

Redes Multimedia

Práctica 1: Introducción a redes multimedia

Turno y pareja: 2461_02

Integrantes:

Henrique Hiram Libutti Núñez

Alvaro Medinilla Armenteros

Fecha de entrega:

Contenido

1 Introducción	2
2 Realización de la práctica	2
3 Conclusiones.....	5

1 Introducción

En esta práctica hemos implementado un módulo emulador.py, que parte del realizado en la práctica 0 pero que hemos adaptado. Las tres implementaciones que se realizan del emulador se han juntado en el mismo main, para poder ejecutarlas con línea de comandos modificando las opciones según la ejecución deseada.

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
```

El programa se ejecuta siguiendo las instrucciones definidas en el enunciado de la práctica, usando variables globales de los argumentos recibidos por parámetros. Cabe destacar que se ha utilizado el módulo *argparse* que gestiona de manera más cómoda la entrada de parámetros.

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.

Primero abrimos el VLC y seleccionamos *Abrir ubicación de red* y se pone `rtp://127.0.0.1:<puerto>`, en este caso, el 5051, por ejemplo.

****Se pueden poner opciones de caché (que es el buffer de recepción)**

Después abrimos otra ventana de VLC y le damos a emitir añadiendo el archivo que queremos reproducir.

Seleccionamos el método de emisión *de RTP / MPEG Transport Stream*.

Luego añadimos la dirección del localhost (127.0.0.1) junto con el puerto base, en este caso el 5050 (por ejemplo), y ponemos un nombre a la emisión.

Luego en la siguiente elegimos el perfil de Video – H.264 + MP3 (MP4).

Seleccionamos *habilitar transcodificar* (por defecto).

En la siguiente ventana podemos *emitir las emisiones elementales* y lo seleccionamos.

Finalmente le damos a *emitir*.

En el método de emisión se pueden elegir diferentes métodos de emisión.

Se pueden elegir más perfiles de Video.

Se puede elegir transcodificar o no.

Se puede elegir transmitir las emisiones elementales.

En este caso usamos RTP para transmitir (UDP)

El perfil del video es en MP4 que es como esta codificado el video.

Por defecto se deja la opción de transcodificar si quieres elegir otra codificación para el video

En este caso elegimos transmitir las emisiones elementales para que se transmita toda la información posible, como puede ser audio, etc.

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.

Se llama a un hilo para mandar un mensaje y así los mensajes no dependan de los otros en la transmisión de los paquetes.

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.

Integrante 1:

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

Integrante 2:

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

Resultado promedio:

Retardo/Pérdidas	0	0,2%	0,5%	0,9%	2%	5%	9%
(0, 0) ms	5	3.5	4	3	2	1.5	1
(0, 250) ms	5	3.5	3	3	2	1	1
(0, 500) ms	4	3	3	2.5	2	1	1
(0, 750) ms	3	3	2	1.5	1.5	1	1
(0, 1000) ms	2.5	2	2	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.

Nosotros consideramos que, en este caso, la pérdida de paquetes nos es más molesta. Creemos que el retardo se da con más frecuencia en la red. Lo aceptable es tener un retardo de hasta 400 ms.

Lo habitual es tener un retardo inferior a 100 aunque depende de la distancia del punto de acceso al servidor.

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

Alrededor de 2 o 3 segundos. Eso se debe al tiempo que pasa hasta que se rellena el buffer y puede empezar a reproducir.

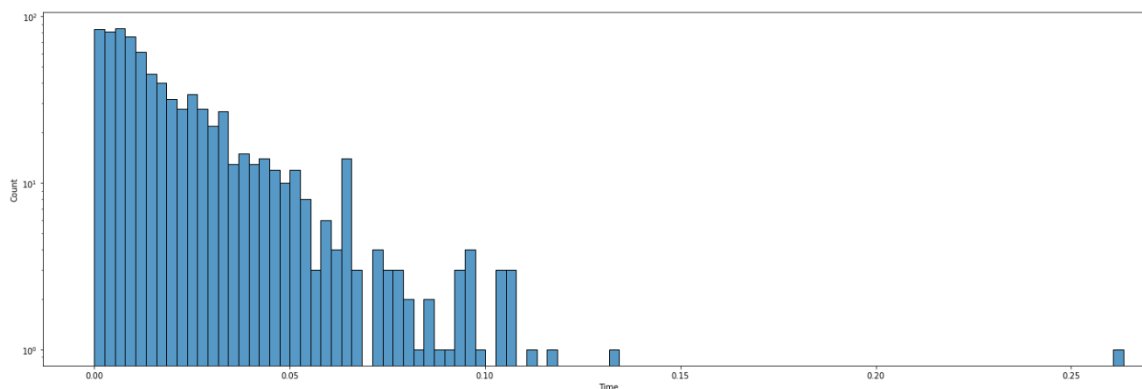
7. Varíe los parámetros de que dispone VLC para compensar el *jitter*. Indique qué valores modifica y qué impacto tienen en la calidad percibida. Explique si considera adecuados estos valores para una comunicación interactiva en la que dos usuarios quieran mantener una videoconferencia.

En este caso solo sabemos editar el tamaño de la caché. Por defecto está en 1000 ms aunque la hemos reducido a 500 ms y no ha sido muy satisfactorio.

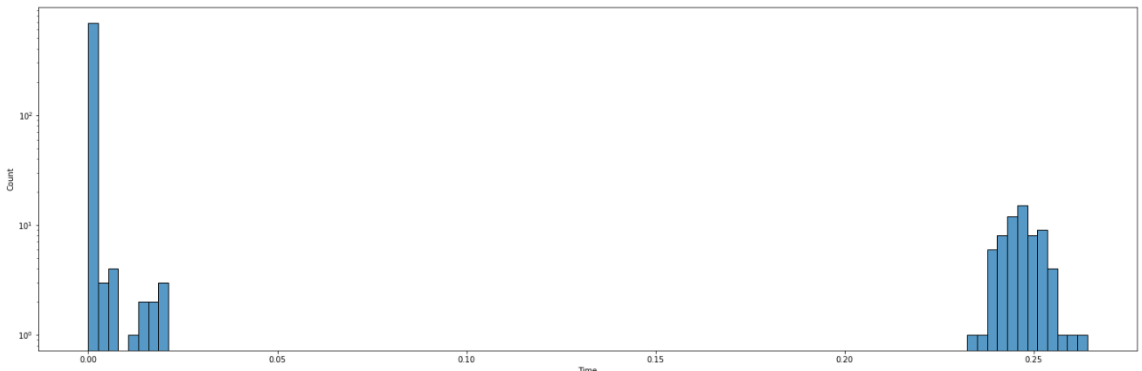
En este caso es preferible que se pierdan paquetes a que se ralentice la transmisión, por lo que debería reducirse el tamaño de la caché.

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*.

Histograma emisor:



Histograma receptor:



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

Se observa que el histograma del receptor tiene más varianza por el retardo impuesto por el proxy, en cambio, los retardos del emisor siguen una forma uniforme.

El jitter genera esa distribución desigual entre los tiempos de llegada de paquetes.

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 las dos anteriores.

Se ha llamado a la función de *normalvariate* del paquete *random* por sus mecanismos de *thread safety*

<p>La media μ es el retardo mínimo + el máximo entre 2</p> <p>Y la desviación es la μ - el retardo mínimo dividido entre 3 porque se requiere que el 99% de los retrasos queden entre el mínimo y el máximo.</p> <p>El jitter en este caso sería más constante y el buffer se comporta de manera óptima. Por lo tanto, la reproducción es mejor.</p> <p>Es debido a la forma de una distribución normal donde la mayoría de los retardos caerán alrededor de la media.</p>
--

3 Conclusiones

Hemos aprendido de manera visual como afecta el retardo y la pérdida de paquetes a un video y hemos analizado los resultados del MOS y cómo varían los resultados según si se usan hilos o una distribución uniforme o gaussiana para la generación de los retardos.

También hemos aprendido a usar y emitir video con el VCL y a recibirlo en otro puerto.