

Virtual Media & Entertainment House

C. K. Yeo, B. S. Lee, V. Anand, Pathik Gupta, Arun Puri, Amol Dhabolkar

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue
S 639798
Republic of Singapore
Email: {asckyeo, ebslee}@ntu.edu.sg

Abstract

This paper focuses on the design and development of a framework that provides a one-stop location to cater to media and entertainment needs of users over the internet. The framework provides a virtual environment whereby users can come together to form a community. The framework provides various services to the users which include *Virtual Cinema Hall*, *Virtual Presentation*, *Virtual Whiteboard* and *Virtual Conferencing*. The framework has been implemented using Java [1] hence providing the support for heterogeneous environment. It uses Java Media Framework (JMF) version 2.1.1b [2]. The framework also uses high level protocols such as Java based RMI and Real Time Protocol (RTP) [3]. The framework has a full support for the Multicast network, hence making it highly scalable.

Keywords: broadband multimedia applications, video-conferencing, whiteboard, shared presentations

1. Introduction

Today's Internet that all of us use is a lonely experience. People seldom browse or watch movies on the computer together. This is a limitation of the computer itself, that it is a one-to-one experience. There is a need for a medium where people can do things together on the Internet, forming online communities. The proposed virtual media and entertainment house does exactly this. It serves the needs of users on broadband networks from all over the globe using multiple platforms. The system comprises the following features:

- A *Virtual Cinema Hall* where users can watch together media clips (movies, interviews, personal events etc) and provide for interaction between the users using chat and video conferencing applications.
- A *Virtual Presentation Application* where corporations (and individuals) can present their power point

presentations over their corporate networks (or web) to viewers irrespective of their location and platform.

- A *Virtual Videoconferencing Application* where people can get together irrespective of their platforms.
- A *Virtual Whiteboard Application* which is integrated with the videoconferencing application to simulate virtual classrooms and other virtual discussions.

The proposed framework uses Java for development as Java is a suitable candidate for creating a highly secure, robust and graphically oriented application for multi-platforms applications. JMF is used to provide the functionality needed to transfer large media files over the Internet by streaming. Streaming is a method whereby the media is transported over the network connection such that it can be retrieved by the user and played instantly. This is in contrast with *caching* whereby the entire media has to be downloaded to the user machine before playing it. The ability of JMF to provide media streaming, support for heterogeneous environment and support for various kinds of media formats makes it a very good choice for the development of such framework. The users book “rooms” which are based on the number of users using it and amount of time it is booked for. The other applications i.e. *Virtual Whiteboard* and *Virtual Presentation* also use Java technology. The framework also uses Java’s Remote Method Invocation

(RMI) so as to create a scaleable and extensible distributed system.

Section 2 of this paper covers the Virtual Cinema Hall. Section 3 describes the design and implementation of virtual presentations while sections 4 and 5 deal with virtual whiteboard and virtual conferencing respectively. Section 6 explores the commercial potential of the proposed system while section 7 concludes the paper.

2. Virtual Cinema Hall

Figure 1 shows the Client-Server Architecture of the *Virtual Cinema Hall* Application. The Client and the Server are Java-based systems that are built on top of Java Media Framework (JMF) and communicate with each other using two separate protocols - basic communication implemented using Remote Method Invocation (RMI) and the media streaming implemented using Real Time Streaming Protocol (RTP).

RMI is a high-level protocol used in modeling of classes and objects in a distributed (multiple JVM) computing environment. The RMI registry acts as a proxy between the remote objects, allowing them to invoke remote methods on each other [4]. JMF is a part of Sun’s Java Media and Communication APIs that are used to design an object oriented feature rich multimedia architecture that can handle a vast collection of multimedia processing [5]. The Server uses SQL Server as a database to perform housekeeping.

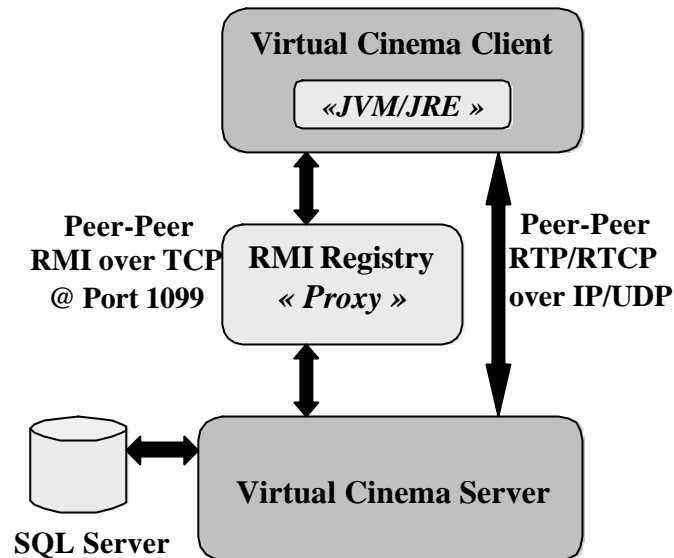


Figure 1: System Architecture Design

2.1 Overview

This architecture supports two kinds of streaming – *Server Side Streaming* and *Client Side Streaming*.

Server Side Streaming: The media files are stored on the server and are streamed to multiple clients using RTP/RTCP.

Client Side Streaming: The media files are stored on a client and is streamed to multiple clients using the servers high bandwidth network. The media file is streamed from the client to the server using RTP/RTCP and is further routed to multiple clients by the server, thus reducing the load on the client's bandwidth.

The features of the system are as follows:

- Exclusive booking of virtual *rooms* thus maintaining privacy within groups.
- Booking charges of rooms are based on number of guests invited and the time involved. (These two factors determine the bandwidth utilized over the network).

- Provides *Synchronization* between the streams of multiple clients, thus providing an effect of a cinema hall.
- A chat session is maintained among all the clients in a particular room, thus allowing them to converse while watching the media clip.
- The host *authorizes* each guest entering the room.
- The host *controls* (start, stop, pause, rewind) the media that is being played.
- System provides *Multicast* support, thus even further reducing the bandwidth constraints on the clients and the server.

2.2 Flow Analysis

Figure 2 shows the flow analysis of the application. The description is given as follows:

- a. The Host (*the user who books a room and initiates the media transfer*) logs on to the System. The login and password are sent over to the server (optionally over SSL) for authorization. This step uses RMI protocol for communication with the server.

- b. The Server authorizes the user and acknowledges the user as a host by sending him the details of his booking. These include the room number, the maximum number of guests that can be accommodated and the duration of the booking. The server also informs the host application of the *port number* on which the media is to be received. In this step the server again uses RMI to invoke a method on the client and returns all the data pertaining to the booking using a Java *serializable* object.
- a. c. The two-way arrow signifies the two streaming scenarios:
- *Server Side Streaming*: The host specifies the file on the server, which is to be streamed and the server starts transmitting the file using RTP/RTCP on the port specified in step b.
 - *Client Side Streaming*: The host specifies the file on the local system
- d. A *guest* logs on by specifying the username and password as in steps a and b. Likewise, RMI protocol is used for this step.
- e. Here, the guest is provided with the rooms that are currently being used by some group. The server also provides the guest application the *port numbers* of all the valid rooms on which the streams are to be received. The server again sends a serializable object with these details.
- f. The guest identifies and chooses a room to enter and sends an *authorization request* to the host of that particular room via the server.

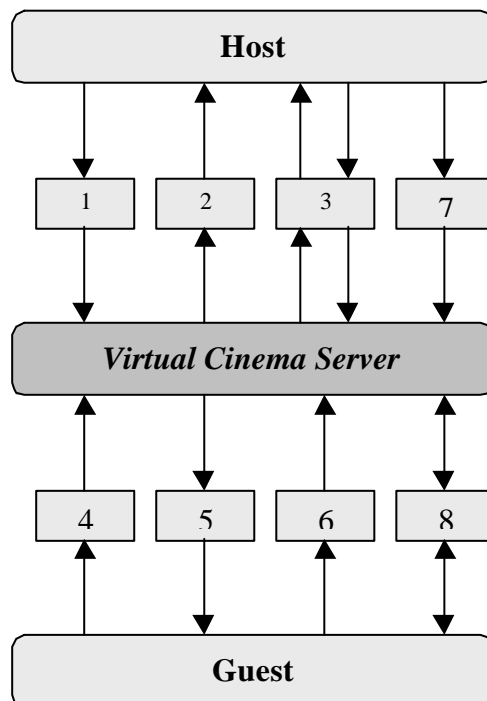


Figure 2: Flow graph of the Virtual Cinema Hall

- g. g. The host may or may not authorize the guest to enter the room. In the case the permission is denied, the guest can book a room or try another room.
- h. h. In the case the guest has been authorized, the guest application will start receiving the streams. In case of server side streaming, the server directly uses the same RTP session that is being used to serve the host for the guest. For client side streaming, the server receives the RTP stream from the host and then routes the stream to the guest via the port specified in step f.

Integrating with a video conferencing application can further extend this application. Using JMF, live audio & video on all client sides can be captured and streamed to the other participants in the same room. This is done by starting another client side stream for each client and routing the appropriate streams to the participants in the room. This is similar to the *virtual*

conferencing application explained in the next section.

3. Virtual Presentations

Figure 3 shows the architecture of the presentation module. The Presentation host starts the presentation application before starting the presentation. The application will convert the presentation slide into a jpeg format and transmit it to the server over a socket connection. The server in turn streams these images to the guests of the presentation. The guest application will convert the images into Java images and display over an applet interface.

The features of the application are:

- Fully *customizable* for corporate networks with the dedicated presentation server to improve performance.
- Highly economical business solution that saves travel time and expenditure.
- Easily integrable with *video conferencing* application.

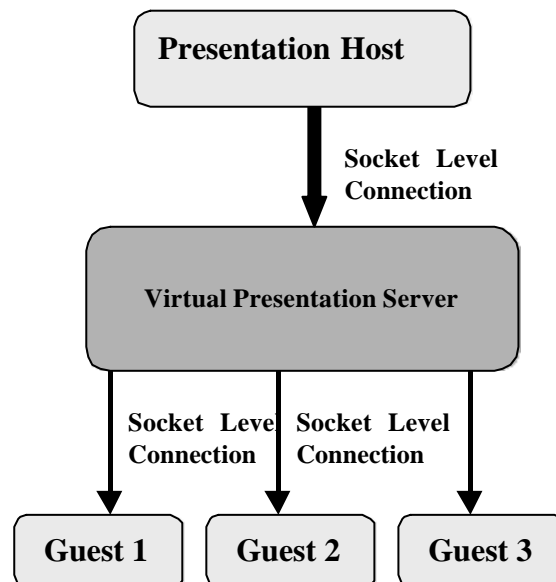


Figure 3: Presentation application Architecture

Sharing of Powerpoint presentations over a heterogeneous environment is accomplished through the following stages:

- a. The host, who is the user who books a room and initiates the media transfer, logs on to the system. The login and password are sent over to the server (optionally over SSL) for authorization. This step uses RMI protocol for communication with the server.
- b. The server authorizes the user and acknowledges the user as a host by sending him the details of his booking. These include the room number, the maximum number of guests that can be accommodated and the duration of the booking. In case the user has not made the booking, he/she has the option of booking a room.
- c. The client application connects with the presentation server using a TCP/IP based socket level connection.
- d. The client application converts the presentation slides to the jpeg images in real time and streams it over to the server using the established socket connection.

- e. The server then identifies the guest in the particular room and converts these jpeg images to java images. These Java images are sent over to the guest application.
- f. The guest application then uses the applet interface to display these Java images.

4. Virtual Whiteboard

Figure 4 shows the architecture of the virtual whiteboard. The *WhiteBoard* application is a value-added application to the *Virtual Media/Entertainment House*. The whiteboard is built on a chat application and users can draw objects, flow charts, diagrams, free hand draws etc for other users to see and discuss.

The system has been implemented using Java network programming by making socket connections between the server and the clients connected via an Applet interface. The *DListServer* maintains a list of the active connections in an *IMList*. Any action performed by any of the clients invokes an *updateEvent* on the client which identifies

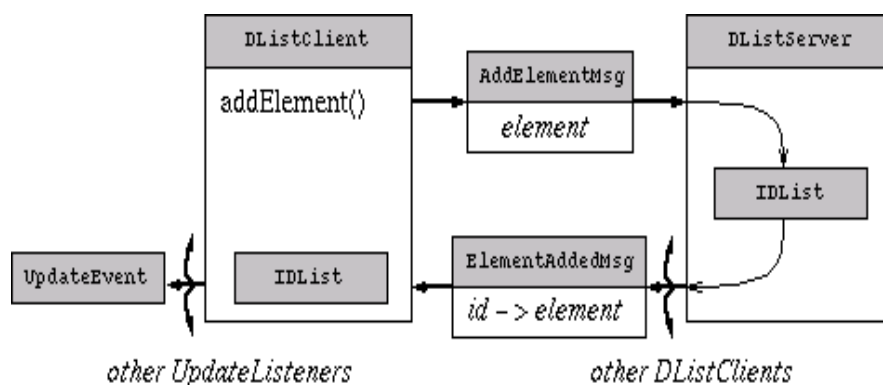


Figure 4: The Virtual Whiteboard Architecture

the action performed and sends an *addElementString* event notifying the other clients to update their whiteboards.

The whiteboard would optionally include a video conferencing application where the users can videoconference with each other while using the whiteboard.

5. Virtual Conferencing

The conferencing application allows all the participants in a room to view each other and at the same time chat with each other. The application uses RMI to communicate between the clients and the server and the video capture is streamed using RTP/RTCP protocol.

When the client initiates the streaming, the application detects a video capture device and starts the *Client side streaming* as explained in the previous section. The server handles all the details of routing the appropriate streams to the participants in the room.

To begin with, the guest/host decides to initiate the *virtual conferencing* application. A request is then sent to the server specifying the port number on which the client performs the live media streaming. The server obtains the request and starts to listen on the specified port for the stream. Once the server obtains the client stream, it obtains the information about the other participants in the same room. It then starts streaming the client stream over a certain port. The server also notifies each of the other participants about the new stream that is available for receiving on the particular port. The other participants will initiate the receiver session on their local machines. At the same time, server also informs the client sending the new stream about the port numbers of the other participants in the room who are sending the live streams. This will invoke the receiver session on the client for each of the other participant in the room allowing the client to view all the other participants. These steps are performed for all the clients who enter the room and initiate the live video streaming.

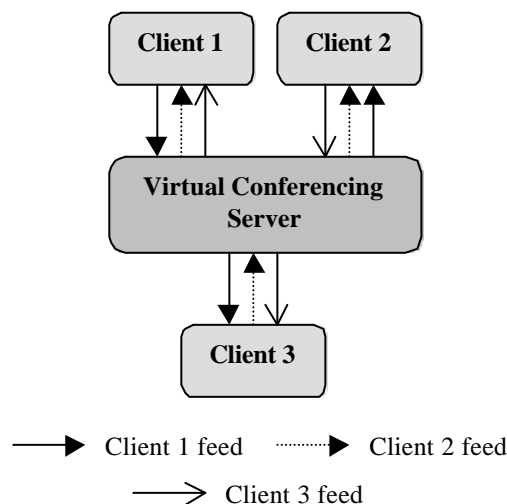


Figure 5: Video Conferencing Stream Flow

As can be seen from the steps, this application performs the client side streaming (refer to Section 2.2) but on very large scale whereby the server has the responsibility of handling the various streams. The *virtual conferencing* application can work both as a stand-alone application and can also be integrated with the existing *virtual cinema hall* system. It can also be integrated with the *virtual whiteboard* and the *virtual presentation* applications. This provides high scalability and extensibility to this application.

6. Commercial Potential

The commercial potential of this system can be divided into 2 categories:

Potential from Individual & Home Users

- The *Virtual Cinema Hall* is an application that brings together relatives all over the world to have some fun together for a small fee.
- From data over telephone lines to telephony over data lines - technology has come a long way and is moving towards extensive use of video conferencing (including video conferencing in 3G phones) in areas such as Customer Relationship Management (CRM), conferencing with near and dear ones who are away.
- In today's busy world, people get less and less time to attend religious gatherings. Not all people make it for family functions and ceremonies. These events can be covered on a video camera and directly streamed in real time to let the ones away participate in the event remotely.

Potential from corporations and distance learning centers

- The *Virtual Presentation* application can help large and small corporations, save travel expenditure, which is one of the highest expenditures for any corporation with international market.
- The presentation application is fully customizable for MNCs and other corporations that may want to have a *Dedicated Presentation Server* behind their firewall. This would help to keep offices in different regions connected more effectively and economically.
- The presentation application is easily integrated with the video conferencing application to provide interactive presentations (appropriate when dealing with remote customers).
- Using the video conferencing and the presentation applications would also benefit Distance Learning Centers.
- Security cameras can be easily integrated with the system and the application can be used to monitor areas using the security cameras remotely.

7. Conclusion

The *Virtual Media & Entertainment House* provides a complete media solution for the home as well as business users. The system is designed to provide a community based service by allowing the users to interact with each other using virtual conferencing and virtual whiteboard. The system also provides a complete media solution by allowing the user to play files from the local machine over the Internet or

from the server itself. The system reduces the client side bandwidth requirement by routing all the streams via the server to the other participants. The system also supports the multicast media transmission and reception. The clients need only to specify the multicast address that they want to join and the server can stream the media accordingly. This kind of arrangement provides a highly scalable solution as multicast is one of the upcoming technologies in the Internet world.

References

- [1] Sun's Java Website, The Source for Java™ Technology, <http://www.javasoft.com> (<http://java.sun.com>)
- [2] Java Media Framework, <http://java.sun.com/products/java-media/jmf/index.html>
- [3] IETF RFC 1889, RTP: A Transport Protocol for Real Time Applications. Current revision, <http://www.ietf.org.internet-drafts/draft-ietf-avt-rtp-new-04.txt>
- [4] Java Remote Method Invocation by Suresh Raj Gopalan, http://www.execpc.com/~gopalan/java/java_rmi.html
- [5] Linden deCarmo, Core Java Media Framework, 1999 Prentice-Hall.