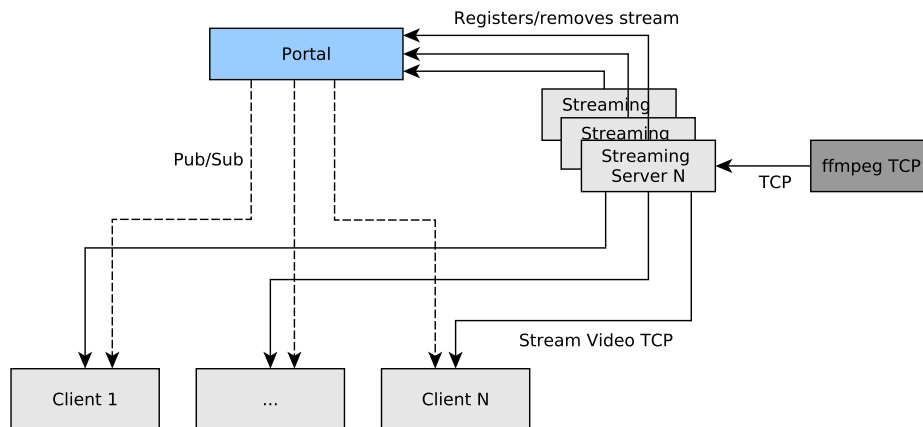


# Assignment II: Video Streaming

Distributed Systems 16/17

November 7, 2016

This assignment requires the implementation of a streaming facility. All the code must be written in Java, C/C++ or Python, with our recommendation being Java. The system **must have exactly** three components, as depicted below:



## Streamer

The streamer is responsible for streaming **one** video. But multiple streamers can and should be present in the system.

Mandatory features:

- Register in the portal as a provider of a video stream. The registration information must include: 1) stream name, 2) endpoint (transport, ip, port), 3) video size, 4) bitrate and 5) associated set of keywords, e.g., {a\_video\_stream\_name, {tcp, 127.0.0.1, 10000}, 480x270, 400, {basketball, Cavs, indoor, sports}}
- When closing a stream, the portal should be notified (through a message)
- Use **ffmpeg** to transcode the video to the desired format, e.g.:

```
ffmpeg -f mpegts -i videofile.mp4 -loglevel warning
-analyzeduration 500k -probesize 500k -r 30 -s 640x360
-c:v libx264 -preset ultrafast -pix_fmt yuv420p
-tune zerolatency -preset ultrafast -b:v 500k
-g 30 -c:a libfdk_aac -profile:a aac_he -b:a 32k
-f mpegts tcp://127.0.0.1:10000?listen=1
```

- Video codec (-c:v): libx264
- Audio codec (-c:a): libfdk\_aac
- Bitrate video (-b:v): target bitrate, e.g., 500k (Kbits/s)
- Bitrate audio (-b:a): target bitrate, e.g., 32k (Kbits/s)
- Video size (-s), e.g., 640x360
- Output is a TCP server listening at 127.0.0.1, port 10000. It only supports one client at a time.
- Create a streaming server **using sockets**:
  - Connects to ffmpeg output and buffers the current stream.
  - Create a server socket that listens to connections, and serves the streaming data (previously got from ffmpeg)

### **Portal**

Mandatory features:

- For streamers:
  - Registers streamers to allow a new video (Ice RPC)
  - Handle the closure of a streamer (Ice RPC)
- For clients:
  - Send stream list (Ice RPC)
- Portal:
  - Create a ICE publisher to announce new or delete streams (IceStorm)

### **Client**

Mandatory features:

- Get stream list from portal (Ice RPC)
- Subscribe to portal publisher to receive notifications of new or deleted streams (IceStorm)
- Use `ffplay` to play streams, e.g., `"ffplay tcp://127.0.0.1:10000"`
- In Java, you can use `"ProcessBuilder"` for launch a `ffplay` client.

The client must implement a CLI that supports the following commands:

1. `stream list`: List all streams
2. `stream search "keywords"`: List all streams that match "keywords"
3. `stream play "name"`: Play stream "name" (it should open a `ffplay` and return to the CLI, so non-blocking)

### **Bonus evaluation**

Extra credits can be achieved if the student implements the following items:

1. Implement UDP as a transport protocol
2. Use h265 instead of h264. Using the same bitrate compare two streams side-by-side
3. Use Erasure Coding to compensate packet loss in UDP:
  - openfec and Jerasure (C)
  - <https://github.com/Backblaze/JavaReedSolomon> (Java)
4. Implement support for Apple's HLS:
  - <https://www.keycdn.com/support/how-to-convert-mp4-to-hls/>
5. Implement support for Google's DASH:
  - [https://developer.mozilla.org/en-US/docs/Web/HTML/DASH\\_Adaptive\\_Streaming\\_for\\_HTML5\\_Video](https://developer.mozilla.org/en-US/docs/Web/HTML/DASH_Adaptive_Streaming_for_HTML5_Video)

**TIP:** Use NGINX, see <https://github.com/arut/nginx-rtmp-module>

Video Files:

Big Buck Bunny (Comedy, Animation)

<https://archive.org/details/PopeyeAliBaba>

The Letter, Lego movie [2003] (Comedy, Animation, LEGO), 6m30s

<https://archive.org/details/t1>

Charlie Chaplin's "The Vagabond" [1916] (Comedy), 24m43s

[https://archive.org/details/CC\\_1916\\_07\\_10\\_TheVagabond](https://archive.org/details/CC_1916_07_10_TheVagabond)

Night of the Living Dead [1968] (Sci-Fi / Horror), 95m17s

[https://archive.org/details/night\\_of\\_the\\_living\\_dead](https://archive.org/details/night_of_the_living_dead)

Popeye the Sailor Meets Ali Baba's Forty Thieves [1937] (Comedy, Animation), 16m58s

<https://archive.org/details/PopeyeAliBaba>