

Redes Multimedia

Práctica 1: Introducción a redes multimedia

Turno y pareja: 4312 / Pareja 02

Integrantes:

Jorge Gutiérrez Díaz

Javier Martín González

Fecha de entrega:

Contenido

Contenido	1
1 Introducción	2
2 Realización de la práctica	2
3 Conclusiones.....	7

1 Introducción

Se ha desarrollado un módulo emulador que transmite paquetes multimedia desde un emisor al receptor. Este emulador permite introducir retrasos en los paquetes y originar pérdidas para analizar las consecuencias de tales acciones. Existen dos versiones, una que utiliza un único hilo para todos los paquetes y otra que utiliza varios hilos para manipular los paquete. Para los retardos de los paquetes se utilizará una distribución uniforme o gaussiana.

2 Realización de la práctica

1. Se realizará un módulo emulador que reciba paquetes multimedia por un puerto y los retransmita a un destino en otro puerto. Dicho módulo simulará comportarse como la red que hay habitualmente entre un servidor y un cliente. Para implementarlo, puede aprovechar el código de la práctica anterior modificándolo adecuadamente.

Para realizar esta simulación, el emulador añadirá un retardo variable (jitter) y una probabilidad de pérdida a cada paquete recibido a la hora de retransmitirlo al destino final. Estos valores deberán ser parametrizables como argumentos de entrada al programa:

- 1.1. Direcciones y puertos donde escucha el emulador, y a dónde se enviarán finalmente los paquetes.
- 1.2. Probabilidad de pérdida de paquete (valores entre 0 y 1).
- 1.3. Rango de variación del retardo, suponiendo una distribución uniforme.

Por ejemplo, para ejecutar el programa, deberá escribir:

```
python emulador.py <dirección_escucha> <puerto_escucha>  
<dirección_destino> <puerto_destino> <pérdida>  
<retardo_mínimo_en_ms> <retardo_máximo_en_ms>
```

Y un ejemplo de valores sería:

```
python emulador.py 127.0.0.1 5002 127.0.0.1 5004 0.01 0 100
```

Escriba aquí comentarios respecto del código realizado y cómo lo invoca desde la línea de comandos.

El problema del proxy con un solo hilo es el colapso de paquetes en el buffer. Al solo existir un hilo que se encargue de los paquetes y que el retardo aplicado al paquete actual es sufrido indirectamente por el paquete siguiente, hace que el buffer se llene de paquetes y se acaben descartando.

Se ha vuelto a utilizar el código desarrollado en la práctica 0 para la apertura de sockets. Creamos dos socket de la familia AF_INET, IP y puerto para la comunicación, que envía paquetes UDP por la red. Para el socket de recepción, hemos aumentado el tamaño del buffer para mejorar la calidad de reproducción. Con respecto a la funcione random para aumentar el *jitter*, se utiliza la función de `random.uniform(a, b)` para obtener un valor entre el retardo máximo y mínimo.

A continuación incluimos un ejemplo de ejecución:

```
python3 proxy1.py 127.0.0.1 5004 150.244.66.52 5006 0.01 0 10
```

2. Utilice el programa VLC para enviar un vídeo al puerto en que escucha el emulador, y otro VLC para recibir el vídeo desde el puerto de destino al que envía el emulador. Indique los parámetros que utiliza para emitir el vídeo y para recibirlo, explicando qué posibilidades existen en cada caso, y el motivo de las escogidas.

Escriba aquí los parámetros que utiliza para emitir y recibir el VLC.

Perdidas = 1%, retardo mínimo = 0 ms, retardo máximo = 10 ms

Escriba aquí las posibilidades que existen.

Se puede probar una transmisión de hasta un 9% de pérdidas ya que un valor superior hace que el video sea imposible de visualizar. Los retardo están acotados de 0 a 1000 ms porque un retardo de más de 1000 ms se vuelve molesto de ver

Escriba aquí el motivo de la escogida.

Decidimos utilizar un 1% de pérdidas para poder apreciar las consecuencias que se producen en el vídeo, y notamos que el reproductor replicaba paquetes o que se rellenaban con píxeles verdes para intentar solucionar las pérdidas

3. Desarrolle una segunda versión del módulo (emulador2.py) para que el retardo introducido a un paquete no dependa del paquete anterior, utilizando para ello *threads*. Explique qué resuelve con este cambio.

El problema de no utilizar hilos implica que el retardo que se aplica a un paquete también es sufrido por el paquete siguiente, un fenómeno que no es nada habitual es la realidad. El uso de hilos soluciona este problema ya que se le aplica un retardo a cada paquete de forma independiente.

Para cada paquete que se recibe, se crea un hilo para calcular el retardo que sufre ese paquete. Dentro de la función calcula que realiza el hilo, comprobamos en primer lugar si el paquete es desechado si el valor obtenido en la función `random.random()` es menor que el valor de pérdidas determinado en los parámetro de ejecución. En el caso de que el paquete no sea desechado, se introduce un retardo mediante la función `time.sleep(a)`, cuyo valor esta determinado por un valor aleatorio entre los retardo máximo y mínimo ,que se pasan como argumentos, mediante la función `random.uniform(max, min)`.

4. Estudie la diferente degradación que se produce con la introducción de jitter y de pérdidas en la transmisión del contenido multimedia, valorándola de acuerdo a la escala MOS (*Mean Opinion Score*, Puntuación de Opinión Promedio):

MOS	Calidad	Degradación
5	Excelente	Imperceptible
4	Buena	Perceptible pero no molesta
3	Justa	Ligeramente molesta
2	Pobre	Molesta
1	Mala	Muy molesta

Utilice para ello los siguientes valores de pérdida de paquetes y de *jitter*:

- Porcentaje de pérdida de paquetes: {0;0,2; 0,5; 0,9; 2; 5; 9}%,
- Retardo mínimo y máximo: {(0, 0); (0, 250); (0, 500); (0,750); (0, 1000)} (ms)

En caso de que no observe variaciones sustanciales con algunos de los valores anteriores, indique para qué valores ha observado variaciones en el MOS.

Rellene la tabla siguiente con los resultados para el integrante 1 de la pareja.

Retardo/Pérdidas	0	0,2%	0,5%	0,9%	2%	5%	9%
(0, 0)ms	4	4	3	3	3	2	1
(0, 250) ms	2	1	1	1	1	1	1
(0, 500) ms	1	1	1	1	1	1	1
(0, 750) ms	1	1	1	2	1	1	1
(0, 1000) ms	1	1	1	1	1	1	1

Rellene la tabla siguiente con los resultados para el integrante 2 de la pareja.

Retardo/Pérdidas	0	0,2%	0,5%	0,9%	2%	5%	9%
(0, 0)ms	4	4	3	3	2	1	1
(0, 250) ms	2	2	1	1	1	1	1
(0, 500) ms	1	1	1	1	1	1	1
(0, 750) ms	1	1	1	1	1	1	1
(0, 1000) ms	1	1	1	1	1	1	1

Rellene la tabla siguiente con los resultados promediados entre ambos integrantes.

Retardo/Pérdidas	0	0,2%	0,5%	0,9%	2%	5%	9%
(0, 0)ms	4	4	3	3	2	1	1
(0, 250) ms	2	1	1	1	1	1	1
(0, 500) ms	1	1	1	1	1	1	1
(0, 750) ms	1	1	1	1	1	1	1
(0, 1000) ms	1	1	1	1	1	1	1

5. Tras el estudio anterior, valore qué problema considera más molesto desde el punto de vista del usuario, y cuál cree que puede darse más frecuentemente en la red. Razone hasta qué punto las pérdidas y retardos propuestos son habituales o no en la red.

Escriba aquí su valoración sobre qué problema considera más molesto, y cuál se dará con más

frecuencia en la red y en qué magnitud.

El mayor problema que hemos sufrido ha sido el de las pérdidas, ya que en este caso la visualización del vídeo se vuelve muy molesta y únicamente se ven breves fragmentos con un lapso de tiempo entre uno y otro muy alto (entre medias se producen paradas constantes de vídeo y cuadros verdes que ocupan casi la totalidad de la pantalla).

En la red el problema más habitual será el de los retardos debido al colapso que habrá por el resto del tráfico. No obstante, ello también puede derivar a pérdidas en menor medida en caso de que el retardo sea tan alto que provoque una saturación de los buffers de los routers.

Escriba aquí su razonamiento sobre lo habitual o no de los rangos de pérdidas y retardos propuestos en una red.

Tras el estudio realizado, con unas pérdidas del 9% es ya imposible visualizar nada, por lo que cualquier valor superior a éste dará todavía peores resultados. En cuanto al tema de los retrasos, aquellos mayores a 500 ms (medio segundo) también provocan una gran cantidad de molestias a la hora de la reproducción y es inviable ver un vídeo de esa forma. Por ello, cualquier valor superior a medio segundo provocaría todavía peores calidades.

Unos retrasos de esas magnitudes si pueden ocurrir una red por la congestión de la misma como hemos comentado en el apartado anterior, de ahí la importancia de una buena gestión de la reproducción mediante buffer con el fin de que el usuario no sea consciente de esto (aunque el usuario reciba el vídeo unos segundos más tarde de la emisión). En cuanto a las pérdidas, el caso del 9% en adelante es un valor muy alto que no debería ocurrir en una red que esté bien diseñada.

6. Indique de manera aproximada cuánto tiempo tarda inicialmente el VLC en comenzar la reproducción del vídeo, e indique a qué se debe.

Escriba aquí su estimación del tiempo que tarda inicialmente el VLC en comenzar a reproducir el vídeo, e indique a qué se debe.

Desde que le damos a reproducir tarda 1-2 segundos en empezar la reproducción de la misma y poder visualizar la imagen que se está emitiendo. Esto es así por el buffer que tiene VLC, con el fin de compensar el jitter.

7. Varíe los parámetros de que dispone VLC para compensar el *jitter*. Indique qué valores modifica y qué impacto tienen en la calidad percibida. Explique si considera adecuados estos valores para una comunicación interactiva en la que dos usuarios quieran mantener una videoconferencia.

Escriba aquí qué parámetros dispone VLC para compensar el Jitter, cuál ha utilizado y con qué valor.

Dentro de la configuración de VLC, a la hora de intentar reproducir un recurso desde la red, se puede modificar el parámetro de Network Caching (opción oculta para lo que hay que pinchar

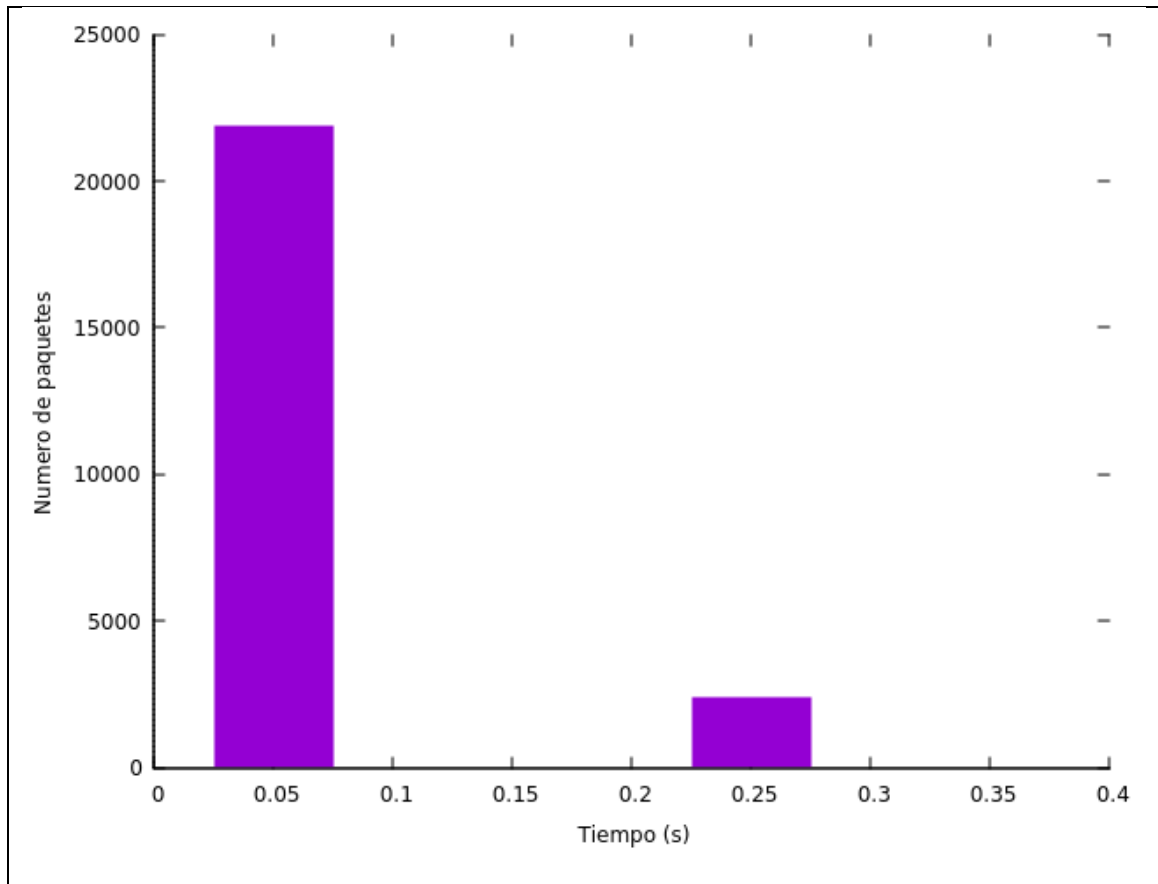
en “más opciones”). Se puede cambiar este valor que por defecto está a 1000 ms tanto hacia arriba como hacia abajo. Hemos aumentado y bajado ese valor, obteniendo unos mejores resultados con valores más altos, ya que esto implica una ampliación del buffer.

Escriba aquí si los valores son adecuados en una comunicación interactiva tal como la videoconferencia.

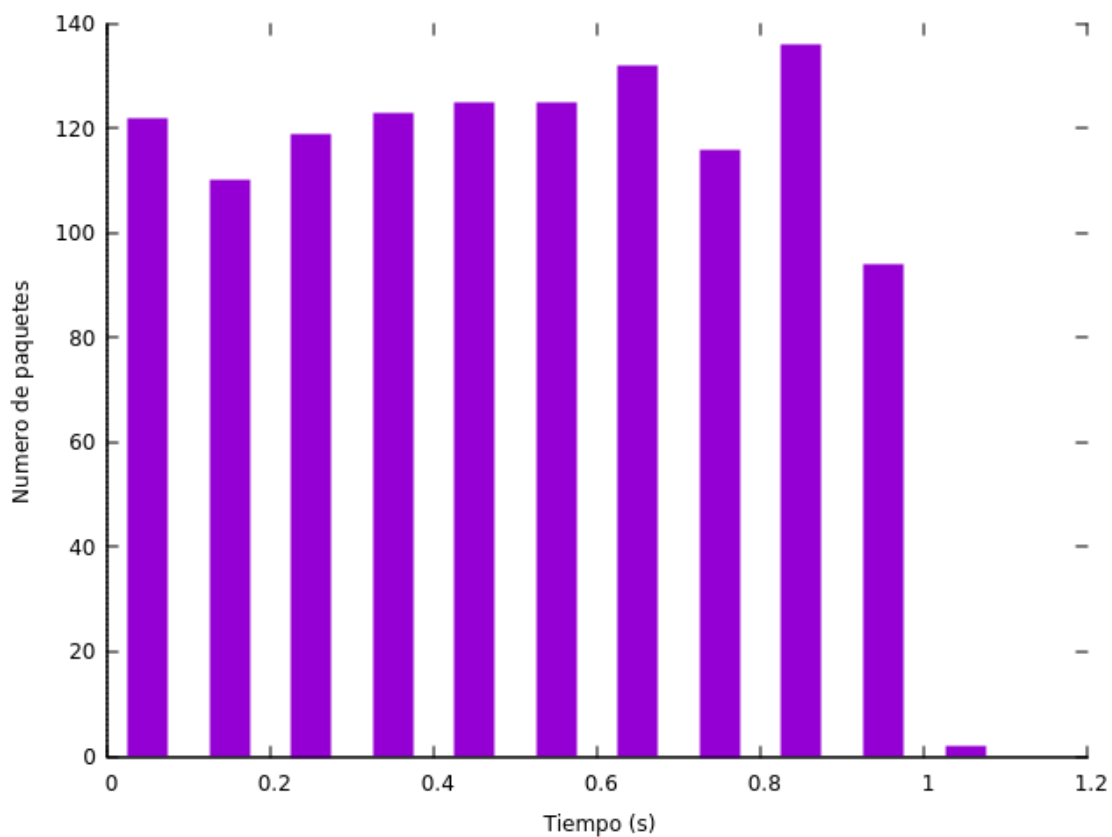
El aumento del valor hacia arriba es adecuado para una comunicación interactiva como es el caso de una videoconferencia, la cual es a tiempo real. Cuando hemos aumentado el valor se ha obtenido una mejor respuesta y los periodos de tiempo desde la emisión hasta la recepción son cortos y aceptables, considerando que si sería aceptable para la videoconferencia propuesta.

8. Capture el tráfico y analice cómo varían los tiempos entre llegadas en el emisor y el receptor para el caso de *jitter* máximo. Obtenga un histograma para ambos casos y evalúe la influencia del *jitter*.

Ponga aquí el histograma obtenido como producto de medir los tiempos entre llegadas en el emisor para el caso de *jitter* máximo.



Ponga aquí el histograma obtenido como producto de medir los tiempos entre llegadas en el receptor para el caso de jitter máximo.



¿Qué diferencia observa entre ambos histogramas? ¿Qué influencia tiene el jitter?

En el lado emisor, los paquetes llegan al proxy con un tiempo constante y ocasionalmente suelen tener un retardo distinto a lo normal.

Con respecto al lado receptor, al haber introducido *jitter* en la carga el tiempo de llegada de los paquetes varía dependiendo del retardo que haya ocasionado el proxy.

9. Genere una tercera versión del módulo (`emulador3.py`) para que la variación del retardo siga una distribución gaussiana con un valor medio y una cierta varianza. Tenga en cuenta que el valor del retardo como mínimo debe ser 0. Los parámetros de entrada de este tercer módulo deben ser iguales a los dos anteriores.

Escriba aquí comentarios respecto a cómo ha hecho para que su código genere retardos según una distribución gaussiana.

Para hacer la distribución gaussiana con respecto a los valores de entrada dados, en primer lugar, hay que calcular el correspondiente μ y σ . Tras ello, obtenemos un valor de retardo según el retorno de la función `random.gauss`. Dicho valor lo verificamos dentro de nuestro rango de pérdidas `[min, max]`. Si se cumple que está dentro del rango, se continúa con la ejecución del hilo, y sino se vuelve a calcular hasta que esté dentro del rango.

¿Qué valores de media y desviación ha utilizado a partir del retardo mínimo y máximo introducidos por línea de comando?

Para obtener μ (la media), teniendo los valores máximo y mínimo recibidos por parámetro, se haya la media aritmética entre ambos.

Para obtener σ (la desviación), al valor de μ que acabamos de calcular se le resta el valor mínimo recibido por parámetro, y el resultado de ello se divide por 3.

¿Qué diferencias aprecia con respecto al retardo que sigue una distribución uniforme en el experimento del MOS para jitter máximo? ¿A qué se debe?

Para el lado receptor, el retardo con una distribución gaussiana sigue una campana de gauss, es decir, la mayoría de los paquetes están centrados en un retraso intermedio, mientras que en la distribución uniforme la mayoría de los paquetes tienen un retardo pequeño, aunque ocasionalmente se encuentra algún paquete con retardo mayor.

Para el lado emisor, al no pasar por el proxy, el retardo era idéntico.

3 Conclusiones

Hemos sacados varias conclusiones tras haber realizado esta práctica.

En primer lugar, al utilizar un solo hilo para tratar cada paquete, hemos observado que la calidad era buena, pero según iba pasando el tiempo el vídeo comenzaba a verse mal debido al retardo que iba acumulando cada paquete. Esto es así porque el buffer se saturaba hasta el punto de que empezaba a descartarse paquetes. Por lo tanto, el retardo era el parámetro más influyente para este caso.

El problema del primer proxy se resolvía utilizando varios hilos, porque al existir más de un hilo que pueda tratar cada paquete el retardo aplicado al anterior paquete no era influyente. En este caso, el problema para estos proxys era el porcentaje de pérdidas porque a partir de un

valor $> 1\%$, el vídeo comenzaba a replicar paquete o utilizar píxeles verdes para solucionar las pérdidas lo que resultaba, en algunos momentos, molesto de ver, pero no tanto como en el proxy de un solo hilo.

En cuanto a la diferencia entre gaussiana y uniforme, las diferencias únicamente afectan al receptor, no al emisor. En el caso gaussiana los retrasos se concentran en base a un retraso intermedio como si fuese una campana de gauss y en el caso de la distribución uniforme, se aprecia que la mayoría de los paquetes tienen un retardo más pequeño en líneas generales. En una red real, se daría más la circunstancia de que existan retardos con distribuciones uniformes.