Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Práctica 7 – Programando en Java (3)

1- Objetivos

El objetivo de esta práctica es utilizar la programación en Java para analizar los resultados obtenidos con wireshark/tshark y ser capaz de realizar medidas del throughput a partir de una captura, programando su propia herramienta que obtenga información sobre una traza de tráfico de red.

2- Capturando tráfico Ethernet

Aprenda a utilizar el comando tshark para obtener un fichero de texto con los tiempos y tamaños de las tramas Ethernet que se hayan visto en la red de su ordenador. Para ello...

Pruebe el comando tshark, recuerde que le permite observar las tramas ethernet que se ven en la red de area local. Por ejemplo haciendo

```
$ tshark -i eth0
```

Verá los paquetes que se transmiten en la Ethernet. El formato con el que se imprimen en pantalla sin embargo es un poco difícil de leer para un programa. Queremos usar el comando tshark para que le proporcione una lista de las tramas ethernet que se han visto en la red pero de una forma mas fácil de procesar. Para eso la opción -T nos permite decir que solo queremos que imprima algunos campos, la opción -E nos deja elegir opciones por ejemplo que los campos estén separados por espacio. Finalmente con la opción -e vamos eligiendo los campos que queremos que aparezcan. Por ejemplo imprimiendo solo de cada paquete el tiempo en el que se envió y el tamaño del paquete. Separados por espacios con el siguiente comando.

```
$ tshark -i eth0 -T fields -E separator=/s -e frame.time_relative -e frame.len
Capturing on eth0
0.000000000 74
0.000146000 74
0.000174000 66
...
```

O bien si queremos obtener también las direcciones ethernet origen y destino

```
$ tshark -i eth0 -T fields -E separator=/s -e frame.time_relative -e frame.len
-e eth.src -e eth.dst
Capturing on eth0
0.000000000 74 00:24:8c:b7:87:7a 00:22:19:c0:74:75
0.000146000 74 00:22:19:c0:74:75 00:24:8c:b7:87:7a
0.000173000 66 00:24:8c:b7:87:7a 00:22:19:c0:74:75
...
```

Si quiere almacenar los resultados para utilizarlos en la siguiente fase puede dirigir la salida a un fichero.

```
$ tshark -i eth0 -T fields -E separator=/s -e frame.time_relative -e frame.len
>captura.txt
```

Para detener la captura use Control-C o bien mire como especificar un numero máximo de paquetes capturados o tiempo máximo de captura

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

3- Analizando el throughput

Se trata de construir programas que analicen las capturas obtenidas en el apartado anterior. Realice un programa que lea un fichero con la lista de los paquetes y muestre la cantidad total de bytes que ve en cada intervalo de tiempo de duración indicada:

```
Uso:
    java TimeThr <nombredelfichero> <duración del intervalo>

Lee un fichero con el nombre indicado con líneas que contienen la
información de una trama Ethernet observada en la red por orden de
aparición. Cada línea tendrá el formato
<tiempo> <tamaño de la trama Ethernet>
El programa agrupará las tramas que aparezcan en cada intervalo de
duración indicada mostrando para cada intervalo que pase: el tiempo final,
la cantidad de bytes observada en el intervalo, la cantidad de bytes
totales observada hasta ese momento y el throughput (en bps)

Ejemplos:
$ cat fichero1
0.000000000 60
0.006421000 60
0.010704000 124
0.187527000 60
0.201154000 60
0.261155000 60
0.261158000 60
0.353060000 149
0.392433000 221
0.392436000 221
0.421082000 60
0.446327000 60
0.479638000 60
0.492356000 60
0.588422000 92
...
$ java TimeThr fichero1 0.1
0.100 244 244 2439.9
0.200 60 304 599.9
0.300 180 484 1799.9
0.400 591 1075 5909.9
0.500 240 1315 2399.9
...
```

Asegúrese de que el programa funciona correctamente. Para ello compruebe el funcionamiento comparando los resultados obtenidos con su programa con los resultados que obtiene wireshark analizando el mismo fichero de captura. Recuerde que puede obtener los parámetros de tiempos y tamaños a partir de un fichero de captura con tshark o wireshark utilizando la opción –r
Por ejemplo, para capturar 1000 paquetes de la tarjeta Ethernet eth0 puede hacer

```
$ tshark –w fichero.cap –c 1000
$ ls
fichero.cap
```

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Para lanzar wireshark y analizar esa captura sólo debe hacer

```
$ wireshark –r fichero.cap
```

Para lanzar tshark y extraer los campos que le interesen de la misma captura

```
$ tshark –r fichero.cap –T fields –e frame.time_relative >datos
```

Para dibujar los ficheros que esta haciendo utilice la utilidad gnuplot. Abra un terminal y vaya al directorio donde esta el fichero de resultados. Utilice los siguientes comandos

```
$ java TimeThr datos 0.5 >f1        # calculamos los throughputs en f1
$ gnuplot
    G N U P L O T
    Version 4.2 patchlevel 6
    [...]
Terminal type set to 'wxt'
gnuplot> set xlabel "tiempo (s)"
gnuplot> set ylabel "bytes recibidos"
gnuplot> plot "f1" using 1:2 with lines
```

O puede dibujar el trafico acumulado con la tercera columna

```
gnuplot> plot "f1" using 1:3 with lines
```

Es fácil dibujar una columna frente a otra de un fichero. Si quiere aprender más sobre gnuplot consulte la ayuda con help o pregunte al profesor para que le guíe. Con eso puede comparar sus resultados con los de wireshark

CHECKPOINT 1: (3%)

Muestre al profesor que su programa funciona con ejemplos, capturando tráfico en el momento para analizarlo

Suba el código fuente de su programa en un fichero TimeThr.java en la página web de la asignatura

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

4- Mejorando el programa

Queremos que el programa sea capaz de separar el trafico que envían diferentes máquinas identificadas por su dirección ethernet. Para ello queremos tener un filtro que nos permita centrarnos en algunos paquetes. En ese caso indicaremos el origen que nos interesa usando el siguiente formato

Uso:

java TimeThr2 <nombrefichero> <duración del intervalo> <origen>

El programa hace lo mismo que el anterior, pero calcula ademas las estadísticas para el origen indicado. Origen debe ser una dirección MAC en el mismo formato que las saca el tshark

Si el origen está presente al leer el fichero de entrada se leerán cuatro parámetros por cada línea

<tiempo> <tamaño> <mac origen> <mac destino>

La salida será similar a la del primer programa pero teniendo 3 columnas mas con las estadísticas del origen indicado.

<tiempo> <bytes> <acumulados> <thr> <bytes_de_orig> <acumulados_de_orig> <thr_de_orig>

Ejemplos:

Para contar solo el throughput que envíe una máquina

```
$ java TimeThr2 fichero1 0.1 00:26:4a:12:3f:a2
```

CHECKPOINT 2: entregue el programa con la mejora (1%)

Suba el código fuente de su programa en un fichero TimeThr2.java en la página web de la asignatura. No es necesario enseñarlo al profesor de prácticas pero si debe hacerlo durante la duración de la sesión de laboratorio.