

REDES DE COMUNICACIONES II

Práctica 3 - Multimedia

[Volver a: Prácticas ➔](#)

4 Recursos

Para llevar a cabo la práctica se proporcionan una serie de recursos:

- La conexión inicial entre pares P2P necesita siempre un servidor de descubrimiento donde consultar su dirección IP y puerto. Para ello, se ha creado un servidor específico, que se describe en la siguiente sección, disponible en el host *vega.ii.uam.es*, puerto 8000.
- El esqueleto del cliente P2P se proporciona también hecho. En función de la funcionalidad que se implemente, puede ser necesario modificar el GUI existente, que está basado en este framework.
- Para la captura de imágenes de la webcam, o la transmisión de un fichero de video, el binding para Python de la librería OpenCV puede ser muy útil. A buen entendedor, pocas palabras bastan...

Cliente P2P

Como se ha comentado, el esqueleto del cliente P2P se proporciona hecho, en el repositorio Git de la asignatura. Si lo ejecutas en tu propia máquina, necesitarás instalar una serie de paquetes:

```
#sudo apt install python3-pil.imagetk python3-tk python3-numpy python-imaging-tk
```

Para OpenCV debe servir con `sudo apt install python-opencv`, aunque quizás sea necesario `pip3 install opencv-python`

Si en la máquina hay conectada una webcam, debería funcionar sin necesidad de configuración adicional con el siguiente comando:

```
# python3 video_client.py
```

Estudia el código para localizar dónde debes incluir la funcionalidad requerida.

Visualización de ambos vídeos simultáneamente

Ejemplo para superponer dos imágenes con OpenCV, una más pequeña (`frame_top`) dentro de la otra (`frame_base`)

```
frame_peque = cv2.resize(frame_top, (320,240)) # ajustar tamaño de la imagen pequeña

frame_compuesto = frame_base

frame_compuesto[0:frame_peque.shape[0], 0:frame_peque.shape[1]] = frame_peque
```

También se puede utilizar una ventana adicional que muestre el otro vídeo durante la llamada.

Esto son sugerencias, se aceptará cualquier solución que permita ver a la vez los dos vídeos (el emitido y el recibido).

Empaquetado de vídeo para la transmisión por red

Como se ha indicado, es necesario incorporar una cabecera a los datos de vídeo comprimidos en JPG, previamente a transmitir los frames por la red.

La cabecera es en texto plano y debe ir concatenada antes de los datos JPG en bytes.

Esto se puede conseguir de varias formas:

- Definiendo las cadenas como bytes poniendo "b" delante: b'cadena en bytes'
- Con la función bytearray() para hacer casting de strings a bytes.
- Con las funciones "struct.pack" y "struct.unpack" incluidas en la librería "struct" de python.
- Mediante otros métodos que se os ocurran y no afecten a la compatibilidad entre prácticas de otros grupos.

Volver a: Prácticas ➡