1. **Introducción**

En esta práctica hemos realizado una monitorización de red. Para ello se nos ha proporcionado un archivo *.pcap* que hemos tenido que analizar utilizando la herramienta *tshark*. Para simplificar la ejecución de la práctica, hemos ido introduciendo todos los comandos en una serie de ejecutables *.sh* que recogen todos los requisitos del enunciado.

El programa generador de trazas nos ha otorgado los siguientes valores:

* IP: 119.25.90.131
* MAC: 00:11:88:CC:33:E5
* Puerto UDP: 27884

1. **Requisitos**

A continuación, vamos a analizar los puntos requeridos en el enunciado:

**2.1 Porcentajes**

Para estudiar los porcentajes de paquetes IP hemos utilizado los siguientes filtros de tshark:

***tshark -r traza.pcap -T fields -e frame.len*** (así obtenemos todos los paquetes y los podemos contar con un *wc* para poder obtener los porcentajes)

***tshark -r traza.pcap -T fields -e ip.dst -e ip.src -e udp.dstport -e udp.srcport -e tcp.dstport -e tcp.srcport -e frame.len -Y ‘eth.type eq 0x00000800 or vlan.etype eq 0x00000800’*** (así obtenemos TODOS los paquetes IP, TCP y UDP y luego analizamos el resultado obtenido mediante *awk*, obteniendo los datos necesarios)

El resultado obtenido es el siguiente:

(imagen)

Como se puede apreciar en la imagen, el 99.01% de los paquetes son IP, lo que indica que la inmensa mayoría del tráfico de red capturado utiliza dicho protocolo en sus comunicaciones.

Por otro lado, el 89.59% del tráfico IP capturado emplea el protocolo de transporte IP-TCP, que garantiza la entrega sin errores de los datos, en el mismo orden que se transmitieron. Claramente, es el protocolo de transporte IP más utilizado (posiblemente gracias a las características descritas), frente al 9.02% del UDP y el 1.29% de otros protocolos.

Podemos sacar ciertas conclusiones acerca de este último punto, como descartar la posibilidad de estar realizando o recibiendo un *streaming,* debido al escaso tráfico UDP. El protocolo TCP estándar se emplea, por ejemplo, en páginas web comunes (es decir, protocolo HTTP), por lo que podemos suponer que los usuarios de la red se dedican a este tipo de navegación.

**2.2 TOPS 10 POR PAQUETES**

Para obtener los tops 10 que nos solicitan, hemos utilizado el resultado obtenido de la consulta *tshark* anterior y, empleando las herramientas *awk* (para quedarnos con las columnas que nos interesan y darles formato)*, sort* (para ordenar los resultados)*, sed* (para borrar líneas vacías)*, uniq* (para agrupar los datos iguales y contabilizar su cantidad) *y head* (para imprimir los 10 primeros)*.*

El resultado obtenido es el siguiente:

(imagen ip destino)

Podemos ver que la mayoría de los paquetes se dirigen a la IP 29.181.234.168. Nuestra IP (119.25.90.131) se encuentra en la sexta posición

(imagen ip origen)

En este caso, la IP que genera el mayor número de paquetes es la 84.98.180.17. Nuestra IP se encuentra en la novena posición.

(imagen tcp destino)

Se puede apreciar que el puerto que recibe el mayor tráfico es el puerto 80, asociado al protocolo HTTP. Esto nos confirma la observación de que los usuarios se dedican a la navegación en páginas web con este protocolo.

(imagen tcp origen)

En el caso de los puertos de origen, de nuevo destaca ampliamente el puerto 80, reforzando aún más nuestra hipótesis.

(imagen udp destino)

La mayoría de paquetes UDP se dirigen al puerto 27884, el cual no hemos podido relacionar con ningún servicio conocido. Sin embargo, en segunda posición se encuentra el puerto 53, asociado al Sistema de Nombres de Dominio (DNS), encargado (entre otras cosas) de “traducir” las URL’s de las páginas web a IP’s con las que poder establecer conexión. De nuevo, estamos reforzando la hipótesis de una navegación estándar en la web.

(imagen udp origen)

En el caso de los puertos de origen, podemos ver que el más recurrido es el 48883, mientras que nuestro puerto referencia (27884) no aparece en la lista. Más adelante veremos que no se generan paquetes con origen en este puerto.

**2.3 TOPS 10 POR BYTES**

(imagen ip destino)

Pese a que la primera posición se mantiene, nuestra dirección IP se ve relegada a la octava posición. Esto nos hace pensar que los paquetes que llegan a nuestra IP son de pequeño tamaño. El tamaño total de los paquetes enviados a la IP que encabeza la lista es casi 20 veces mayor que el segundo. Esta relación aumenta frente al número de paquetes, que equivale a casi 10 veces el de la segunda posición. Por ello es fácil concluir que los paquetes enviados a la IP 29.181.234.168 son de gran tamaño en general.

(imagen ip origen)

En este caso, la primera posición se mantiene invariante mientras que nuestra IP ha avanzado dos posiciones. Podemos concluir que los paquetes UDP enviados desde nuestra IP son, relativamente, de mayor tamaño que los enviados desde otras direcciones.

(imagen tcp destino)

Se puede apreciar que el puerto 80 no está en el top 10 por número de bytes. Esto parece indicar que los paquetes destinados a este puerto, pese a ser muchos, son de escaso tamaño. Posiblemente se trate de peticiones a páginas web.

(imagen tcp origen)

En esta imagen volvemos a tener de nuevo al puerto 80 como el más recurrente. El hecho de que los paquetes originados en este puerto sean también los que acumulen el mayor número de bytes puede hacernos pensar que haya algún servidor transmitiendo contenido web HTTP (por ejemplo, algún servicio Apache).

(imagen udp destino)

Al igual que en la ordenación por paquetes, la primera posición la ocupa nuestro puerto: el 27884. De nuevo, el puerto 53, asociado al DNS, se encuentra en la segunda posición.

(imagen udp origen)

En el caso de los puertos de origen, podemos ver que el más recurrido es el 48883, mientras que nuestro puerto referencia (27884) no aparece en la lista. Más adelante veremos que no se generan paquetes con origen en este puerto.

**2.4 ECDF’S**

En este apartado analizamos las diferentes gráficas ECDF que se nos solicitaba. Para ello hemos realizado diversas consultas *tshark*, para posteriormente analizarlas mediante *awk* y obtener las frecuencias acumuladas para las ECDF en los script *hacer\_ECDF\_...sh*

***tshark -r traza.pcap -T fields -e frame.len -e eth.src -e eth.dst -Y "eth.addr=00:11:88:CC:33:E5"***

Con esta consulta obtenemos el tamaño de los paquetes que utilizan nuestra dirección MAC como origen o destino, y separamos ambos casos usando *awk*. El resultado es el siguiente:

(ecdfs nivel 2)

En el caso de los paquetes entrantes (dirigidos a nuestra MAC), podemos ver que la inmensa mayoría tiene un tamaño alrededor de los 1500 Bytes. Hay menos de un 10% con un tamaño inferior. Sin embargo, viendo la gráfica de los paquetes originados en nuestra dirección, vemos que cerca del 80% tienen un tamaño inferior a 200 Bytes y que, además, hay un considerable número de paquetes de tamaño cercano a los 500 Bytes.

***tshark -r traza.pcap -T fields -e frame.len -e tcp.srcport -e tcp.dstport***

Con esta consulta obtenemos el tamaño de los paquetes TCP y vamos a quedarnos sólo con los HTTP (aquellos que empleen el puerto 80) mediante comandos *awk*.

***awk -v port=80 '$2 == port {print $1}'*** (con origen en el puerto 80)

***awk -v port=80 '$3 == port {print $1}'*** (con destino el puerto 80)

(ecdfs HTTP)

Podemos ver que más del 90% de los paquetes HTTP dirigidos a nuestra dirección tienen un tamaño inferior a 200 Bytes, mientras que los originados en dicha dirección son, en su mayoría, de un tamaño cercano a los 1500 Bytes. De nuevo, esto nos hace pensar que tratamos con paquetes originados en algún tipo de servidor que recibe muchos paquetes de pequeño tamaño (peticiones) y sirve paquetes de gran tamaño, que serán el contenido del web.

***tshark -r traza.pcap-T fields -e frame.len -e udp.srcport -e udp.dstport***

Con esta consulta obtenemos el tamaño de los paquetes UDP y vamos a quedarnos sólo con los DNS (aquellos que empleen el puerto 53) filtrando mediante *awk.*

***awk -v port=53 '$2 == port {print $1}'*** (con origen en el puerto 53)

***awk -v port=53 '$3 == port {print $1}'*** (con destino el puerto 53)

(ecdfs DNS)

En el caso de los paquetes DNS entrantes, observamos una distribución logarítmica de los datos, de forma que la mayoría de paquetes tienen un tamaño de entre 70 y 100 Bytes. Si nos fijamos en los paquetes salientes, la distribución es ligeramente similar, aunque menos estable, y en este caso los tamaños se reparten entre los 70 Bytes y los 300 Bytes. Viendo que los paquetes que salen son de mayor tamaño que los de entran, se puede suponer que se están realizando *querys* y *responses* DNS, ya que los paquetes de pequeño tamaño se pueden asociar a peticiones DNS, mientras que los de mayor tamaño pueden asociarse a respuestas que incluyan la dirección asociada al nombre solicitad0.

***tshark -r traza.pcap-T fields -e frame.time\_relative -e ip.src -e ip.dst -Y 'tcp'***

Con esta consulta obtenemos los tiempos absolutos de todos los paquetes TCP. Mediante comandos *awk* vamos a filtrar aquellos que tienen nuestra IP como origen/destino y luego vamos a calcular el tiempo entre paquetes.

***awk 'BEGIN{antigua = $1} {delta = $1-antigua; printf("%.7f\n", delta); antigua = $1 }'*** (ejemplo de comando *awk* para calcular el tiempo relativo entre paquetes)

(graficas de tiempo TCP)

Se puede ver que, en el caso de los paquetes entrantes, hay ciertas oscilaciones hasta los 0.0004 nanosegundos donde ya se estabiliza, lo que quiere decir que más del 80% de los paquetes tienen una diferencia de tiempo menor a 0.0004. Por motivos visuales, nuestra gráfica no alcanza la frecuencia total, sino que se queda en 0.9. Esto se debe a que, si aumentamos el rango de tiempos, no podemos apreciar correctamente las oscilaciones que tienen lugar antes de los 0.004 nanosegundos.

En el caso de los paquetes salientes, tenemos unas oscilaciones muy similares, que llegan hasta poco más de los 0.0002 nanosegundos. En este caso, algo más del 90% de los paquetes tienen una diferencia de tiempo menor que 0.0003 nanosegundos.

***tshark -r traza.pcap-T fields -e frame.time\_relative -e udp.srcport -e udp.dstport -Y 'udp.port == 27884'***

Con esta consulta obtenemos los tiempos absolutos de los paquetes UDP que utilizan el nuestro puerto. De la misma forma que en el caso anterior, usaremos *awk* para separar entre tráfico entrante y saliente y calcular el tiempo relativo entre paquetes, obteniendo los siguientes resultados:

(graficas tiempo UDP)

Se puede ver que el tiempo entre los paquetes entrantes es muy similar, del orden de 0.02 nanosegundos, en todos los casos. Sin embargo, no hay paquetes UDP cuyo origen sea nuestro puerto.