

# **CS5248: Systems Support for Continuous Media** **Spring 2010**

## **Project Assignment**

**Due: Report Draft – 5 Apr 2010 (11:59:59 pm)**

**Due: Code – 11 Apr 2010 (11:59:59 pm)**

**Due: Presentation & Final Report – 12 Apr 2010 (in class)**

**THIS IS AN INDIVIDUAL or TEAM-OF-2 HOMEWORK**

REMEMBER TO CHECK FOR ANNOUNCEMENTS ON THE CLASS WEB SITE:

<http://www.comp.nus.edu.sg/~cs5248.html>

### Homework Description

In this project, your task is to enable existing Yima Server PE v1.1 to communicate with an existing RTSP/RTP client software. Your implementation will be added to the Yima Personal Edition streaming media server code. You will be given the source code of Yima PE.

**NOTE: The description below may still be revised and updated a little bit in the near future.**

More specifically, your implementation with this framework consists of the following sub-tasks.

#### 1) **Implement RTSP Commands**

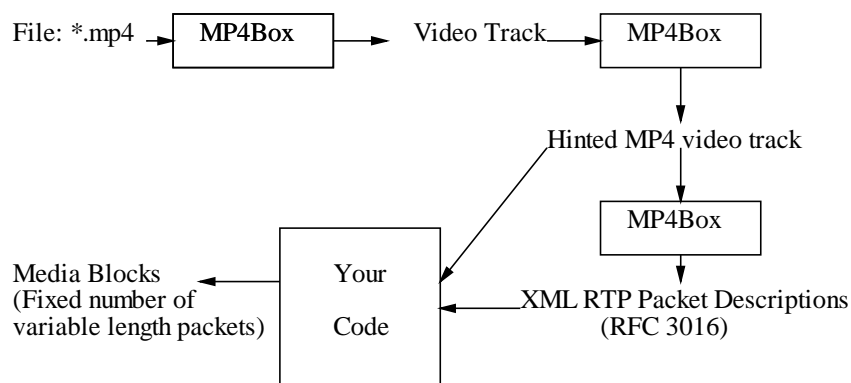
Your task is to revise the RTSP command-handling code in the Yima server such that it is compatible with the QuickTime Player (QTP). The Yima server understands basic RTSP commands, but they do not completely follow the latest standard. The individual steps for this task are as follows:

- Launch the QTP and play a movie from the Darwin Streaming Server (DSS). DSS is installed on the Cervino server and RTSP communication between the client and the server is on the standard port 554.
- Observe the RTSP communication between the client and the server with a packet sniffing program like WireShark or MS Network Monitor (see links below).
- The minimal command set that needs to be implemented is DESCRIBE, SETUP, PLAY, and TEARDOWN. (Pause and resume are optional.)
- The Yima server code related to RTSP processing is located in the `Server/RTSP` directory (i.e., the files `rtsp_front.c` and `rtsp_back.c`).

#### 2) **Create Media Blocks for Server**

Your second task is to produce media blocks that contain RTP packets such that the Yima server can load them and stream them to a client. The final goal is to create blocks that contain a fixed number of variable length RTP packets that are encoded according to the standard and recommendations of the RFC 3016 (RTP Payload Format for MPEG-4

Audio/Visual Streams). Note that we will deal only with the video portion, not the audio part (to simplify the task). Your main tool for this task is MP4Box, and you will need to write a little bit of your own software to create the server media blocks. Figure 1 illustrates the overall flow of the different transformations that need to be applied.



**Figure 1:** Media preparation flow from \*.mp4 file to server media blocks.

Basically, every media file is split into a number of block files, each of which contains a fixed number of RTP packets. The length of the RTP packets can be either variable or fixed.

### 3) **Modify the FLIB Interface of Yima Server**

The final task is to modify the file input/output portion of the Yima server so that it can successfully read the media blocks created under Task 2. The code for the file I/O is located in the Server/FLIB directory. Your objective is to make sure that each the RTP packet in every block can be successfully read, scheduled and sent over the network. You may also need to modify the network delivery routines and the scheduler to feed the RTP packets to a client in a real-time manner.

The research paper includes existing RFC standards for MPEG-4 (especially, H.264 format) streaming (RFC 3016, RFC 3640, and RFC 3984), surveys other error-resilient methods, and discusses potential improvements.

## Reference Information and Software Version

All the students talking this module have an account on [cervino.ddns.comp.nus.edu.sg](http://cervino.ddns.comp.nus.edu.sg). The machine is running Red Hat Enterprise Edition Linux [RHEL 4.6](http://www.redhat.com/rhel/) (<http://www.redhat.com/rhel/>). You will need to use `ssh` to connect to the machine.

To analyze the RTSP protocol of QTP, you may test with Darwin Streaming Server (DSS), an open-source version of QuickTime Streaming Server. Additionally, several network sniffing tools are useful.

During the second stage, you are allowed to use any useful MPEG-4 parsing utilities such as MPEG4IP or MP4Box. However, the use of existing RTSP and RTP libraries except our package is strictly prohibited.

### Additional information and links:

- Yima PE documentation: get it from the course web site ([http://www.comp.nus.edu.sg/~cs5248/105/YimaPE\\_doc.pdf](http://www.comp.nus.edu.sg/~cs5248/105/YimaPE_doc.pdf))
- Yima PE source code tarball (YimaPE\_v1.1.tgz): get it from the [IVLE workbin](#)
- Slides: get them from the course web site (<http://www.comp.nus.edu.sg/~cs5248/105/05-project.ppt>)
- Darwin Streaming Server v5.5.5 (<http://dss.macosforge.org/post/previous-releases/>)
- MP4Box (<http://www.videohelp.com/tools/mp4box>)
- Network Sniffing Tools: [MS Network Monitor](#) (<http://blogs.technet.com/netmon/>), [WireShark](#) (<http://www.wireshark.org/download.html>)

## Submission Guidelines

### 1. Materials which you need to submit

- (1) Your modified Yima PE source code. Create a tarball of all your **sources** (please exclude sample videos, object files, etc.).
- (2) A detailed README.txt file, that should include the following information:
  - a) Your name(s), matriculation number(s), and your username(s) on the host cervino.
  - b) A brief description of each of the modified/new files and how it works.

Please create a tarball that collects all your files into a package with your matriculation number. Then compress it (with gzip) before you submit the file. **Make sure that your code compiles without ANY errors!**

- (3) A project report. Your write-up should be similar to a standard research paper (see details below). Please submit a PDF file.
- (4) Submit to your report and compressed package into the course module's [IVLE workbin](#) to the folder called 'Student Submission' by 23:59 on 05/04/10 (report draft) and 23:59 on 11/04/10 (source code).
- (5) Late submission policy for report.
  - late within 24 hours: 30% reduction in marks;
  - late within 3 days: 50% reduction in marks;
  - late within a week: 70% reduction in marks;
  - after one week: zero mark.Note that the project demonstrations will be held on 12 April and cannot be done later.

## Grading Policy (35 full marks)

1. **Project Demonstration (20 marks)**. You will present your project in class on 11 April 2010. You will have approximately 20 minutes to demo your project. The demo should include (1) the play of the assigned media file (**10 marks**), (2) on-demand media

preparation of a new MP4 media file and its play (**5 marks**), and (3) the rendering of the two videos in a SecondLife viewer (**5 marks**). The play includes PLAY and STOP (TEARDOWN) but excludes FastForward and Rewind. Since the second step may take longer time, you should run in an automated way. For the demo, make sure you are ready and everything works. We will not have time to debug any issues.

2. **Project Source Code (5 marks)**. I will have a look at your code. Criteria are for example, how well it is documented, how well it is understandable, how robust it is (i.e., can it process all relevant input files).
3. **Updated: Project Report (10 marks)**. Your write-up should be similar to a standard survey paper. Please include (at least) the following sections:
  - a) Abstract
  - b) Introduction and motivation for your project work (your specific design and implementation choices)
  - c) Description of your implementation, i.e., details on what you did, any difficulties encountered, any special features that you implemented, etc.
  - d) Results and discussion, i.e., what should we learn from this (and maybe possible improvements)
  - e) Conclusions
  - f) References

Most of the marks will be given for sections b), c) and d).

### Additional Notes

No plagiarism is tolerated. While you can talk to other teams and you are encouraged to use the IVLE Forum, you cannot copy source code and/or report text. If you are not sure whether something is permissible, either talk to the instructor or consult the university policy at <http://www.scholars.nus.edu.sg/uspinfo/academiccode.html>.