Systems and Networks

Daniel Hagimont

We want to implement a video streaming service. This service is composed of 2 parts: a video-library (RMI question) and a streaming-server (socket question).

RMI

In a first step, we want to implement the video-library allowing to register for a video the address (host, port) of its streaming-server.
A client (a video player program or an application wishing to register a video) can remotely invoke (with RMI) the video-library which implements a ***Library*** interface providing 2 methods:

```
void register(Video v);
Video lookup(String title);
```

Video is a class which implements a data structure (passed by **copy** as parameter of these methods) with the following fields:

- String title;
- String host;
- int port;

Video objects are registered in the video-library to indicate that the video identified by *title* is accessible for streaming on machine ***host*** and port ***port***.

For the moment, we are not considering the streaming of the video (this is the subject of the next question).

You have to implement with RMI the ***Library*** interface, the ***LibraryImpl*** class, the ***Video*** interface and the ***VideoImpl*** class.
The implementation of the *LibraryImpl* class will simply rely on a HashMap table for registering Video objects.
You also have to give an example of client program (a ***Client*** class) which invokes the 2 above methods.

Sockets

We now want to implement the streaming part with sockets.

A ***StreamingServer*** class can be used as follows: ***java StreamingServer OneTitle OneFile OnePort***
The parameters are giving respectively a video title, a file name (of a file local to the machine) and a port (for receiving connections). A *StreamingServer* serves only one file.

The *StreamingServer* must register the video in the video-library of the previous question, and then wait for incoming connections from clients. *StreamingServer* is multi-threaded. When a connection is accepted, *StreamingServer* sends directly the video content on that connection. To read the content of the video from the local file, *StreamingServer* will use the *FileInputStream* class. For the programming of IOs (with a file or a socket), we will use:

```
InputStream
        public int read(byte[] b); // blocking, returns the number of read bytes, -1 if end of stream
OutputStream
        public void write(byte[] b, int off, int len); // write len bytes at position off
```

You have to provide an implementation of the ***StreamingServer*** class.

You have to provide an implementation of a ***StreamingClient*** class, which takes a ***title*** parameter, query the video-library and connects with the *StreamingServer*. For each received buffer, *StreamingClient* calls a static method: **VLC.show(buffer);** which displays the video.