AN EMBEDDED P2P-BASED POSITIONAL AUDIO SYSTEM IN VIRTUAL ENVIRONMENTS

Beomjoo Seo, Roger Zimmermann, Min Min Htoon[*]

Department of Computer Science
NUS, Singapore 117417
Email: [seobj, rogerz, htoonmm]@comp.nus.edu.sg

Chung-Dau Wang

Audiary Pte Ltd.,
NUS, Singapore 119613
Email: chungdaw@audiary.com

ABSTRACT

Networked virtual environments are increasingly used for collaboration tasks and other interactive applications. While the graphics in such virtual worlds are usually three-dimensional, interactive 3D voice support is still in its infancy. Here we describe our demonstration system that supports P2P-based positional audio and interactive voice communication with the SecondLife platform.

***Keywords*—** Peer-to-Peer, positional audio, overlay topology, VoIP, XMPP, virtual environments

1. INTRODUCTION

Positional live audio, which refers to the rendering of voiced sounds in a three-dimensional space, has emerged as an intriguing and desirable service in many networked virtual world applications such as online multi-player games.

One of the existing techniques for implementing such audio spatialization is based on the client/server model, as exemplified by the engine of vendor *Vivox*. In this approach, a server collects positional statistics of the avatars (i.e., connected clients), then constructs spatialized audio bitstreams and delivers them individually for each player over a voice channel. Although available commercially, this model requires ample server resources such as bandwidth, making it less attractive for small- or medium-sized game publishers. Moreover, our recent end-to-end delay measurements have revealed that the Vivox server may introduce a significant processing delay (e.g., more than 530 milliseconds, even excluding the transmission delay), hence dominating the total end-to-end latency.

To achieve high interactivity while providing 3D voiced conversations, we propose a different approach that is based on a self-scaling peer-to-peer (P2P) overlay network model and hence leverages the resources of the clients. Our project *Passer* is a flexible and scalable communication framework for networked positional audio applications which allows developers to distribute the necessary and relevant media streams via a unified communication paradigm. Our spatialized audio framework supports multiple sources and sinks, and the audio streams can originate either from capture devices or from neighboring audio nodes. In this report we describe the design choices we made and the implementation details of the overall architecture, while our demonstration shows the system in action.

2. RELATED WORK

Traditional client/server-based positional audio systems like *Vivox* and *Mumble* provide users who possess very limited resources with positional audio, at the expense of server resources such as processor cycles and network bandwidth. To reduce the server load, Boustead *et al.* [1] proposed a discrimination of a user's auditory scene by dividing it into two types: an interactive and a background area. In their client/server model, the voiced sounds from neighbors in the interactive zone (typically closer to the user) are delivered via an audio server separately and then rendered locally. The remaining nodes in the user's hearing range are placed in the background zone and combined in one of the available groups that has a minimal spatial cue error (i.e., angular clustering). The server mixes the voices of the nodes in each group and delivers them to the users. This solution achieves the synchronization of audio and movement events which the traditional model cannot maintain, but it still suffers from a server bottleneck.

Bharambe *et al.* [2] proposed the use of a small set-of-interest group that stores neighboring players whom a game player is paying attention to, instead of maintaining a list of all neighbors within the player's area of interest. To recognize the player's interest correctly, the authors proposed a simple estimation method, in which every player maintains an individual attention value per neighbor. Since players are likely to pay more attention to a closer neighbor in the vicinity, the method proposed three heuristic metrics: proximity, orientation-based aim, and interaction recency. We also use these metrics when estimating a user's auditory attention, assuming that auditory scene is affected by visual cues.

3. SYSTEM OVERVIEW

Our peer-to-peer design consists of four major components (illustrated in Fig. 1): a P2P connection engine, a game client plug-in, an overlay topology manager, and a 3D audio mixer. The P2P connection engine establishes a direct or indirect con-

[*] This research has been funded in part by A*Star SERC grant no. 082 101 0028. We also acknowledge the support of the NUS Interactive and Digital Media Institute (IDMI).

978-1-4244-7493-6/10/$26.00 ©2010 IEEE

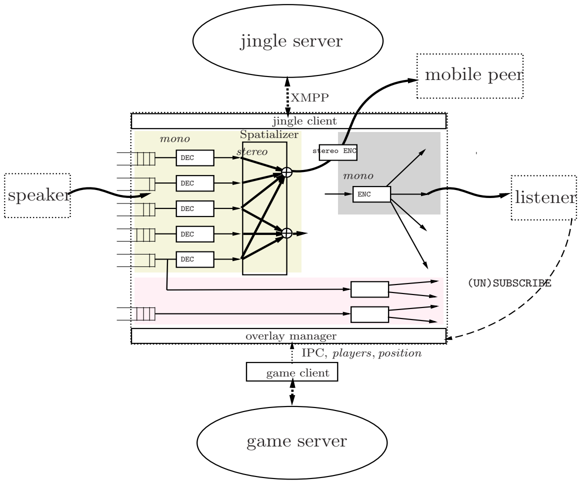
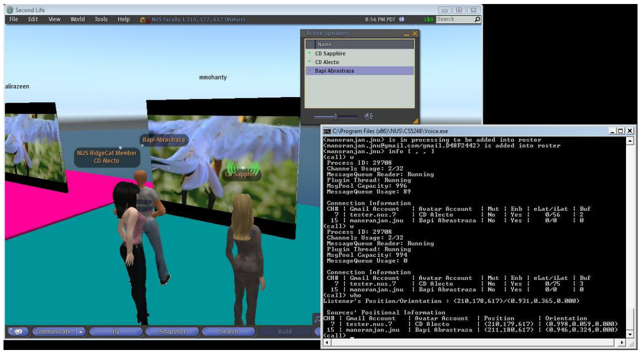


Fig. 1. The system diagram of the positional audio demonstration system consisting of four components: P2P engine, game plug-in, overlay manager, and 3D audio spatializer.



Active Speakers

Name
CD Sapphire
CD Alecto
Bapi Abrastraza

C:\Program Files (x86)\NUS\CS5248\Voice.exe

```
<manoranjan.jnu> is in processing to be added into roster
<manoranjan.jnu@gmail.com/gmail.D48F2442> is added into roster
<manoranjan.jnu> info [ , , ]
<call> w
 Process ID: 29708
 Channels Usage: 2/32
 MessageQueue Reader: Running
 Plugin Thread: Running
 MsgPool Capacity: 996
 MessageQueue Usage: 89

 Connection Information
 CH# | Gmail Account   | Avatar Account  | Mut | Enb | eLat/iLat | Buf
   7 | tester.nus.7    | CD Alecto       | No  | Yes |   0/56    | 2
  15 | manoranjan.jnu  | Bapi Abrastraza | No  | Yes |   0/0     | 0
<call> w
 Process ID: 29708
 Channels Usage: 2/32
 MessageQueue Reader: Running
 Plugin Thread: Running
 MsgPool Capacity: 994
 MessageQueue Usage: 0

 Connection Information
 CH# | Gmail Account   | Avatar Account  | Mut | Enb | eLat/iLat | Buf
   7 | tester.nus.7    | CD Alecto       | No  | Yes |   0/75    | 3
  15 | manoranjan.jnu  | Bapi Abrastraza | No  | Yes |   0/0     | 0
<call> who
Listener's Position/Orientation : <210,178,617>/<0.931,0.365,0.000>

 Sources' Positional Information
CH# | Gmail Account   | Avatar Account  | Position      | Orientation
  7 | tester.nus.7    | CD Alecto       | <210,179,617> | <0.998,0.059,0.000>
 15 | manoranjan.jnu  | Bapi Abrastraza | <211,180,617> | <0.946,0.324,0.000>
<call> _
```

Fig. 2. A screenshot of the 3D audio engine embedded in a Second Life viewer. To additionally aid in the spatial sensing of a voice, we enabled the lip-sync and visual cue features for voice activity in the viewer.

nection to remote nodes. Since many of those nodes are behind NAT routers (network address translation), NAT traversal methods must be employed to ensure connectivity. The game client engine receives all players' information and their movements within a client's proximity range (*i.e.*, the area of interest). To regulate the outgoing network traffic, the overlay topology manager determines the audibly closest players in an ad-hoc manner and opens transmission channels with them to receive media packets. The last component, the 3D audio mixer, combines incoming mono streams and recreates spatialized stereo sound for playback.

To connect peers directly and reliably, we used the open source library *libjingle*. Libjingle is an implementation of Jingle, an extension of XMPP (Extensible Messaging and Presence Protocol) that embeds the delivery of media channels. The message exchange among peers is handled via an XMPP server reliably, while a UDP-based P2P transport channel can also be established with this mechanism.

For the efficient delivery of audio signals across different modules, we have developed our proprietary filter chaining mechanism called *Passer*, which provides low-level functionalities for audio processing. A filter, in Passer's context, represents one stage in the processing of media data and can be connected to other filters. Passer defines five types of filters named as follows: *capture*, *render*, *audio codec*, *binaural localization*, and *relay*. The capture filter acquires audio samples from sound devices. The render filter plays uncompressed mono or stereo audio samples. The audio codec, running either a (stereo) encoder or decoder, is an intermediate filter that encodes uncompressed sound samples or decodes encoded bitstreams. The binaural localization filter receives uncompressed audio samples from multiple sources and localizes them according to their associated positions. Finally, the relay filter receives compressed samples and delivers them to designated peers.

To collect the avatar positions in 3D space, we embedded a *plug-in module* in a Second Life viewer application. It filters the positions efficiently within a user's audible range and delivers them to the P2P engine.

The overlay manager constructs an ad-hoc mesh topology, where every peer maintains its visible neighboring nodes and connects to a fixed number of audibly related peers within hearing range. To minimally utilize the upload bandwidth, it continues to attempt to find connections whose audio packets are spatially mixed, while preserving the spatial cues of the original sources.

The audio mixer is the core feature of our system. It supports not only multiple heterogeneous audio sources but also multiple sinks, which allows resource-limited clients, such as a cellular-phone user, to still be served by a proxy node which collects incoming streams and delivers an appropriate stream mix long a stereo channel. Among available audio codecs, we chose the Speex codec because of its light weight, royalty-freeness, and high quality. Using Speex, we enabled features such as variable bit rate, voice activity detection, discontinuous transmission, echo cancellation, noise suppression, automatic gain control, and adaptive jitter management.

4. CONCLUSIONS

We have implemented a demonstration system to support positional audio and interactive voice communication (3D audio) in existing, networked virtual world environments. Our prototype system demonstrates the feasibility and effectiveness of our approach with multiple Second Life clients.

5. REFERENCES

[1] P. Boustead, F. Safaei, and M. Dowlatshahi, "DICE: Internet Delivery of Immersive Voice Communication for Crowded Virtual Spaces," in *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, March 2005, pp. 35–41.

[2] Ashwin Bharambe, John Douceur, Jacob R. Lorch, Thomas Moscibroda, Jeffrey Pang, Srinivasan Seshan, and Xinyu Zhuang, "Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games," in *SIGCOMM '08*, Aug. 2008.