

## Assignment 4

### CS414, Multimedia Systems (Instructor: Klara Nahrstedt)

Posted: April 5, 2008

Deadline: April 30, 2008

#### Disclaimer

Details in the following assignment are subject to change. Although no additional requirements will be added, it is possible that some current requirements might be relaxed. Clarifications and additional hints, which will also be posted on the course website, might also be added to this document.

#### Introduction

This is the fourth assignment for your Internet Television (IPTV) project where you will experiment with the following: (1) synchronization between audio and video streamed from servers through proxy to clients, (see Figure 1), (2) graphical user interface to have VCR functions: PLAY, PAUSE, STOP, MENU, (3) channel switching, and (4) integrate MP3 assignments with the GUI, channel switching and synchronization.

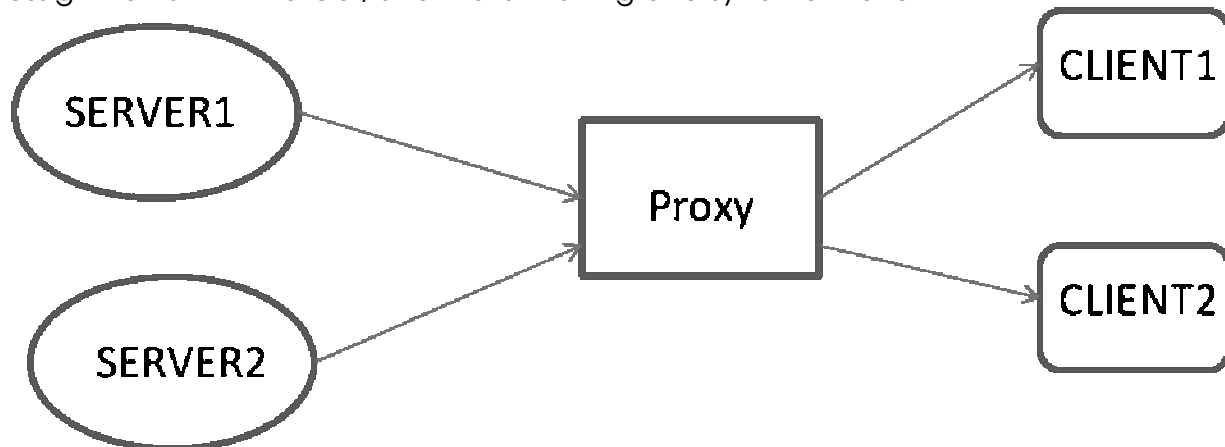
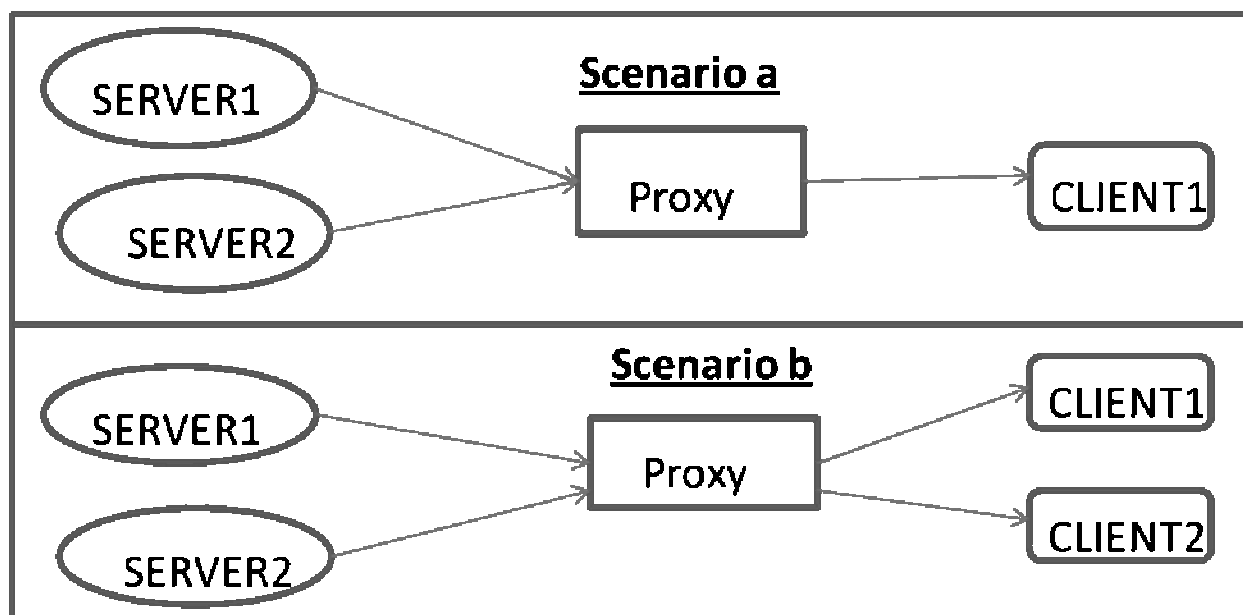


Figure 1: General distributed architecture for IPTV in MP4



**Figure 2:** Two different scenarios that will be tested in MP4

## Assignment Description

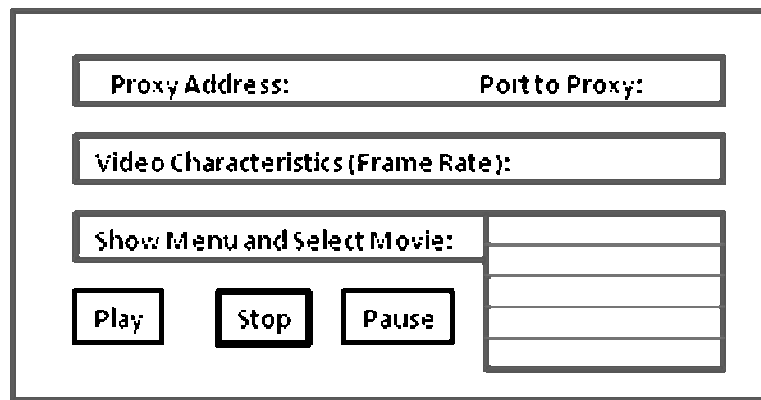
**Task 1:** To *synchronize audio and video* from the proxy to clients, it is important that each media unit (Session Data Unit) has timestamps. Transmission from the servers to the proxy does not need to have fine-granularity synchronization, although you must make sure that you cache the same time interval (e.g., 5 seconds) of audio and video, so timestamps are also useful when caching the data on the proxy. From the proxy to the clients, it is important that data units are time-stamped, especially for determining the synchronization points in the stream. *Synchronization points* are time points at which the client display thread will check both audio and video data units to see if they correspond to each other. We would suggest that you select a synchronization point every second. This means that at each one-second boundary you check for corresponding video and audio frames. For example, you check if your streams are synchronized at the end of the 1<sup>st</sup> second, at the end of 2<sup>nd</sup> second, 3<sup>rd</sup> second, etc. *YOUR AUDIO AND VIDEO FILES MUST CLEARLY DEMONSTRATE YOUR MEDIA SYNCHRONIZATION IMPLEMENTATION. IT MIGHT BE GOOD TO SELECT A VIDEO WITH A CLOSEUP OF A SPEAKER'S LIP MOVEMENTS AND THE CORRESPONDING AUDIO OF HIS OR HER VOICE.*

*Example:* If your video stream has a frame rate of 20 frames per second, and your audio stream has 80 samples per second, then at the end of 1<sup>st</sup> second, video frame number 20 and audio frame 80 should be played together, at the end of 2<sup>nd</sup> second, video frame number 40 and audio frame 160 should be played together, etc. If video is behind, it means at the end of 1<sup>st</sup> second you are only getting, for example, frame 15, then you display frame 15, but in the next round you should drop frames (at least 5-8 frames) to catch up at the end of the 2<sup>nd</sup> second. If the delays of video frames happen again at the end of 2<sup>nd</sup> second, then you may use your feedback channel from client

to proxy to ask the proxy to decrease the frame rate from 20 frames per second to 15 frames per second.

Note that audio is the more important medium among the two streams (i.e., audio is more important than video), hence the audio should be the *leading time synchronization stream*. It means that all other streams will be re-synchronized against audio. For example, if video is ahead of audio, video needs to slow down. If video is behind audio, video needs to speed up. Audio synchronization points will be the determining time point against which all other streams re-synchronize.

**Task 2:** In terms of *graphical user interface*, you can use Java or other languages to create a simple interface to control the streaming media. An example interface is shown in Figure 3. Keep in mind that Figure 3 is just an example that may or may not contain all of the necessary GUI components. You are free to choose whatever interface design that you find to be appropriate.



**Figure 3:** Example interface

Note that the *Menu feature* should allow the viewer to see all available movies on the servers. You should show all movies and allow the client to select a movie. The movie should be represented through two file names (video file and audio file corresponding to the selected movie), e.g., test.mjpg and test.au. If you click on both file names, both audio and video will be played, if you select on video file only, only video should play.

Note that the *Stop feature* should stop the play of the movie. If a viewer wants to play the movie, she clicks on 'Play' button and the movie starts to play from the beginning.

Note that the *Pause feature* should work as follows: when a viewer selects 'Pause' button, the movie (video and audio) should stop at the frame number x. If the viewer wants to continue to play this movie, she clicks on the 'Pause' button to continue to play the movie from the frame x.

**Task 3:** *Movie/Channel switching* should be executed through the *Menu feature*. It means that as you play movie1, you should allow the viewer to scroll the menu and

select another movie2. Once the viewer selects movie2, the IPTV system switches from movie1 to movie2.

**Task 4:** *Integrate MP3 establishment/streaming protocols with the synchronization approaches, so that the video plays or audio and video play synchronously in Scenarios (a), (b), shown in Figure 2. Also, all these protocols should be started from the graphical user interface (e.g., as shown in Figure 3).*

## Runtime Execution Commands:

1. Each Multimedia Server (sender side):

```
iptv SERVER port
```

After executing this command, the server will listen for control negotiation messages from the client to initiate an audio or video streaming session. *Note: you will need to start two servers on different machines during your demo.*

2. Multimedia Proxy

```
iptv PROXY port serveraddr1 sport1 serveraddr2 sport2
```

e.g., `iptv PROXY 9999 csil-core10 9998 csil-core11 9997`

The proxy will connect to 'serveraddr1' on 'sport1' and 'serveraddr2' on 'sport2'. After connecting to the servers, the proxy will listen to its local 'port' for connections from clients. *Note: the proxy will act as a client to the servers and a server to the clients.*

3. Multimedia Client (receiver side):

```
iptv CLIENT
```

Multimedia client will be called from a graphical interface (see Figure 3).

## Delivery

Each group delivers:

- Source code of your iptv program in the your group directory. The source code evaluation will be based on how well your code is documented. If you use some code that you found on the Internet (you must understand the code you found and include in your code, not just blindly copy the code !!!) or in a local system directories, then document it. It is very important that you give credit to people who developed the previous code. Your own code should include the following

information at the beginning of each file (Note: your code will be used next Fall by another class so it is important that you give yourself credit.)

- Each major source file should include
  - File Name: Name of the File
  - Description: Short description what the file includes (general description, what kind of functions are embedded in the file).
  - Version: version of your code. You start with version 0 and as you improve the code, at some point you increase the version.
  - Programmer's Name: your name(s) who developed the code
  - Company/University Name: you put the name of the course, department and university you implemented the code for;
  - Date
- Each function in your source code file should have a header with information:
  - Function Name: Name of the Function
  - Description: Short description what the function does.
  - Arguments: Specification of each input argument parameter entering the function and its description.
  - Results: Specification of returning parameters exiting the function and their description.
  - Comments: some special system issues connected with this function
- Group representatives come at the scheduled time (we will have a sign-up sheet) between 3 and 5pm on Wednesday, April 30th and show a demo of the required programs in 0216 Siebel Center.

## Evaluation of the Assignment (100 Points)

- **IMPLEMENTATION OF GRAPHICAL USER INTERFACE (15 points)** - the interface must have the PLAY, PAUSE, STOP buttons as well as the MENU capability to show and select movies. Demonstration 10 points, correct answers to questions 5 points.
- **VIDEO STREAMING WITH SCENARIOS (A) AND (B) IN FIGURE 2 AND GUI INTEGRATION (30 points):**
  - (5 points) – show STOP and restart feature
  - (5 points) – show PAUSE and continue play feature
  - (5 points) – PLAY video in scenario (a) of Figure 2
  - (5 points) – PLAY video in scenario (b) of Figure 2
  - (5 points) – show MENU and switch channels
  - (5 points) – respond to questions
- **AUDIO/VIDEO STREAMING WITH SYNCHRONIZATION AND GUI INTEGRATION (35 points)**
  - (5 points) - show STOP and restart movie
  - (5 points) – show PAUSE and continue play feature
  - (10 points) – PLAY audio/video in sync in scenario (a) of Figure 2 and switch channels
  - (10 points) – PLAY audio/video in sync in scenario (b) of Figure 2 and switch channels
  - (5 points) – respond to questions

- **DOCUMENTATION (20 points):** Each group should include a short PDF file (2-4 pages) that gives a brief overview of the components from your program followed by a description detailing how these software components implement the protocols and functionality specified for this assignment. Email this document to the TA [wconner@uiuc.edu](mailto:wconner@uiuc.edu) by midnight on April 30th, and also store it in your group directory. In particular, your documentation should describe your design and implementation of the following:
  - Graphical user interface
  - Synchronization of audio and video using timestamps
  - VCR-like functions (PLAY, PAUSE, and STOP) and MENU
  - Channel switching operations
  - Integration of all new requirements with existing MP3 functionality

The demonstration of the whole assignment for one group should take 30 minutes.