Redes de Comunicaciones I – Prácticas 2020

Práctica 3: Análisis de tráfico

Turno y pareja: L: 11-13, 1392\_01

Integrantes:

Carlos García Toledano

S. Xiao Fernández Marín

Fecha de entrega: 13/12/2020

Contenido

[1Introducción 2](#__RefHeading___Toc677_1860594804)

[2Realización de la práctica 2](#__RefHeading___Toc679_1860594804)

[3Conclusiones 5](#__RefHeading___Toc683_1860594804)

# Introducción

En esta práctica vamos a monitorizar una red, es decir, analizar el rendimiento de una red dada con el fin de obtener diversas métricas y estudiar su funcionamiento y rendimiento. Lo vamos a hacer de manera pasiva, analizando el tráfico que está atravesando.

La traza utilizada en esta práctica va a ser la obtenida mediante el “generadorPCAPx64”, dado en clase, donde se simula el tráfico entre 2 routers que transportan información.

Al obtener esta traza, haremos como un gestor de red y obtendremos métricas para comenzar a estudiar el tráfico.

En este trabajo analizaremos, desde la traza dada, qué IPs son las más utilizadas al igual que qué puertos son los que más se utilizan en los protocolos TCP y UDP gracias a gráficas que muestran el top 5 IPs y puertos en bytes y paquetes.

Tras esto, estudiaremos las series temporales del caudal/thoughput/tasa/ancho de banda en el ethernet cuando es origen y cuando es destino.

Por último, veremos también qué diferencia hay entre estos dos protocolos mencionados anteriormente y el cómo afecta esto al tiempo entre paquetes de su flujo también guiándonos por graficas hechas en el script y la teoría aprendida en clase.

# 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 |
| 88.03533144670268 | 11.964668553297315 |

|  |
| --- |
| **'tshark -r {} -T fields -e frame.number eth.type == 0x0800'.format(args.tracefile)**  Los paquetes IP son un 88.03533144670268% frente a los no IP que son un 11.964668553297315%  Las IP son calculadas con el comando superior, donde entre corchetes metemos el .pcap obtenido con el generador. Te devuelve un frame number de las IP con ethernet = 0x0800 donde cada línea es una IP diferente, por lo que contamos las líneas y luego calculamos su porcentaje. |

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 |
| 96.19902978822887 | 2.308636443336126 | 0.0019482164078785873 |

|  |
| --- |
| **'tshark -r {} -T fields -e tcp.port eth.type == 0x0800'.format(args.tracefile)**  Para obtener el porcentaje de paquetes TCP utilizamos el comando de arriba, donde nos devuelve los puertos TCP con el ethernet type = 0x0800, ya que se entiende como paquete IP todo aquel cuyo tipo etheret sea el mencionado.  **'tshark -r {} -T fields -e udp.port eth.type == 0x0800'.format(args.tracefile)**  A diferencia que, en el comando anterior, nos devuelve los puertos UPD.  **'tshark -r {} -T fields -e udp.port -e tcp.port eth.type == 0x0800'.format(args.tracefile)**  Este comando nos da los puertos UDP y TCP que son IP.  La suma nos da 98.5096144479728745873. Este error se puede dar porque …??? |

1. Obtención de top 5 de direcciones IP

|  |
| --- |
| **'tshark -r {} -T fields -e ip.src -e frame.len'.format(args.tracefile)**  Nos devuelve los paquetes que manda a las ips destino con su tamaño en bytes, una por cada línea.  Lo separamos y lo ordenamos, luego, guardamos en un diccionario la dirección IP junto con la cantidad de bytes que han sido mandados. Luego, para un bucle con limite 5, porque solo queremos los 5 primeros bytes, vamos metiendo en dos listas diferentes el numero de la IP y las veces que se ha repetido para más tarde, llamar a pintarTarta |
| Se trata de una gráfica que representa el top 5 de IPs origen por bytes que se han podido recoger en la traza dada.  Como podemos observar, más de la mitad de la gráfica es representada por la IP 53.115.149.152. Esto significa que la IP que más bytes ha mandado ha sido esta misma. |
| **'tshark -r {} -T fields -e ip.dst -e frame.len'.format(args.tracefile)**  En este caso, hemos hecho lo mismo que en la gráfica Top IPs origen por bytes, pero en vez de haber puesto en el primer comando ip.src, hemos puesto ip.dst. |
| Se trata de una gráfica que representa el top 5 de IPs destino en bytes que se han podido recoger en la traza dada.  Como podemos observar, el 91’6% de esta gráfica está ocupada por la IP 63.127.54.15, ubicada en [EEUU](https://www.ntunhs.net/cgi-bin/whois20_1.cgi). |
| **'tshark -r {} -T fields -e ip.src'.format(args.tracefile)**  Nos devuelve los paquetes que manda a las ips destino, una por cada línea.  Lo separamos y las guardamos en un nuevo archivo para luego, con **uniq -c salida\_ip\_src\_paquetes.dat**, averiguar cuantas veces una ip ha sido llamada.  Esto nos devolverá lo que pone en el archivo que ordenaremos de mayor a menor, luego, volvemos a guardar cada línea en una lista.  Cuando esto ya está hecho, cogemos los 5 primeros y separamos el número de IPs que hay y el nombre de la IP. Estas dos cosas las metemos en dos listas diferentes como hemos hecho anteriormente y llamamos a la función pintarTarta. |
| Se trata de una gráfica que representa el top 5 de IPs origen por paquetes que se han podido recoger en la traza dada.  Como podemos observar, la IP dominante es 53.115.149.152, casualmente es la misma que se encuentra en el puesto 1 del top 5 IPs por bytes, aunque en comparación con el mencionado, ocupa un 37’8%. Esta IP pertenece a [Mercedes-Benz, Alemania](https://www.ntunhs.net/cgi-bin/whois20_1_allip3.cgi?HPLang=ES&LV=3&IP=53.115.149.152).  Vemos que esto no tiene por qué ser así siempre, ya que las otras 4 IP que entran dentro del top 5, no son las mismas que se han recogido cuando hemos hecho la gráfica por bytes. |
| **'tshark -r {} -T fields -e ip.dst'.format(args.tracefile)**  **uniq -c salida\_ip\_dst\_paquetes.dat**  En este caso, hemos hecho lo mismo que en la gráfica Top IPs origen por paquetes, pero en vez de haber puesto en el primer comando ip.src, hemos puesto ip.dst. |
| Podemos ver un diagrama en forma de tarta donde están representado el top 5 de IP por paquetes.  Con una diferencia del 20’8% sobre la gráfica “top 5 destino por bytes”, la IP 63.127.54.15 sigue la primera en la que más paquetes ha recibido.  La IP 44.13.32.11, que era la segunda en la gráfica de bytes, es ahora la cuarta y la que era la tercera IP en la gráfica anterior (118.250.107.190), es ahora la segunda.  Esto confirma que, porque una IP haya recibido más bytes, no tiene por qué haber recibido más paquetes, ya que se podrían haber mandado pocos paquetes muy grandes, lo que haría que hubiese un porcentaje grande en bytes y uno pequeño en paquetes y viceversa. |

1. Obtención de top 5 de puertos:

TCP:

|  |
| --- |
| **'tshark -r {} -T fields -e tcp.srcport -e frame.len'.format(args.tracefile)**  Hemos aplicado la misma lógica que en IP origen por byte pero cambiando ip.dst por tcp.srcport |
| El puerto más utilizado al mandar bytes en TCP es el 80 con un 99’2%, Este puerto es el usado por el protocolo de [transferencia http](https://isc.sans.edu/port.html?port=80) de servicios world wide web. Tiene sentido que sea el mas utilizado con diferencia, ya que es el que te conecta con internet y en esta época, es lo que más utilizamos al día.  El siguiente es el 443, el predeterminado para el protocolo [https](https://es.adminsub.net/tcp-udp-port-finder/443), transferencia de hipertexto seguro.  Los últimos tres son asociados a [apple](https://es.adminsub.net/tcp-udp-port-finder/), más concretamente al acceso a [Xsan Filesystem](https://support.apple.com/es-es/HT202944), una red de área de almacenamiento. |
| **'tshark -r {} -T fields -e tcp.dstport -e frame.len'.format(args.tracefile)**  En este caso, hemos hecho lo mismo que en la gráfica Top TCPs origen por bytes, pero en vez de haber puesto en el primer comando tcp.srcport, hemos puesto tcp.dstport. |
| El primer puerto más usado en el TCP destino es el 5534, asignado a [apple](https://es.adminsub.net/tcp-udp-port-finder/42165) y usado por el acceso a [Xsan Filesystem](https://support.apple.com/es-es/HT202944), una red de área de almacenamiento. Todos los puertos de este grafico son asociados a esto, excepto el 43585, que está [sin asociar](https://es.adminsub.net/tcp-udp-port-finder/43585). Esto quiere decir que la IANA (autoridad de números asociados a internet), los asigna para algún servicio a solicitud de una entidad solicitante que no tiene puertos asociados a ella.  Con esta información y la recogida anteriormente podemos saber que el dispositivo desde el que se están mandando y recibiendo información se trata de un apple, ya que utiliza [puertos específicos para esta empresa](https://support.apple.com/es-es/HT202944).  Este gráfico está mucho mas igualado frente al TCP origen por bytes, ya que el mayor porcentaje aquí es del 32’4% mientras que en el anterior era de un 99’2% |
| **'tshark -r {} -T fields -e tcp.srcport'.format(args.tracefile)**  **uniq -c salida\_tcp\_src\_paquetes.dat**  Hemos hecho el mismo código y comandos que en IP origen por paquetes pero cambiando ip.dst por tcp.srcport |
| El puerto mas utilizado por nuestra red para recibir paquetes es el 80. Este puerto es el asignado por defecto al [http](https://isc.sans.edu/port.html?port=80), un protocolo no seguro para navegar por la web. Tiene sentido que este puerto sea de los más utilizados ya que es el que se usa a navega por la web, cosa que estamos haciendo todo el rato cuando estamos con el ordenador.  El siguiente es el 55934, el cual es un puerto privado y dinámico, asignado a [IANA o apple](https://es.adminsub.net/tcp-udp-port-finder/55934), como pasaba con el tercer puerto del top 5 puertos en bytes TCP. Lo mismo pasa con el tercero, cuarto y quinto.  Podemos ver, al igual que el top 5 TCP origen por bytes, que el primer puesto es usado un 89’3% frente a los demás. |
| **'tshark -r {} -T fields -e tcp.dstport'.format(args.tracefile)**  **uniq -c salida\_tcp\_dst\_paquetes.dat**  Hemos hecho lo mismo que en la gráfica Top TCPs origen por paquetes, pero en vez de haber puesto en el primer comando tcp.srcport, hemos puesto tcp.dstport. |
| En esta grafica podemos ver el top puertos destino por paquetes.  El primero, sin ser ninguna sorpresa, es el puerto 80, el puerto que usa el http. Puerto que como explicado anteriormente, te conecta con la world wide web.  Los tres que vienen a continuación están asociados a apple, al acceso xsan filesystem.  El ultimo no está asignado.  Al igual que pasaba en los gráficos por bytes, por paquetes también vemos que el top destino está mucho mas igualado respecto a porcentajes que el origen. |

UDP:

|  |
| --- |
| **'tshark -r {} -T fields -e udp.srcport -e frame.len'.format(args.tracefile)**  Hemos aplicado la misma lógica que en IP origen por byte pero cambiando ip.dst por upd.srcport |
| En esta gráfica vemos que el puerto más utilizado, con diferencia, como salida es el 24704, puerto no [asignado](https://es.adminsub.net/tcp-udp-port-finder/24704) en el protocolo UDP con un 91’3%.  El siguiente es el 53, el cual es utilizado por [la IANA o Facetime](https://es.adminsub.net/tcp-udp-port-finder/53), una aplicación de [Apple](https://support.apple.com/es-es/HT202944), utilizando un 5’6% del total.  El tercer puerto, al igual que el 53, es uno utilizado por las mismas empresas y el cual sirve (en apple) para detectar la impresora entre otros. El cuarto, el 546, está asociado a la [IANA](https://es.adminsub.net/tcp-udp-port-finder/546), al igual que el 1900, que también está asociado a [IANA](https://es.adminsub.net/tcp-udp-port-finder/1900) en el SSDP, el protocolo simple de descubrimiento de servicios. |
| **'tshark -r {} -T fields -e udp.dstport -e frame.len'.format(args.tracefile)**  Hemos aplicado la misma lógica que en UDP origen por byte pero cambiando udp.srcport por udp.dstport |
| En este gráfico podemos ver que el puerto más utilizado es el 21041. Al igual que en el grafico de arriba, es un puerto sin asignar y que mas tarde IANA se lo “da” a alguna entidad solicitante con un 93’3%.  Los últimos cuatro siguientes puertos, son los mismos que en la grafica de top UDP origen por bytes. |
| **'tshark -r {} -T fields -e udp.srcport'.format(args.tracefile)**  **uniq -c salida\_udp\_src\_paquetes.dat**  Al igual que las anteriores veces, hemos aplicado el mismo algoritmo que en top IP origen por paquetes pero en vez de ip.src hemos puesto upd.srcport. |
| Como podemos observar, en esta grafica predomina el puerto 24704, al igual que en el top puertos origen por bytes. La diferencia es que, en esta, el uso de este puerto es de un 99’6% frente a los otros cuatro siguientes.  El segundo es el 1900, de Microsoft; el tercero el 49169, que es un puerto dinámico y/o privado, al igual que el cuarto y quinto. |
| **'tshark -r {} -T fields -e udp.dstport'.format(args.tracefile)**  **uniq -c salida\_udp\_dst\_paquetes.dat**  Para finalizar, hemos hecho lo mismo que en top UPD origen paquetes pero cambiando la variable udp.srcport pot udp.dstport |
|  |
| En esta última gráfica vemos que el puerto más utilizado es el 21041. Es un puerto sin asignar y que más tarde IANA se lo “da” a alguna entidad solicitante con un 93’3%.  El segundo es el 1900, de Microsoft; el tercero 12013 pertenece a [vipera ssl](https://es.adminsub.net/tcp-udp-port-finder/12013); el cuarto y el quinto están sin asignar. |
|  |

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

|  |
| --- |
| Dirección MAC está usada para filtrar las series temporales y separar los sentidos: 00:11:88:CC:33:3D |
| Inserta aquí la serie temporal del ancho de banda cuando la MAC usada es el origen de la comunicación  Indica aquí el filtro o filtros utilizados en tshark para obtener los datos |
| Comenta y explica la gráfica anterior |
| Inserta aquí la serie temporal del ancho de banda cuando la MAC usada es el destino de la comunicación  Indica aquí el filtro o filtros utilizados en tshark para obtener los datos |
| Comenta y explica la gráfica anterior |

1. ECDFs de los tamaños de los paquetes

|  |
| --- |
| Inserta aquí la gráfica del ECDF del tamaño de todos los paquetes (a nivel 2)  Indica aquí el filtro o filtros utilizados en tshark para obtener los datos |
| Comenta y explica la gráfica anterior |

1. ECDF tiempos entre paquetes

|  |
| --- |
| Inserta aquí la gráfica del ECDF del tiempo entre paquetes del flujo TCP  Indica aquí el filtro o filtros utilizados en tshark para obtener los datos |
| Los puertos TCP son protocolos orientados a la conexión confiable con recepción en orden y control de flujo y congestión. Es por esto por lo que cuenta con una **velocidad de transmisión** más lenta que el UDP, ya este no verifica la correcta transmisión de datos entre dispositivos.  Si comparamos el grafico del ECDF del flujo TCP con el de UDP podemos ver que se confirma, ya que el tiempo de transmisión entre paquetes de esta gráfica es menor al UDP. |
| Inserta aquí la gráfica del ECDF del tiempo entre paquetes del flujo UDP  Indica aquí el filtro o filtros utilizados en tshark para obtener los datos |
| En comparación con la gráfica ECDF del tiempo entre paquetes del flujo TCP, el tiempo es mayor al TCP por lo explicado anteriormente.  Este tiempo que gana, es a costa de una perdida de precisión en la transmisión. |

# Conclusiones

La memoria debe contener unas conclusiones útiles y razonadas acerca del trabajo realizado. Las conclusiones NO deben ser simples resúmenes de la memoria

En esta práctica hemos aprendido cómo funciona la monitorización de una red consistente en el análisis de rendimiento y estado, más concretamente, como funciona la monitorización pasiva.

También hemos obtenido para cada

Esta práctica nos ha parecido muy interesante. Además de esto, hemos asentado muchos conocimientos que antes de esta práctica no teníamos muy asentados.

Lo que más nos ha gustado ha sido ver donde estaban ubicadas cada IP de nuestras gráficas, aunque, sobre todo, el ver los puertos TCP y UDP y a qué estaban asignados cada uno.