Práctica 3: Monitorización

Alejandro Sánchez Sanz (alejandro.sanchezsanz@estudiante.uam.es)
Ricardo Riol González (ricardo.riol@estudiante.uam.es)

Redes de Comunicaciones I
Universidad Autónoma

16 de noviembre de 2017

Introducción

El objetivo principal de esta práctica es ponerse en la piel de un gestor de red y valorar el comportamiento de la red mediante monitorización pasiva, es decir, analizando el tráfico que atraviesa la red midiendo ciertas magnitudes sin modificar el estado de la misma. Para lograr nuestro objetivo hemos tenido que aprender a manejarnos con *awk*, *shell scripting* y *tshark*.
La herramienta *awk* nos ha facilitado muchísimo el manejo de los ficheros generados con *tshark*, permitiéndonos calcular probabilidades, sumas, tamaños totales, arrays asociativos, etc. de un fichero de más de 50.000 líneas.
Por otro lado, la técnica de *shell scripting* nos ha ayudado a englobar todas las pruebas en un único script, por lo que para realizar el análisis de una red bastaría con conectarse a ella y ejecutar dicho script. El *shell scripting* también nos ha facilitado el trabajo de realizar gráficas con GNUplot (herramienta que permite hacer gráficas a partir de ficheros independientemente de su tamaño. Esta herramienta es fundamental para observar los resultados del análisis de una forma más intuitiva).
Por último *tshark* es una herramienta muy fuerte que no solo nos ha permitido extraer datos de una captura (fichero .pcap) a un fichero de texto, sino que también nos ha permitido aislar y filtrar datos que necesitábamos para realizar ciertas pruebas.
En definitiva, el análisis se resume a una combinación de las técnicas anteriormente mencionadas y comandos de bash, que tienen como resultado un script que genera diferentes métricas pasivas de la red y que nos permite intuir cuál es el estado de la misma.

Generación del archivo PCAP

La práctica requiere que cada pareja utilice una traza pcap diferente. Para ello con el enunciado se nos da un generador de trazas. El comando que pusimos en bash fue el siguente:
./generadorPCAPx32 1302 09 traza_1302_09.pcap.

Obtuvimos los siguientes resultados:

La dirección MAC que deberá tener en cuenta es: 00:11:88:CC:33:1
La dirección IP (origen o destino) del flujo TCP que deberá tener en cuenta es: 81.84.126.202
El puerto (origen o destino) del flujo UDP que deberá tener en cuenta es: 54771

Resultados

A continuación se explicarán los resultados obtenidos en cada de una de las pruebas realizadas. Analizaremos las pruebas individualmente para acabar con un enfoque global de todo el conjunto de las pruebas que determinará el estado de la red.

Porcentajes de paquetes

Porcentaje de paquetes IP y NO-IP

Para determinar qué porcentaje de la traza generada tiene como protocolo de nivel 3 IP (hemos considerado que se nos pedía IPv4, pues es el que hemos estudiado y porque de lo contrario, toda la traza sería IP, ya que los restantes son IPv6), no solo se consideraron los paquetes cuyo campo eth.type (nivel enlace) es 0x0800, sino que también contamos con los paquetes que tienen por debajo del nivel de enlace el protocolo VLAN (0x8100) y debajo de este IP (el campo vlan.etype es 0x0800).
Para lograr obtener el porcentaje de paquetes IP de la traza, implementamos **ejercicio1.awk**. Este awk se encarga de ir leyendo línea a línea el fichero **tipos.txt**, que contiene los campos eth.type, ip.proto (después se explicará la razón) y vlan.etype (extraídos con tshark), e irá contando cuántos paquetes tienen en su campo eth.type o vlan.etype (dependiendo de si IP se monta sobre VLAN o no) el valor 0x0800. Finalmente, imprime el porcentaje.
El total de paquetes con protocolo IP es 99.0066 %, de los cuales el 92.2072 % tiene directamente IP después del nivel de enlace y el 6.79936 % tiene el protocolo IP sobre VLAN.
Como se observa, la gran mayoría de paquetes tienen protocolo IP, lo que confirma que este es el protocolo de nivel 3 dominante en la red.

Porcentaje de paquetes UDP, TCP y otros

En el segundo apartado se nos pedía que encontrásemos los porcentajes de los distintos protocolos de nivel de transporte de los paquetes IP del apartado anterior. Para ello añadimos al fichero **tipos.txt** la columna ip.proto y calculamos porcentajes igual que en apartado anterior. Los resultados son los siguientes:

1. UDP 9.1404 %
2. TCP 89.5925 %
3. ICMP 1.39347 %

(En los resultados obtuvimos una fila con dos protocolos 1 y 17. Ese pqueño porcentaje se lo sumamos al primer protocolo que aparecía, el de ip.proto igual a 1, es decir ICMP. Esto lo hicimos debido a que este protocolo transporta el mensaje de error de otro paquete que fue descartado, y el segundo protocolo es el de dicho paquete).
Si observamos los datos domina el protocolo TCP al igual que antes el protocolo IP (sin VLAN), lo que nos da a entender que la mayoría de los paquetes de la traza cumplirán la pila de protocolos **ETH\IP\TCP**.

TOPS 10

En este apartado se pide calcular los top 10 tanto en bytes como en número de paquetes de direcciones IP y puertos. En bytes se refiere qué direcciones y puertos tienen asociados más bytes (suma de la longitud de los paquetes). Por otro lado, por paquetes significa que encontremos las direcciones IP y puertos que más se repiten en la traza. El formato es: #paquetes/bytes dirección/puerto

IP origen

```
TOP 10: IP Origen por paquetes
  15454 111.68.162.197
  11463 56.138.8.177
   5805 102.144.108.230
   4657 56.91.199.177
   2906 38.31.150.209
   2188 106.145.81.104
   2161 74.184.73.23
   2048 14.133.151.145
   1883 81.84.126.202
   1652 66.162.99.108
```

```
TOP 10: IP Origen por bytes
  23098523 111.68.162.197
   6918040 56.91.199.177
   4344112 38.31.150.209
   3245100 106.145.81.104
   3193577 74.184.73.23
   3009353 14.133.151.145
   2730262 81.84.126.202
   2473818 66.162.99.108
   1970587 102.144.108.230
   1025537 56.138.8.177
```

IP destino

```
TOP 10: IP Destino por paquetes
  34986 56.138.8.177
   3881 111.68.162.197
   3785 5.221.41.131
   2857 102.144.108.230
   1273 56.91.199.177
   1046 81.84.126.202
    983 38.31.150.209
    666 74.184.73.23
    664 105.24.234.78
    619 66.162.99.108
```

```
TOP 10: IP Destino por bytes
  50345203 56.138.8.177
   2853122 102.144.108.230
   1816645 5.221.41.131
    249160 111.68.162.197
    115206 105.24.234.78
     79229 56.91.199.177
     76301 17.121.109.203
     70017 81.84.126.202
     59576 38.31.150.209
     47886 74.184.73.23
```

TCP origen

```
TOP 10: Puerto origen TCP por paquetes
  36640 80
   1423 55934
   1096 55860
   1046 54615
    617 55865
    607 43585
    603 33896
    471 55173
    418 55848
    380 33903
```

```
TOP 10: Puerto origen TCP por bytes
  52857665 80
    217800 443
     88065 55934
     70017 54615
     67367 55860
     40574 55865
     36512 43585
     35533 33896
     28338 55173
     26382 46832
```

TCP destino

```
TOP 10: Puerto destino TCP por paquetes
  12342 80
   5486 55934
   4313 55860
   3204 55865
   2188 43585
   1883 54615
   1813 33896
   1717 55173
   1396 55848
   1174 46371
```

```
TOP 10: Puerto destino TCP por bytes
   8236507 55934
   6437994 55860
   4808618 55865
   3245100 43585
   2730262 54615
   2707440 33896
   2566453 55173
   2072650 55848
   1756652 46371
   1690967 57063
```

UDP origen

```
TOP 10: Puerto origen UDP por paquetes
   3785 48883
    592 53
    124 546
     95 5353
     12 1900
      6 63423
      6 58532
      6 55421
      6 49169
      3 61153
```

```
TOP 10: Puerto origen UDP por bytes
   1816645 48883
     85720 53
     23317 5353
     18337 546
      6447 1900
      1080 63423
      1080 58532
      1080 55421
      1080 49169
       624 61153
```

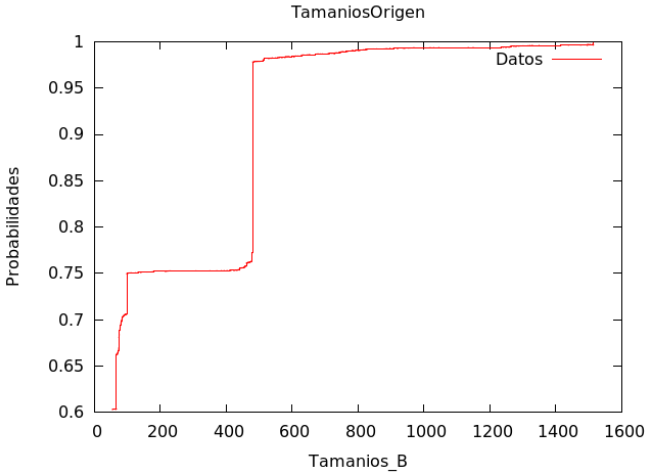
UDP destino

```
TOP 10: Puerto destino UDP por paquetes
   3785 54771
    591 53
    134 5355
    124 547
     95 5353
     42 1900
      2 5035
      2 12013
      1 9920
      1 9800
```

```
TOP 10: Puerto destino UDP por bytes
   1816645 54771
     46391 53
     23317 5353
     18337 547
     12015 1900
     11460 5355
       533 12013
       461 5035
       394 64925
       318 23710
```

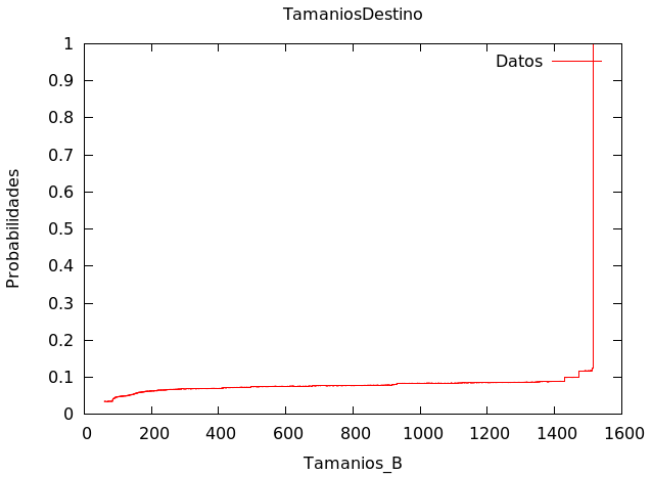
Tamaño de los paquetes nivel 2

Origen (MAC = 00:11:88:CC:33:1)



Observando la gráfica y los datos podemos apreciar que los tamaños que más se repiten son 54 y 480. El primero es el más pequeño de todos y se aprecia su alta frecuencia (es el más repetido) al acumular una probabilidad de 0.6; y el segundo provoca una pronunciada subida en la gráfica al acumular una probabilidad de aproximadamente 0.2. También provocan subidas menores los valores 66 y 98. Todos estos no son tamaños muy grandes y eso se debe a que los paquetes que enviamos nosotros (y por tanto, tienen nuestra MAC como origen) suelen ser peticiones a servidores o dominios, y su bajo tamaño viene de la búsqueda de reducir el número de colisiones. Por otro lado, el resto de tamaños aparecen pocas veces y el máximo es 1514 (que coincide con el tamaño máximo de un paquete Ethernet).

Destino (MAC = 00:11:88:CC:33:1)

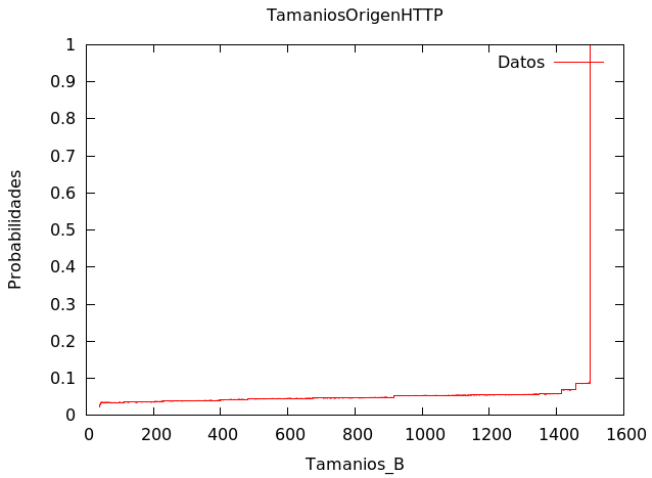


Esta ECDF nos muestra con claridad que aproximadamente el 90 % de los paquetes son de tamaño 1514. Esto se debe a que un servidor, tras aceptar nuestra petición, nos envía grandes cantidades de datos, como puede ser el caso de una descarga (de GitHub por ejemplo) o la visualización de un vídeo (en YouTube por ejemplo).

Tamaño a nivel 3 de los paquetes HTTP

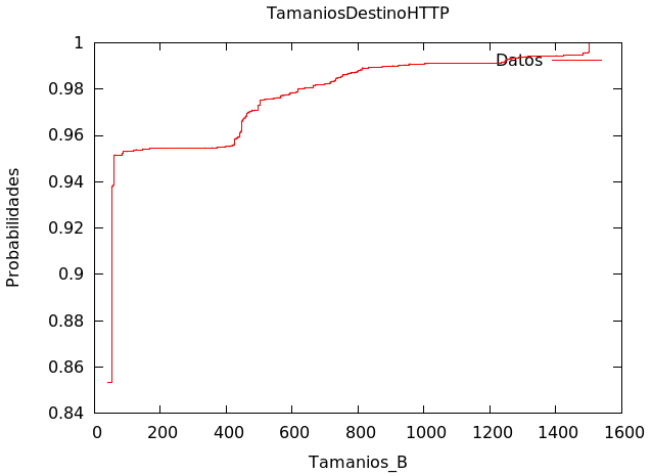
Se entiende HTTP como aquellos paquetes que sobre IP (nivel 3) tienen TCP y usen el puerto 80 como origen o destino. Ahora, está muy bien saber qué es HTTP, pero ¿para qué se utiliza?.
El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL.

Origen



En esta gráfica están representados los tamaños de los paquetes que siguen la pila de protocolos IP\TCP (de nivel 3 y 4) frente a la probabilidad de que un paquete sea de dicho tamaño. Como se puede observar la inmensa mayoría (90 % aproximandamente) de los paquetes tienen tamaño 1500 (tamaño máximo de datagrama). Esto se debe a que estos paquetes, que salen del servidor web con dirección a nuestro navegador (ya que el puerto origen es el 80), contienen los datos que se solicitan al servidor web en la solicitud HTTP. Por otra parte, el resto de paquetes (tamaño menor que 1500) tienen una probablilidad de en torno al 10 %, ya que su frecuencia en estas "conversaciones"servidor web - navegador es muy pequeña.

Destino

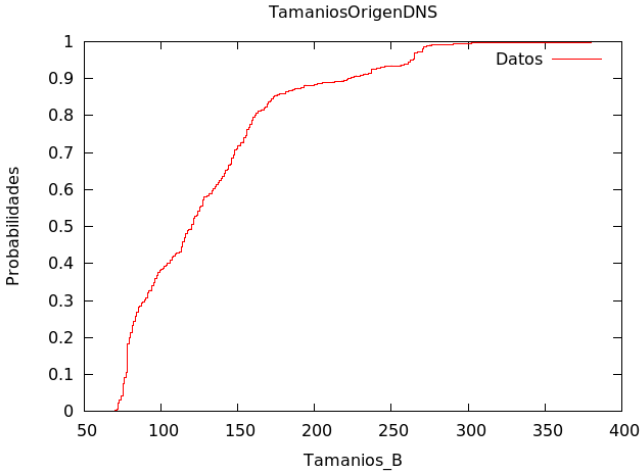


Al igual que en la gráfica anterior se representa el tamaño de los paquetes IP\TCP, pero ahora con puerto destino 80, frente a la probabilidad de que un paquete sea de dicho tamaño. La diferencia es que ahora somos nosotros los que desde nuestro navegador enviamos una solicitud HTTP al servidor web. En esta solicitud requerimos al servidor que nos envíe el recurso ubicado en la URL especificada. Esto se traduce en que los paquetes tengan tamaños pequeños y muy similares. La gráfica muestra esta situación de forma muy clara, la probabilidad de que aparezcan paquetes con tamaño próximo a 50 es del 80 % (muy alta). El valor más repetido en este tipo de conversaciones navegador - servidor es el tamaño 40, que es el que produce ese salto en la probabilidad nada más empezar. Por otra parte, a medida que aumentamos el tamaño, baja la probabilidad, lo que significa una menor frecuencia de este tipo de paquetes.

Tamaño a nivel 3 de los paquetes DNS

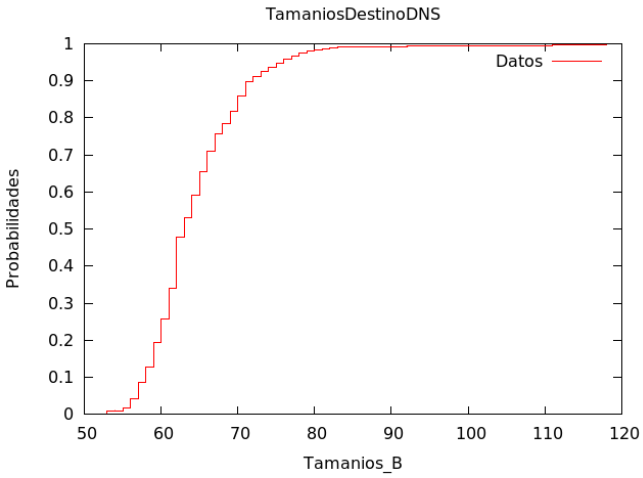
Cuando se quiere acceder a una página web en internet necesitamos la dirección IP del servidor donde está almacenada, pero por lo general el usuario solo conoce el nombre del dominio. El protocolo DNS (Domain Name System) es el que realiza esta traducción entre dominios y direcciones IP. Si desde nuestro navegador queremos acceder a una página web (mediante la URL), y esta no está en la caché que relaciona URLs con direcciones IP, se enviará una petición al servidor DNS (puerto 53) para que se traduzca la URL de la página web solicitada en la dirección IP correspondiente.

Origen



En esta gráfica se muestran los tamaños de los paquetes UDP (nivel 4) con puerto origen 53 frente a la probabilidad de que aparezcan paquetes de dicho tamaño. Como se ha explicado anteriormente, DNS se encarga de la traducción entre dominios y direcciones IP. Concretamente esta gráfica muestra los tamaños de los paquetes que salen del servidor DNS dirección nuestro navegador con la IP requerida. Como se observa no son paquetes muy grandes (ya que el tamaño máximo es 1500) con tamaños comprendidos entre 100 y 200 (la inmensa mayoría), ya que estos paquetes transportan la IP requerida por el navegador.

Destino



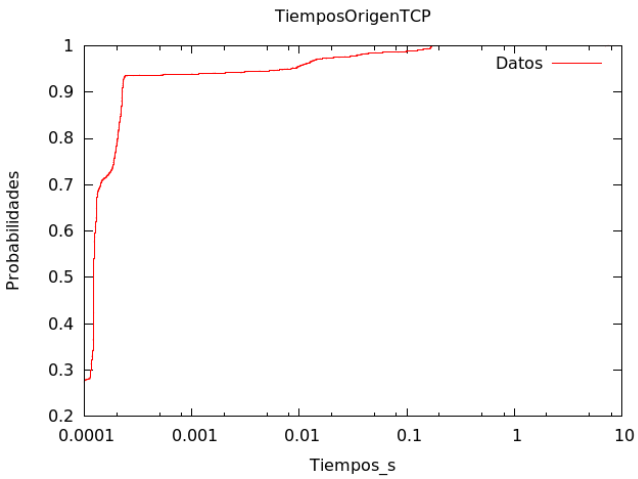
Al igual en gráfica anterior, se muestra el tamaño de los paquetes UDP con puerto destino 53 frente a las probabilidades. Los paquetes enviados en este sentido, navegador - servidor DNS, contienen la URL (que es simplemente una cadena de caracteres) de la cual el cliente quiere la dirección IP. De ahí, que estos paquetes tengan tamaños tan pequeños, comprendidos entre 60 y 80. Esta ECDF es la que más claramente sigue una distribución Normal. Si comparamos ambas gráficas, la forma

es muy similar, sin embargo es sencillo observar que el tamaño medio de los paquetes en el sentido origen es mayor que en el sentido destino. Por lo tanto, podemos concluir que el los paquetes que contienen dirección IP tienen un mayor tamaño de media que los que contienen la URL (cadena de caracteres) de la página web.

Tiempos entre llegadas del flujo TCP (Dirección IP a tener en cuenta: 81.84.126.202)

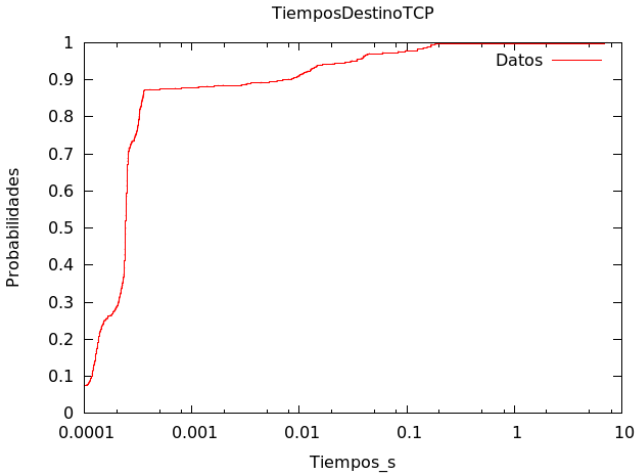
El protocolo TCP está orientado a conexión. Cuando una máquina A envía datos a una máquina B, la máquina B es informada de la llegada de datos, y confirma su buena recepción. Aquí interviene el control CRC de datos que se basa en una ecuación matemática que permite verificar la integridad de los datos transmitidos. De este modo, si los datos recibidos son corruptos, el protocolo TCP permite que los destinatarios soliciten al emisor que vuelvan a enviar los datos corruptos. Esto ocasiona una sobrecarga a cambio de la fiabilidad.

Origen



En esta gráfica hemos representado el tiempo entre llegadas de paquetes en el eje de abscisas con escala logarítmica debido al reducido tamaño de los valores. Podemos observar que la mayoría de los tiempos son muy pequeños, por lo que podría decirse que los paquetes llegan "a la vez". Esto se debe a que TCP es un protocolo que establece una conexión previa entre cliente y servidor, así que después se puede realizar el envío bidireccional sin preocupaciones. De todos modos, existen algunos paquetes que llegan tras tiempos mayores y eso se debe a que hay algún intervalo notable de tiempo en el que no existe tráfico TCP de este tipo.

Destino



A la hora de interpretar esta ECDF (también con escala logarítmica en el eje de abscisas) se pueden apreciar muchas similitudes con la gráfica anterior, debido a que la conexión establecida previamente permite enviar tráfico en los dos sentidos (el sentido influye a la hora de estudiar los tamaños, pero en cuanto a los tiempos es razonable que sea similar). Existe mayoría de paquetes separados por pequeños intervalos de tiempo, pero otros poseen una diferencia mayor, debido al tiempo sin tráfico de este tipo.

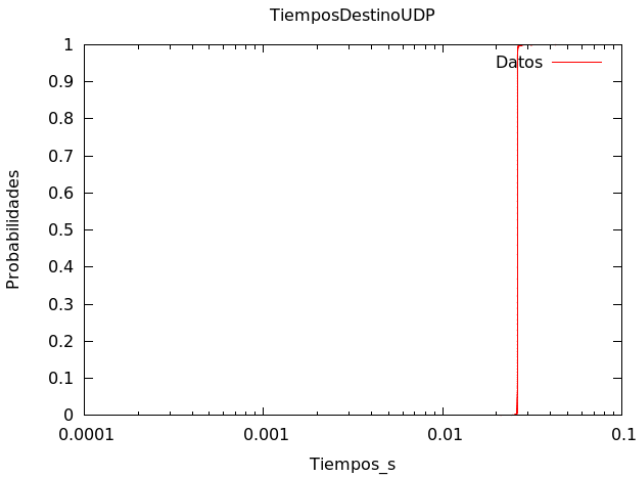
Tiempos entre llegadas del flujo UDP (Puerto a tener en cuenta: 54771)

UDP es un protocolo no orientado a conexión. Es decir, cuando una máquina A envía paquetes a una máquina B, el flujo es unidireccional. La transferencia de datos es realizada sin haber realizado previamente una conexión con la máquina de destino (máquina B), y el destinatario recibirá los datos sin enviar una confirmación al emisor (la máquina A). Esto es debido a que la encapsulación de datos enviada por el protocolo UDP no permite transmitir la información relacionada al emisor. Por ello el destinatario no conocerá al emisor de los datos excepto su IP. La sobrecarga introducida no es muy elevada y aumenta en velocidad pero pierde en fiabilidad.

Origen

En este apartado no existe ECDF debido a que el filtrado mediante tshark no nos proporciona ningún dato. Esto quiere decir que no existe ningún paquete de nuestra traza que posea el puerto 54771 como puerto origen de UDP, es decir, no enviamos paquetes de este tipo. La razón por la que esto ocurre es por el empleo común de este protocolo para medios de *streaming* de audio o vídeo (como podrían ser Spotify o Twitch, respectivamente), los cuales establecen una comunicación unidireccional (no existe por tanto conexión previa para sincronizarse entre cliente y servidor). Es por ello que nosotros como clientes únicamente estamos recibiendo y no enviando estos paquetes.

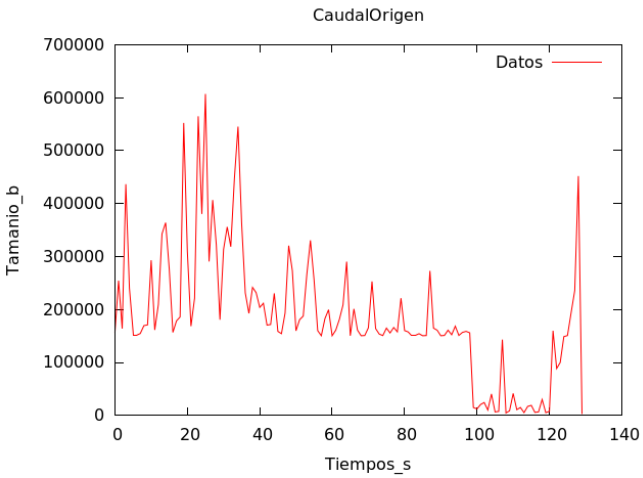
Destino



Aquí, por lo dicho en el apartado previo, observamos todo el tráfico UDP hacia nuestro puerto (el 54771), el cual poseerá toda la información que se nos difunde mediante este servicio. Se aprecia que aparece toda condensada aproximadamente en el instante 0.025 segundos. El hecho de que la inmensa mayoría de los paquetes lleguen con un retardo similar es una de las características propias de UDP, y es que al no intentar establecer una conexión ni reenviar paquetes perdidos, no introduce ningún retardo adicional que ocasione *jitter*. La ECDF presenta el tiempo en el eje de abscisas con una escala logarítmica debido al reducido tamaño de los valores.

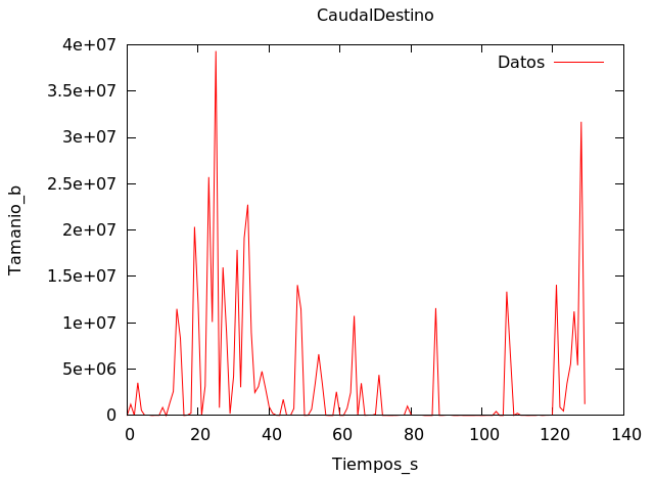
Ancho de banda a nivel 2 (b/s)

Origen



La serie temporal que obtenemos del ancho de banda de subida presenta una distribución bastante "picuda.ª simple vista, lo cual indica que el tráfico que generamos no es de tamaño constante, pero sí es cierto que observamos que en su mayoría, presenta una cota inferior de unos 150000 bits. La excepción ocurre en el intervalo final de tiempo a partir de los aproximadamente 100 segundos, en el cual el ancho de banda se reduce hasta valores más próximos al 0 (en el intervalo entre 100 y 120 segundos es donde mejor se aprecia esto). Sin embargo, se observan algunos valores atípicos en esta parte final de la gráfica alrededor del segundo 107 y del 129, donde el caudal aumenta considerablemente en comparación con su entorno. Por último, se puede mencionar que en general el tráfico no es muy elevado pues los valores máximos son de unos 600000 bits.

Destino



El comportamiento del ancho de banda de bajada es todavía más picudo que el anterior, y observamos que presenta tamaños mucho mayores (los de subida eran del orden de las centenas de millar y estos rondan las decenas de millón). Esto se debe a que el servidor nos envía enormes cantidades de datos en contraposición a los mensajes de solicitud, que son más pequeños. De todos modos, hay varias zonas donde estos paquetes sí aparecen con más frecuencia, en las que la gráfica baja notablemente y se acerca al 0. Por último, debemos destacar como valores atípicos los de alrededor del segundo 25 y del segundo 129, donde el caudal alcanza casi los 40 y los 33 millones de bits, respectivamente, pudiendo tratarse de un backup o una descarga.

Conclusiones

La realización de esta práctica en la que actuábamos como unos gestores de red, nos ha servido para familiarizarnos con diversas herramientas como son *tshark* y *awk*, y avanzar en el uso de otras como *shell scripting* y *GNUplot*.

Los porcentajes nos hacen ver que los protocolos dominantes en la red son IPv4 (nivel 3) y TCP (nivel 4).
Por otro lado, los tops nos muestran datos importantes como el hecho de que en UDP el número de paquetes en un sentido y en otro coincide en muchos de los valores, y eso se debe a casos donde la comunicación es unidireccional, como puede ser el *streaming* que mencionábamos con anterioridad, en el que el servidor con puerto UDP origen 48883 nos envía una gran cantidad de información al puerto UDP 54771 (en concreto, 3785 paquetes con un tamaño total de 1816645 bytes).
En cuanto al análisis de los datos obtenidos, se puede concluir que estos son razonables.
Las ECDF presentan distribuciones similares a la de la Normal en su mayoría. Además, encontramos escenarios posibles para cada caso, como han sido algún streaming (cuando solo hay tráfico de tipo UDP y en un único sentido) o algún acceso a internet (cuando existen secuencias de HTTP y DNS); y en todos ellos somos nosotros el cliente.
A la hora de analizar las figuras del ancho de banda observamos una similitud durante el intervalo 100 - 120 segundos debida a una conversación bidireccional (por ejemplo, una llamada de Skype).

En conclusión, los resultados son razonables y nos indican que la traza generada es la de nosotros como clientes accediendo a datos y conversando con y mediante distintos servidores.