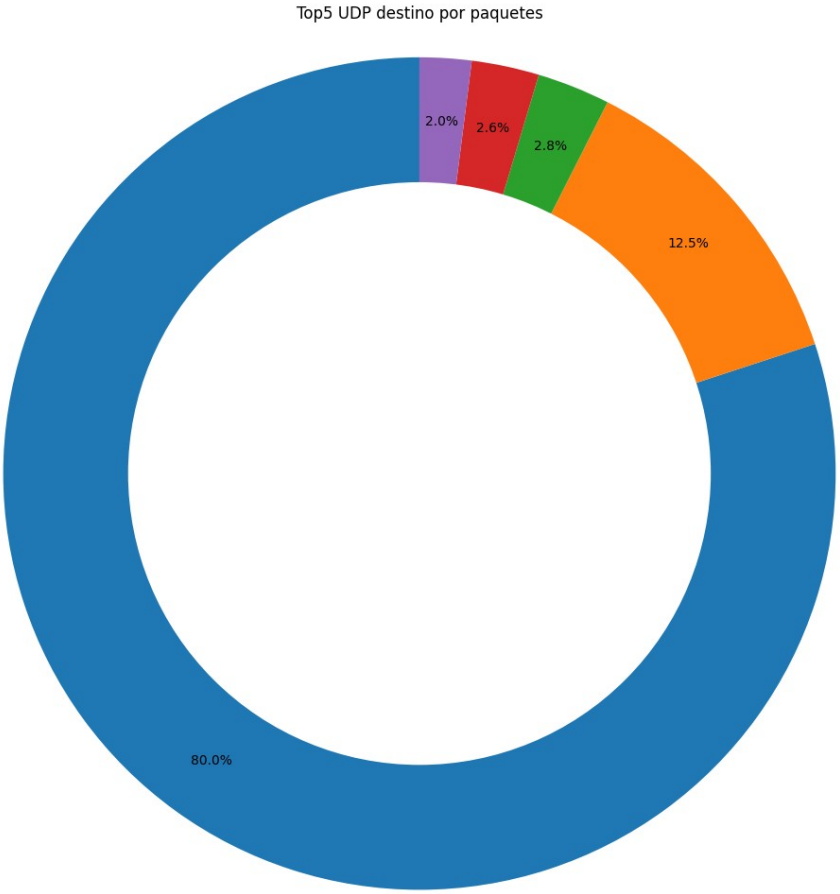
El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e udp.dstport -e frame.len -Y 'udp'
Con este filtro obtenemos el puerto UDP destino y el tamaño de cada paquete, que iremos almacenando en un diccionario, donde la clave será el puerto y el valor la cantidad de bytes del total de paquetes de ese puerto. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

Para la llegada de paquetes UDP en medidas de bytes como podemos ver en la gráfica se ve que el puerto 8343 es el que acapara los bytes con un 94,8%, el segundo es el 53 con un 2,4%, el tercero 5353, el cuarto el 547 y el quinto es el 1900.



El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e udp.srcport -Y 'udp'
Con este filtro obtenemos el puerto UDP origen y vamos incrementando el valor de la clave puerto UDP cada vez que leemos un paquete que tenga como puerto origen esa clave. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

En esta gráfica vemos que el puerto que envía más paquetes es el puerto 48883 con un 82.1%, que además corresponde con el puerto que tiene más carga en bytes de paquetes, el segundo es el puerto 53 con un 12.8%, el tercero es el puerto 546 con un 2.7%, el cuarto el 5353 con un 2.1% y el quinto el 1900 con un 0.3%.

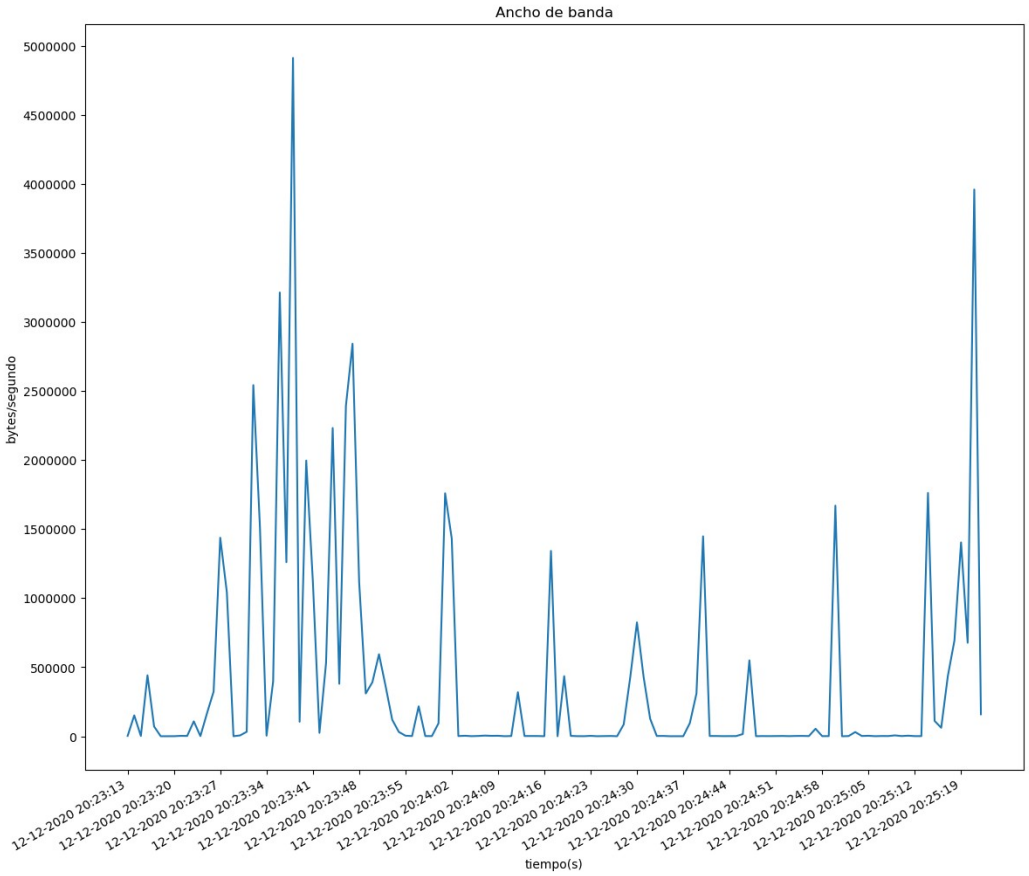


El filtro utilizado para la obtención de los datos es el siguiente:
- tshark -r p3.pcap -T fields -e udp.dstport -Y 'udp'
Con este filtro obtenemos el puerto UDP destino y vamos incrementando el valor de la clave puerto UDP cada vez que leemos un paquete que tenga como puerto destino esa clave. Finalmente obtenemos el top5, teniendo en cuenta el valor de las claves, del diccionario y le pasamos esos datos al método pintarTarta.

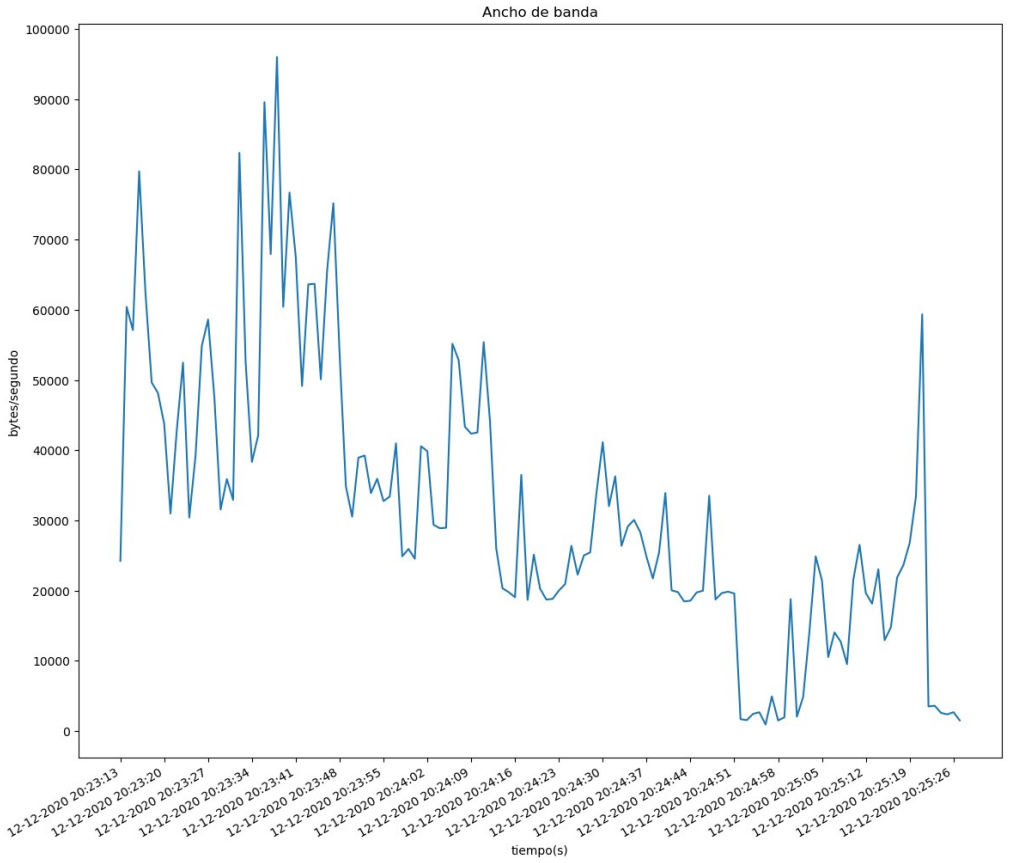
En esta gráfica vemos que el puerto que recibe más paquetes es el puerto 8343 con un 80%, que además corresponde con el puerto UDP destino que tiene más carga en bytes de paquetes, el segundo es el puerto 53 con un 12.5%, el tercero es 5353 puerto 5355 con un 2.8%, el cuarto el 547 con un 2.6% y el quinto el 1900 con un 2%.

4. Series temporales de ancho de banda/tasa/caudal:

tshark -r {} -T fields -e frame.number -e frame.len -e frame.time_epoch -Y \'eth.src == 00:11:88:CC:33:78\'
Cambiamos la palabra subrayada por dst para obtener el ancho de banda cuando la MAC es de destino

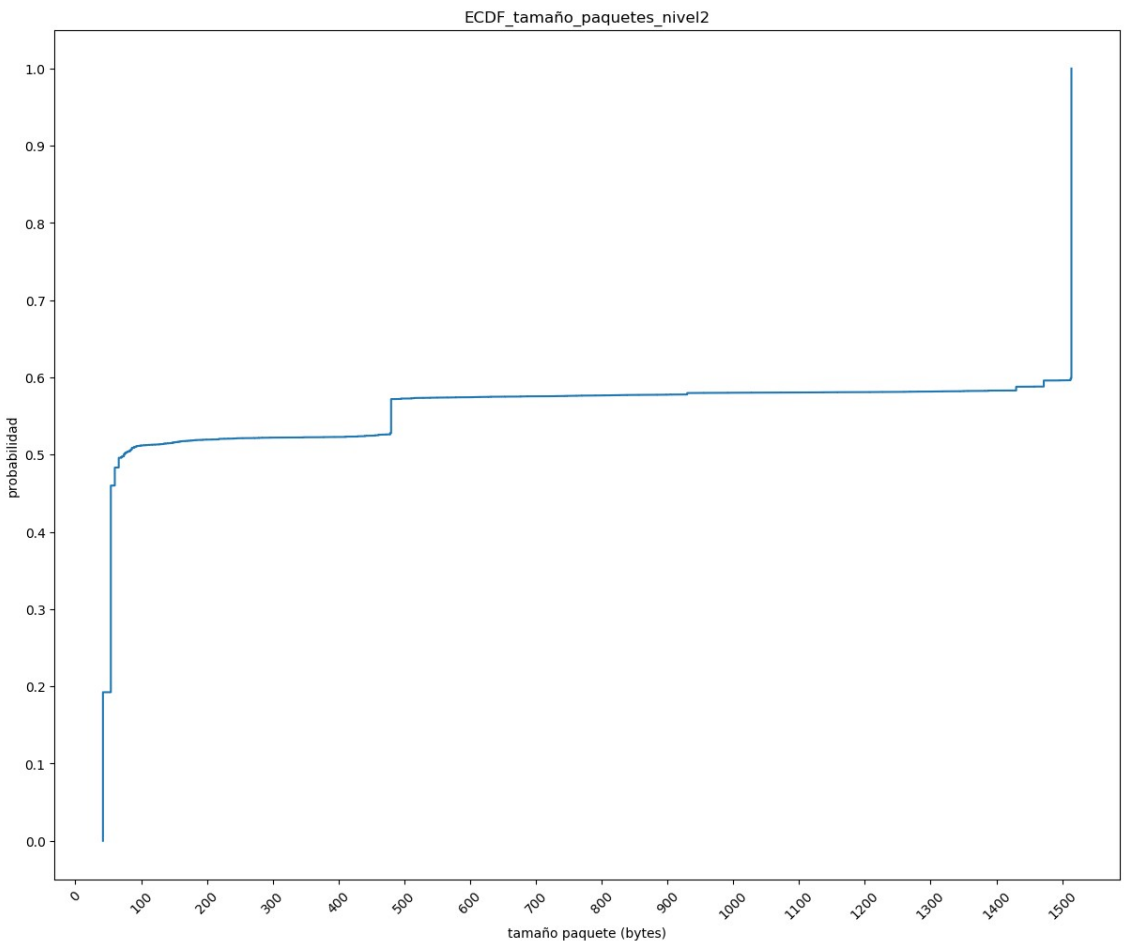


En esta gráfica observamos como el ancho de banda no es constante llegando a tener momentos donde el paso de paquetes por la red es nulo ya que nuestra dirección IP no envía constantemente paquetes a través de la red.
Se aprecia a su vez como el tráfico en la red puede ser especialmente alto durante periodos muy cortos de tiempo indicando una gran cantidad de paquetes más pequeños o posiblemente la aparición de un paquete más pesado.



A diferencia del análisis anterior, en este caso se puede apreciar como el ancho de banda tiende a disminuir de forma controlada, no existen tramos de tiempo donde el tráfico de paquetes sea nulo.
Podríamos intuir por la forma del datagrama en los últimos periodos de tiempo (20:24:51 en adelante) que el usuario ingresó en una página como por ejemplo Youtube, por lo que encontramos una crecida progresiva del tráfico en la red (carga de miniaturas, previews, etc) pasando a la selección y visualización de un video para terminar cerrando el navegador y cortando así de golpe toda petición y movimiento de paquetes.

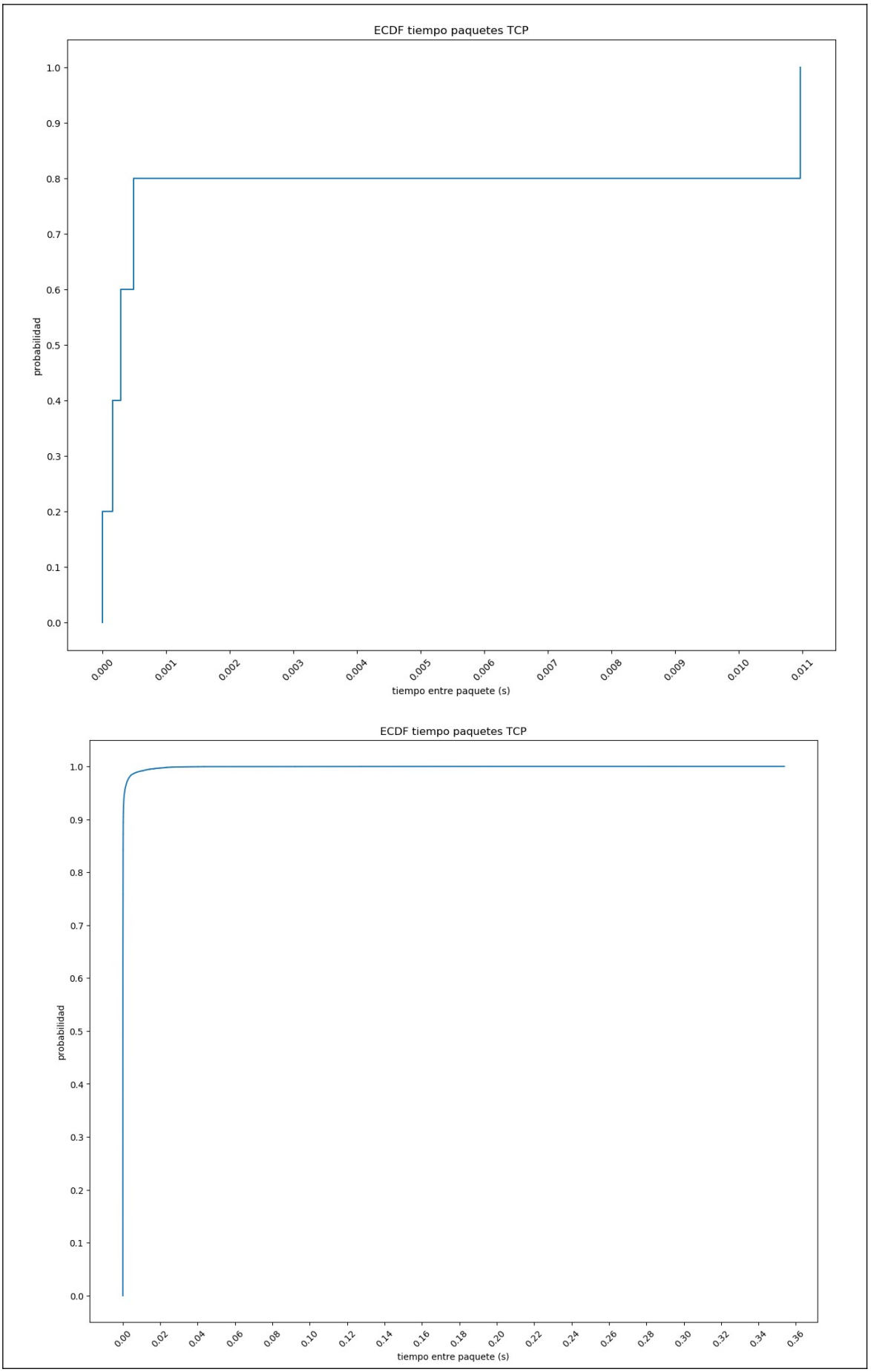
5. ECDFs de los tamaños de los paquetes



El filtro utilizado es el siguiente:
- tshark -r p3.pcap -T fields -e frame.number -e frame.len -e frame.time_epoch -Y 'eth.dst == 00:11:88:CC:33:78

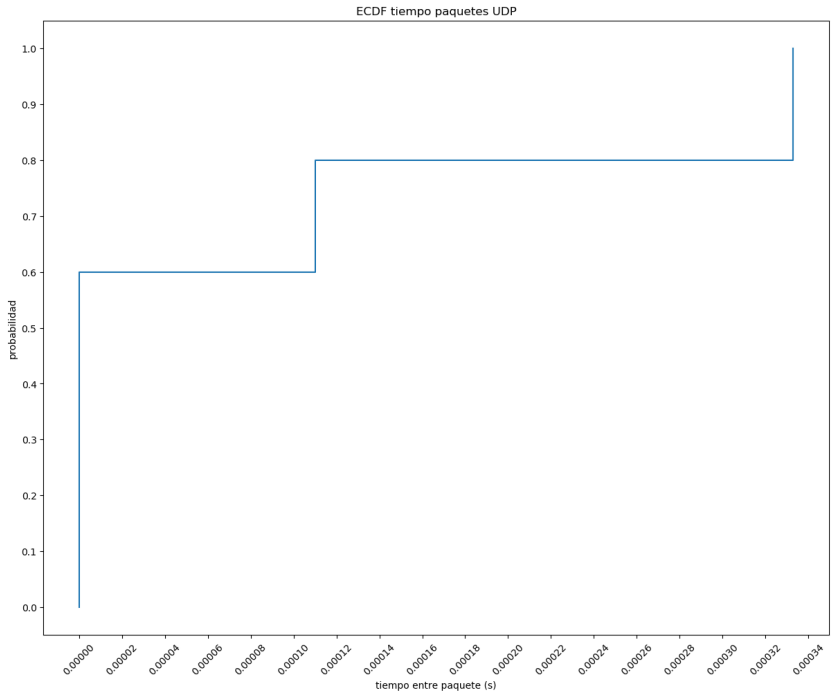
Con esta grafica podemos observar como la mayoría de los paquetes de la traza pcap tiene un tamaño de entre 40-50 bytes o más de 1500 bytes ya que suponen aproximadamente un 88% entre los dos (44% y 44% respectivamente). Con esto podríamos afirmar que esta traza representa un periodo donde se han obtenido paquetes muy pequeños o muy grandes y apenas paquetes medianos de entre 100 y 1500 bytes.

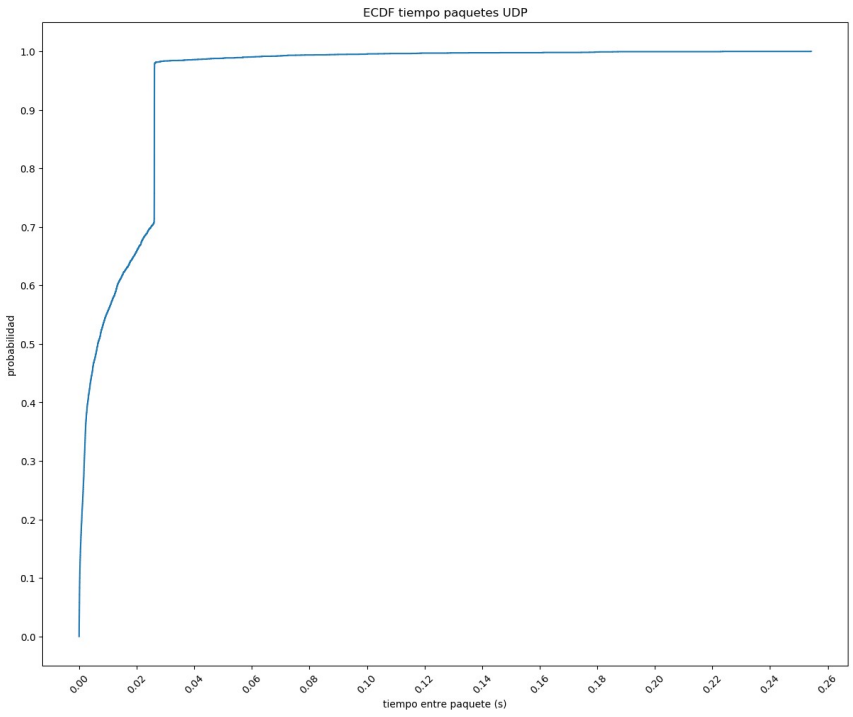
6. ECDF tiempos entre paquetes


ECDF tiempo paquetes TCP
probabilidad
tiempo entre paquete (s)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.010
0.011
ECDF tiempo paquetes TCP
probabilidad
tiempo entre paquete (s)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.24
0.26
0.28
0.30
0.32
0.34
0.36

El filtro utilizado es el siguiente:
- tshark -r p3.pcap -T fields -e frame.time_delta -Y 'tcp'

Con esta grafica podemos observar como gran parte del tiempo de llegada entre paquetes varia entre 0.000 y 0.0005 segundos, suponiendo un 80% del total de las muestras.
No obstante, estos valores solo representan 5 muestras del total ya que si generamos esta misma grafica con todas las muestras de la traza el resultado obtenido es de prácticamente un 100% de obtener un paquete en 0.00 segundos. Esto se debe a que la gráfica no cuenta con suficiente precisión pero es complicado determinar la escala oportuna ya que existen valores desde 0.1 hasta 10^{-7} segundos.





El filtro utilizado es el siguiente:
- tshark -r p3.pcap -T fields -e frame.time_delta -Y 'udp'

Con esta gráfica podemos observar como el tiempo entre paquetes con puerto origen o destino UDP se encuentra en su gran mayoría entrono a 0.00 y 0.03 segundos. Esto es bastante lógico ya que cuentan con una cabecera más ligera (menos información) y, aunque para ciertos usos como la retransmisión de datos de un recurso en streaming sea idoneo, en el momento en que se precisa de una menor cantidad de tiempo entre

paquetes se puede observar con la grafica anterior de un protocolo con más información sobre el destinatario como es el caso de TCP, como este contempla mayores cantidades de tiempo entre recepciones.

3 Conclusiones

Gracias a esta entrega hemos confirmado varios conceptos aprendidos en clase como la diferencias entre protocolos, uso del ancho de banda y trafico de red.

Por otro lado, hemos hecho uso de una nuevo tipo de datagrama, el *ECDF* por el cual podemos saber el porcentaje de una variable aleatoria mediante el numero de veces que aparece en el muestrario.

Consideramos que esta práctica ha sido muy útil para conocer el trabajo que realiza a *grosso modo* un gestor de redes y a su vez entretenida con una cierta dificultad añadida por el uso de tshark y los datagramas.