

Prácticas 2 y 4 – Programando en Java

Objetivos

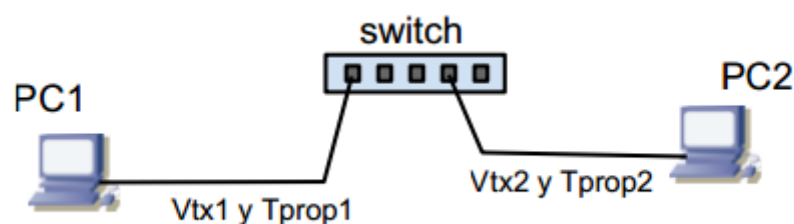
El objetivo de estas prácticas es familiarizarse con la programación en Java, adquiriendo las habilidades mínimas necesarias para hacer programas que procesen la entrada estándar, usen la información que contienen ficheros, realicen cálculos y muestren resultados por la salida estándar o los guarden en ficheros. Para ello se propone realizar programas que permitan calcular fórmulas que resultarán útiles como herramienta en la resolución de ejercicios y problemas, así como llevar a cabo procesamientos sencillos de datos para el análisis de tráfico en redes.

Estas sesiones de prácticas no se evalúan, por tanto aproveche el tiempo para afianzar los conocimientos de programación en Java impartidos en las clases de teoría y consultar todas sus dudas.

1. Tiempos de envío de paquetes en red

1.1 Escenario

Tenemos una red formada por un conmutador Ethernet en el que hay conectados dos ordenadores. Se pretende realizar un programa que calcule el tiempo que empleara el envío de un paquete de la longitud determinada entre los dos y que genere los datos necesarios para plasmar en una gráfica este tiempo de entrega de datos en función del tamaño que se desee enviar.



1.2 Especificación

Realice un programa que calcule el tiempo que tarda un paquete en atravesar el sistema para un conjunto de tamaños a partir de los variables:

- $p1Vtx$ - velocidad de transmisión del puerto 1 (en bps)
- $p1Tprop$ - tiempo de propagación en el cable conectado al puerto 1 (en s)
- $p2Vtx$ - velocidad de transmisión del puerto 2 (en bps)

- p2Tprop - tiempo de propagación en el cable conectado al puerto 2 (en s)
- tam1 - tamaño inicial del barrido (en bytes)
- tam2 - tamaño final del barrido (en bytes)
- tamdelta - incremento del tamaño en el barrido (en bytes)

Uso:

```
java Grafica
```

Imprime una línea por cada tamaño indicado en las variables desde tam1 hasta tam2 en incrementos tamdelta. En cada línea indica el tiempo que tardara un paquete de ese tamaño en ser entregado en el ordenador 2 desde que empezó a ser enviado por el ordenador 1. El formato de la línea será

```
<tamaño del paquete en bytes> <tiempo en segundos empleado>
```

Las variables indicadas estarán definidas en el propio fichero Grafica.java

Ejemplos:

```
$ java Grafica
500 0.004402
750 0.006602
1000 0.008802
1250 0.011002
1500 0.013202
```

Pruebe el programa generando diferentes barridos con diferentes valores. Para dirigir la salida a un fichero por ejemplo caso1 puede hacer

```
$ java Grafica > caso1
```

Pruebe a obtener la gráfica con el programa gnuplot.

```
$ gnuplot
> plot "caso1" using 1:2 with linespoints
```

1.3 Versión avanzada

Realice un programa en java que cumpla con el siguiente patrón.

Uso:

```
java Grafica <puerto1Vtx> <puerto1Tprop> <puerto2Vtz>  
<puerto2Tprop> <tamaño> [<tamaño max> <incremento>]
```

Si tamaño max no está presente calcula el tiempo en segundos que emplea un paquete del tamaño indicado en atravesar el sistema desde que el ordenador 1 empieza a enviarlo por el puerto 1 hasta que el ordenador en el puerto 2 lo recibe completamente. El formato de salida será una única línea con dos campos

```
<tamaño del paquete en bytes> <tiempo en segundos empleado>
```

Los parámetros de entrada puerto1Vtx y puerto2Vtx se deben interpretar como valores en bps. Los parámetros de entrada puerto1Tprop y puerto2Tprop deben interpretarse como valores en segundos. Los parámetros de entrada tamaño y tamaño max si está presente deben interpretarse en bytes.

Si tamaño max esta presenta el programa generara el resultado para cada tamaño desde el indicado hasta tamaño max en el incremento indicado

Ejemplos :

```
$ java Grafica 10000000 0.000001 1000000 0.000001 500  
500 0.004402  
$ java Grafica 10000000 0.000001 1000000 0.000001 500 1500 250  
500 0.004402  
750 0.006602  
1000 0.008802  
1250 0.011002  
1500 0.013202
```

2. Simulando transmisiones en CSMA

2.1 Escenario

Tenemos un sistema CSMA p-persistente, en el que cuando voy a transmitir y encuentro el canal libre se decide si transmitir o no con una probabilidad p. En cuanto veo libre el canal con una probabilidad p transmito y con una probabilidad 1-p espero un tiempo Δt . Si espero que pase ese Δt microsegundos con una probabilidad p transmito y con 1-p vuelvo a esperar otros Δt y así sucesivamente. Eso quiere decir que a veces tardare más y otras menos en transmitir. Queremos

estudiar este tiempo. Para eso haremos un programa que haga experimentos, pruebe una y otra vez a transmitir y al final haga estadísticas.

2.2 Especificación

Realice un programa que haga una simulación del tiempo que tarda un paquete en empezar a transmitirse con el canal vacío en CSMA p-persistente.

Uso:

```
java SimulaCSMAp p deltat
```

Simula la transmisión en CSMA p-persistente generando probabilidades para ver cuando se decide a transmitir imprimiendo cada paso para comprobar el funcionamiento

Ejemplos:

```
$ java SimulaCSMAp 0.3 0.000001
Intentando transmitir con persistencia p=0.3000
slot 1 ... espero
slot 2 ... espero
slot 3 ... espero
slot 4 ... espero
slot 5 ... transmito
conseguido en 5 intentos tiempo= 0.000004 s
```

Pruebe el programa con diferentes valores asegurándose que da resultados razonables

2.3 Calcular estadísticas

Realice un programa que haga la simulación anterior n veces calculando cuantas veces el resultado es 1 slot, 2 slots... para poder obtener el histograma. Como teóricamente podría esperar infinitos slots le diremos un número máximo de slots en que estamos interesados.

Uso:

```
java SimulaCSMAp p deltat n max
```

Simula n veces la transmisión en CSMA p-persistente generando el histograma de los max primeros slots (0 a max-1)

Ejemplos:

```
$ java SimulaN 0.3 0.000001 8 5
transmisión en 1 intentos
```

```
transmisión en 2 intentos
transmisión en 1 intentos
transmisión en 1 intentos
transmisión en 1 intentos
transmisión en 3 intentos
transmisión en 3 intentos
transmisión en 1 intentos
```

Histograma:

```
0 slots -- 0 veces (0.00 %)
1 slots -- 5 veces (62.50 %)
2 slots -- 1 veces (12.50 %)
3 slots -- 2 veces (25.00 %)
4 slots -- 0 veces (0.00 %)
Más de 5 slots -- 0 veces (0.00 %)
```

Pruebe el programa con diferentes valores asegurándose que da resultados razonables, incluyendo que haga el experimento miles de veces (puede quitar el imprimir cada caso y dejar solo el histograma final)