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 6 – Programando en Java (3)

## 1- Objetivos

El objetivo de esta práctica es avanzar con la programación en Java, mejorando las habilidades necesarias para hacer programas que procesen ficheros y utilizándolas para realizar un programa que obtenga información sobre una traza de tráfico de red.

## 2- Capturando tráfico Ethernet

Utilice 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.

## 3- Analizando el throughput

Realice un programa que lea 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 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
```

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

```
      0.588422000 92
      ...
      $ java TimeThr fichero1 0.1
      0.100 244 2439.9
      0.200 60 599.9
      0.300 180 1799.9
      0.400 591 5909.9
      0.500 240 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
```

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
```

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

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*

**CHECKPOINT 1:**

**Muestre al profesor que su programa funciona con ejemplos, capturando tráfico en el momento para analizarlo (3%)**

**Suba el código fuente de su programa en un fichero TimeThr.java en la página web de la asignatura (puntuación 2%)**

**El programa entregado no debe hacer las funciones pedidas en la parte avanzada. Entregue el programa cuando lo tenga o guarde una versión que haga solo el checkpoint 1**

## 4- Mejorando el programa

Elija una de las siguientes mejoras propuestas e incorpórelas al programa. Sólo se puede entregar una. No se puntúa por cantidad, más bien el entregar un programa incorrecto "a ver si cuela" se considerará no ser capaz de decidir si el programa cumple lo que se pide. Hay que saber valorar cuál de las mejoras es capaz de hacer en el tiempo disponible.

### Mejora 1

Queremos que el programa sea capaz de calcular el histograma de los tamaños de paquetes vistos en la red. Dado que los tamaños de paquetes pueden ser muy variados se le indicará al programa un valor de tamaño de cubo. El programa debe contar cuántos paquetes (y que porcentaje) hay en cada cubo. Por ejemplo si se indica que queremos el histograma en cubos de tamaño 100 quiere decir que debemos contar cuántos paquetes hay en cada cubo [0-100],[101-200],[201-300]… El programa debe proporcionar el número de paquetes y la frecuencia en cada cubo

**Uso:**

**java TimeThr <nombrefichero> -h <tamaño cubo>**

El programa hace lo mismo que el anterior, pero si el segundo parámetro en lugar de un tiempo es –h el programa calculará el histograma de los tamaños y lo imprimirá por pantalla

La salida será una línea por cada cubo de interés indicando la cantidad de apariciones de esos tamaños y la frecuencia con respecto al total. Tambien indicará el cubo observado, el menor y mayor valor y la mitad del intervalo por si se quiere dibujar. El formato de la salida debe ser el que se ve en el ejemplo. Si a partir de un cubo no hay más valores observados el resto pueden omitirse.

**Ejemplos:**

```
$ java TimeThr fichero1 –h 100
# Total: 15 paquetes observados
[1-100] 1 100 50 : 11 0.7333
[101-200] 101 200 150 : 2 0.1333
[201-300] 201 300 250 : 2 0.1333
$
```

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*

## Mejora 2

Queremos que el programa sea capaz de realizar la tabla de tráfico frente al tiempo pero filtrando sólo los paquetes enviados o recibidos por diferentes máquinas. Para ello queremos tener un filtro que nos permita centrarnos en algunos paquetes. En el caso en que indiquemos filtros el programa esperará que las líneas de entrada tengan cuatro valores

**Uso:**

**java TimeThr <nombrefichero> <duración del intervalo> <origen> <destino>**

El programa hace lo mismo que el anterior, pero solo cuenta los paquetes si tienen el origen y destino indicados. Origen y destino pueden ser una dirección MAC en el mismo formato que las saca el tshark o bien pueden ser la palabra "any" para indicar que nos da igual cualquier destino o cualquier origen.

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

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

La salida será igual que en el primer programa pero teniendo en cuenta solo los paquetes que pidamos.

**Ejemplos:**

Para contar solo el throughput de los paquetes que vayan a broadcast

```
$ java TimeThr fichero1 0.1 any ff:ff:ff:ff:ff:ff
```

Para contar solo el throughput que envíe una máquina a otra

```
$ java TimeThr fichero1 0.1 00:26:4a:12:3f:a2 00:26:4a:32:f1:18
```

**CHECKPOINT 2: entregue una de las mejoras**

**Suba el código fuente de su programa en un fichero TimeThr.java en la página web de la asignatura (puntuación 3%). No es necesario enseñarlo al profesor de prácticas pero si debe hacerlo durante la duración de la sesión de laboratorio.**