

Assignment 4

CS414, Multimedia Systems (Instructor: Klara Nahrstedt)

Posted: April 7, 2008

Deadline: May 1, 2008 (pre-competition on April 30, 2009)

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 P2P-VOD project where you will experiment with: (1) synchronization between audio and video streamed from servers to clients, (see Figure 1), (2) graphical user interface to have functions: search, insert, delete, play, fast forward, rewind, pause, stop, menu, and implement these VCR functions, i.e., integrate MP1, MP2 and MP3 assignments with the GUI and synchronization.

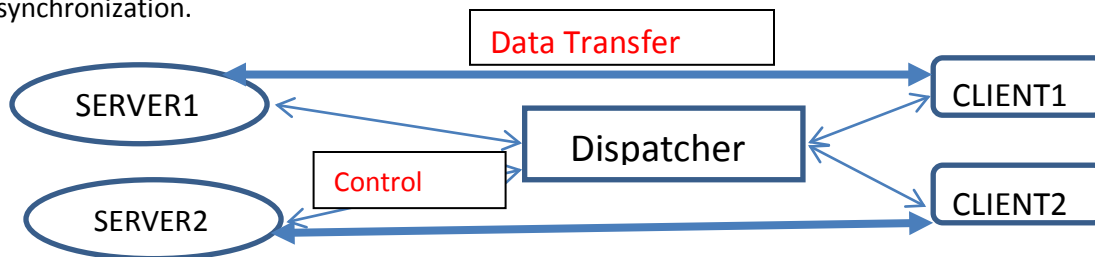


Figure 1: General Distributed Architecture for P2P-VOD in MP4

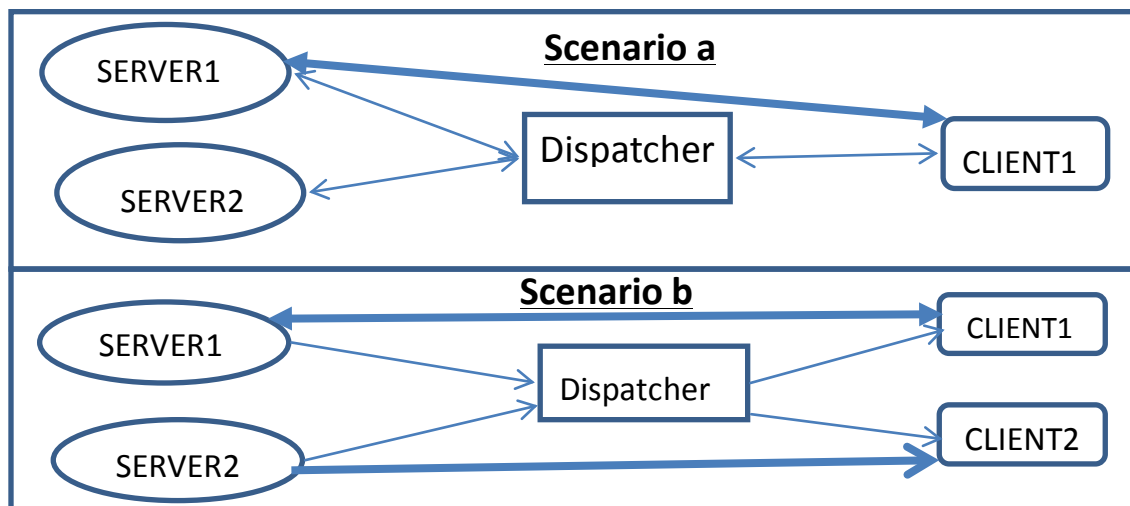


Figure 2: Two Different Scenarios that will be tested in MP4

Assignment Description

Task 1: To **synchronize audio and video** from a server to a client, it is important that each medium unit (Session Data Unit) has time stamps. Data units are time-stamped, especially that the streams decide on the synchronization points. **Synchronization points** are time points at which the client display thread will check both audio and video data unit if they are corresponding to each other. We would suggest that you select a synchronization point 1 second, it means that at each second boundary you check that you have video and audio corresponding frames. For example, you check if you synchronized at the end of 1st second, at the end of 2nd second, 3rd second, etc. It is important that you show synchronization of two different files, i.e., audio is stored in a different file, e.g., sound.wav file and video is stored in a different file, e.g., test.mjpg. It is not acceptable that you find a file that already has audio and video together interleaved in a single file. With this kind of file, synchronization is trivial!

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 1st second, video frame number 20 and audio frame 80 should be played together, at the end of 2nd second, video frame number 40 and audio frame 160 should be played together, etc. If video is behind, it means at the end of 1st 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 2nd second. If the delays of video frames happen again at the end of 2nd second, then you may use your feedback channel from Client to the Server to ask the Server 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 audio and 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 point will be the determining time point against which all other streams re-synchronize.

Server Address:		Port to Server:		
Frame Rate:	Duration:	Resolution:		
Show Menu and Select		Upload		
		Search		
		Delete		
		...		
FF	Play	Stop	Pause	Rewind

Figure 3: A Possible Interface (you have free hand to select/design an interface)

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 of an interface is shown in Figure 3.

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 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 sound.wav. 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. The GUI also needs to have a file management GUI to perform functionalities implemented in MP1, MP2, and MP3. That means, the GUI allows the user to **insert/search/delete files at the server(s), as well as VCR functions such as play, rewind, fast forward, pause and stop.**

Note that the **Stop feature** should stop the play of the movie and 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: Integrate MP1, MP2, 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). Make sure that

Runtime Execution Commands:

The servers and the dispatcher are executed the same way as in MP2.

1. Each Multimedia Server (sender side):

p2p-vod SERVER1 port
p2p-vod SERVER2 port

2. Dispatcher

Dispatcher server1 port1 server2 port2

3. Multimedia Client (receiver side):

Multimedia client will be called from a graphical interface (e.g., Figure 3).

Delivery

Each group delivers:

- Source code of your p2p-vod 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 representative(s) come at the scheduled time (we will have a sign-up sheet) between 5 and 7pm on **Thursday, April 30** and shows a demo of the required programs in 0216 Siebel Center. On Thursday will be a pre-competition where 4 demonstrations will be selected for the final competition on Friday 5-7pm in 216 SC. The final competition of the final 4 projects will be on Friday 5-7pm in front of judges.

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 CASES (A) AND (B) IN Figure 2 and GUI Integration (30 points):**
 - **(5 points)** – show fast forward
 - **(5 points)** – show rewind
 - **(5 points)** – play video in scenario (a) of Figure 2
 - **(5 points)** – play video in scenario (b) of Figure 2
 - **(5 points)** – menu functions – insert, delete, search files from the p2p-vod system
 - **(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
- **(10 points)** – play audio/video in sync in scenario (b) of Figure 2
- **(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 @uiuc.edu, and also store it in your group directory. In particular, your documentation should describe your design and implementation of the following: (a) synchronization protocol/approach, (b) integration of interface and the underlying streaming protocols.

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