RESEARCH ONLINE

University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information Sciences

2004

A distributed server architecture for providing immersive audio communication to massively multiplayer online games

C. D. Nguyen
University of Wollongong

F. Safaei
University of Wollongong, farzad@uow.edu.au

P. Boustead
University of Wollongong, boustead@uow.edu.au

Publication Details

This article was originally published as: Nguyen, CD, Safaei F & Boustead, P, A distributed server architecture for providing immersive audio communication to massively multiplayer online games, Proceedings 12th IEEE International Conference on Networks (ICON 2004), 16-19 November 2004, vol 1, 170-176. Copyright IEEE 2004.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

RESEARCH ONLINE

A distributed server architecture for providing immersive audio communication to massively multiplayer online games

Abstract

This paper investigates the assignment of audio mixing operations to a geographically distributed set of servers to provide an immersive audio communication environment for massively multiplayer online games. The immersive audio communication service enables each avatar to hear a realistic audio mix of the conversations in its audible range. There are three primary delivery architectures for this service, namely, peer-to-peer, central server and distributed servers. We focus on the distributed server architecture, which partitions the virtual world into regions or locales and then assigns the computation associated with the creation of audio scenes for all avatars in each locale to a server. Our aim is to find the optimal way to partition the virtual world into locales and then choose the locale servers in such a way that reduces the total delay perceived by all avatars. We have produced a mathematical formulation for the optimal partitioning and server assignment and developed a heuristics approach based on a graph algorithm. We have developed a simulation environment that creates both the physical world (geographic distribution of participants and the Internet topology model) and the virtual world (distribution of avatars based on different avatar aggregation behaviors). We have solved the problem exactly as well as using the heuristics algorithm for a range of simulated virtual and physical worlds. In many cases, the heuristics results were within 5% of the optimal. Our algorithms and simulation study will be of benefit to future immersive audio communication service providers in the design of a cost effective delivery architecture for this service.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Nguyen, CD, Safaei F & Boustead, P, A distributed server architecture for providing immersive audio communication to massively multiplayer online games, Proceedings 12th IEEE International Conference on Networks (ICON 2004), 16-19 November 2004, vol 1, 170-176. Copyright IEEE 2004.

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/184

A DISTRIBUTED SERVER ARCHITECTURE FOR PROVIDING IMMERSIVE AUDIO COMMUNICATION TO MASSIVELY MULTI-PLAYER ONLINE GAMES

Cong Duc Nguyen, Farzad Safaei, Paul Boustead
Telecommunications and Information Technology Research Institute
University of Wollongong, Australia
Email {cong, farzad, paul}@titr.uow.edu.au

Abstract—**This paper investigates the assignment of audio mixing operations to a geographically distributed set of servers to provide an *immersive audio communication* environment for massively multi-player online games. The immersive audio communication service enables each avatar to hear a realistic audio mix of the conversations in its audible range. There are three primary delivery architectures for this service, namely, peer-to-peer, central server and distributed servers. We focus on the distributed server architecture, which partitions the virtual world into regions or locales and then assigns the computation associated with the creation of audio scenes for all avatars in each locale to a server. Our aim is to find the optimal way to partition the virtual world into locales and then choose the locale servers in such a way that reduces the total delay perceived by all avatars.**

We have produced a mathematical formulation for the optimal partitioning and server assignment and developed a heuristics approach based on a graph algorithm. We have developed a simulation environment that creates both the physical world (geographic distribution of participants and the Internet topology model) and the virtual world (distribution of avatars based on different avatar aggregation behaviors). We have solved the problem exactly as well as using the heuristics algorithm for a range of simulated virtual and physical worlds. In many cases, the heuristics results were within 5% of the optimal. Our algorithms and simulation study will be of benefit to future immersive audio communication service providers in the design of a cost effective delivery architecture for this service.

I. Introduction

Massively multi-player online games (MMOG) have become attractive applications over the Internet. A recent report [1] shows that the revenue for MMOG is estimated at $635 million in 2003, and by 2005 should reach $1.8 billion. In current MMOG, although the 3D graphics have been improved significantly, communications between game participants are limited and often based on text messages.

We believe that these games will be far more attractive if *immersive* audio communication can be provided to game players. The immersive audio communication service allows all the avatars to hear a realistic mix of voices of other avatars in their area of interest or hearing range. This audio scene is a personalised mix of voices of other avatars, spatially placed and attenuated according to distance to the listener. Although accuracy is desired, there is variation among the avatars about the importance of different audio signals. Let us define the *interactive zone* as the immediate vicinity of the avatar where active communicative interaction may take place, while the *background zone* as the region outside the interactive zone stretched to the limits of hearing range.

In some cases, the user may only be interested in the interactive zone. The background audio scene, therefore, appears as 'noise' and can either be blocked or simulated by a background chatter of suitable volume. However, there are situations where the multi-person voice communication is either the primary purpose of gathering in the virtual space or at least a very important means for achieving the actual goal. This will be particulary important for the genres of social games and interactive entertainment. Natural multi-person communication is often characterised by the presence of multiple simultaneous conversations among the people gathered in an environment such as a cocktail party, cafe, foyer of a conference, or market place. The ability to pick up interesting conversations in one's vicinity and join these groups or simply be aware of the peripheral discussions is critical to our sense of satisfaction of being in the presence of a crowd. The current style of audio teleconferencing, when a strict protocol of 'one at a time' for conversation has to be followed, may be too restrictive for crowded virtual spaces. We, therefore, believe that in certain situations, realistic presentation of the background audio zone is desirable, if not critical. Nevertheless, it is reasonable to assume that the accuracy requirements (in terms of delay and spatial placement) are more stringent for the interactive zone compared to the background zone.

We assume that a game service provider can have access to a number of servers located over the Internet. One way to distribute computation is to partition the virtual world into *locales*, and the audio mixing for all the avatars in each locale is performed by one of these servers. In the system design developed by our research team, the server compute a fixed number of partially mixed audio streams for each avatar in that locale and send these streams back to that avatar. We refer this architecture to as *distributed locale servers*. Our aim is to find the optimal way to partition the virtual world into locales and then choose the locale servers in such a way that reduces the total delay perceived by all avatars. We have produced a mathematical formulation for the optimal partitioning and server assignment and developed a heuristics approach based on a graph algorithms. A simulation study is carried out to evaluate our proposed server assignment algorithms and to discuss various factors that might improve the group audio communication delay. In particular, we investigate the effect of changes in the number of servers and the correlation between distribution of avatars and game participants on communication delays and network resource usages in different game scenarios.

The rest of this paper is organized as follows: In Section II, we briefly describe primary delivery architectures for this service and the model of a physical world and a virtual world. In Section III, we propose server assignment algorithms for the distributed locale server architecture. Simulation results are presented in Section IV. We discuss related work in Section V and draw con-

0-7803-8783-X/04/$20.00 © 2004 IEEE

clusion in Section VI.

II. Service models

A. Delivery architectures

There are three primary delivery architectures for the creation of immersive audio communication, namely, peer-to-peer, central server and distributed servers. The central server architecture (shown in Fig. 1a) has the ease of implementation for functionality, such as, security, privacy and billing. The limitations of central server architecture are high delay, processing scalability and single point of failure. The peer-to-peer architecture (shown in Fig. 1c) has low delay but is subject to high bandwidth usage. As a result, this architecture has serious access bandwidth problems, especially on upstream paths. An additional disadvantage of the peer-to-peer architecture is that security and anonymity is reduced by the fact that this method of delivery will provide a game player's IP address to all people in his/her hearing range. The central server and peer-to-peer architectures have been evaluated in [9].

Our focus is on a *distributed* server architecture. We have identified two major drivers for the distribution of applications over a network of servers: *resource driven distribution* (RDD), and *latency driven distribution* (LDD). The aim of RDD is to move components of the application to other servers to overcome the processing capacity limitations in one site. If there are sufficient resources available in one processor, however, the distribution would not lead to any performance improvement. For LDD applications, the ability to control the *spatial* location of the processing is of pivotal importance. So even if the processing resource is abundant in one location, the application will perform poorly in terms of response time and latency if it is not distributed over a suitable set of geographically diverse servers. The immersive audio communication is an LDD application and we will demonstrate that the distributed locale server architecture will lead to significant improvements even when there are abundance of processing resources in each server.

Fig. 1b shows the basic operation of the distributed locale server architecture. These servers form an overlay network, connected in full-mesh. The virtual world is partitioned into locales, and the audio mixing operations in each locale is performed by one server. It is possible to assign one server to more than one locale. Avatars in each locale only need to send their audio streams to the server assigned to that locale. If avatars in two adjacent locales, which are assigned to two different servers, are in the hearing range of each other, these two servers need to exchange the required audio streams (shown as broken lines in Fig. 1b). When servers are located in different parts of the Internet, distributed locale servers can provide better delay performances compared with the central server, especially when people from a particular geographic region aggregate as a group in the virtual world. Since computations are distributed, this architecture is more scalable than the central server and has no single point of failure. Finally, similar to the central server architecture, implementation of security, privacy and billing is also possible.

B. Model of a physical network and a virtual world

We model the network and server topology as a graph $G_p(V_p, E_p)$; where V_p denotes a set of nodes, E_p denotes a set of edges; a set $S \subseteq V_p$ denotes a set of potential processing server; $R \subseteq V_p$ is a set of Internet Service Provider Points of Presence (ISP POPs). It is noted that R and S are disjoint sets and all nodes can support routing functions. Each link has two metrics: a link cost for policy-based shortest path routing and a link delay representing the propagation delay between the two nodes. The number of game players located at ISP POPs are randomly generated based on a uniform distribution. We do not consider the delays from ISP POPs to game players since these delays are fixed and can not be influenced by us. Shortest path first (SPF) routing is implemented. This is the current routing protocol in the Internet.

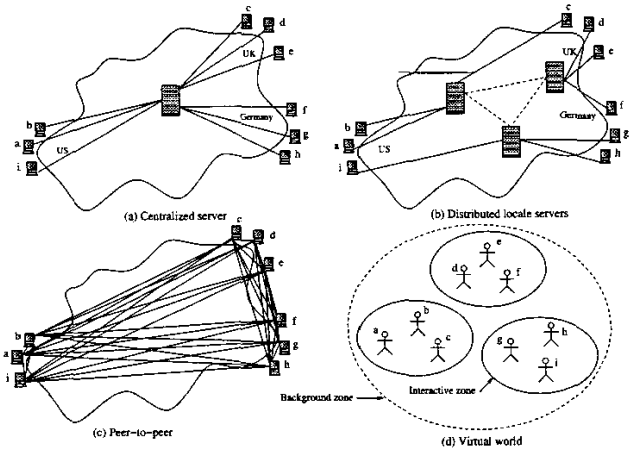


Fig. 1. Server architectures for immersive audio scene creation

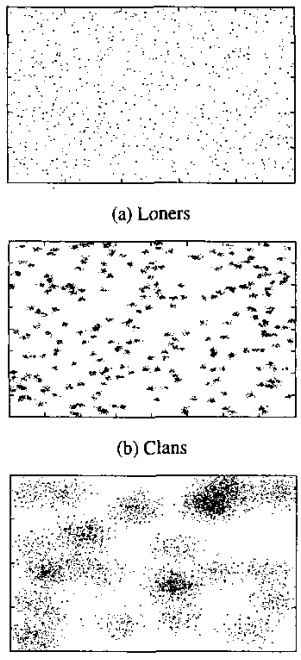
The virtual world is modelled as a square area of certain size, in which, avatars are distributed according to the following distributions:

- Uniform distribution: avatar (x,y) coordinates are set according to uniform random distribution. This results in a uniform spread of avatars in the virtual world.
- Clustered distribution: Cluster centers are randomly placed in the virtual world. At each cluster center, avatars are positioned around the center according to a normal distribution with the mean of zero.

For different types of games, we envisage that there are a range of avatar grouping patterns, referred to as *game grouping behaviors*. The following terminology is proposed to capture this range of game grouping behaviors:

- *Loners* represent isolated avatars which are relatively far away from each other and have low chance of interaction.
- A *clan* represents a medium size group, such as a hunting group, which is very common in online games.
- A *crowd* represents a very large group, such as a stadium or a market place.

The above avatar distributions are demonstrated in Fig. 2. In Fig. 2a, we model the virtual world of games which mainly have loners as a uniform avatar distribution with low density. Clans are modelled as small clusters, which consist of up to about 30 avatars. Crowds are larger clusters, which consist of about 100 avatars or more. Fig. 2b and Fig. 2c shows a virtual world which consists of 250 clans and 25 crowds, respectively.



(a) Loners

(b) Clans

(c) Crowds

Fig. 2. Avatar distribution in different games

There may be a tendency for game players from close geographic regions to aggregate together due to language, culture and lifestyle preferences. As an example, people in Japan may prefer to gather together in some games for language and cultural reasons. A parameter loosely referred to as "correlation" parameter is introduced to specify how people aggregate in each clan/crowd based on their real world geographic locations. In particular, the correlation parameter states the probability that people in each crowd/clan reside in a particular geographic region. This region is modelled as an ISP POP or a number of close POPs.

To model this correlation, we use a correlation parameter x, $0 \leq x \leq 1$. Each time an avatar is populated in a clan/crowd, a random number r between 0 and 1 is generated. If $r > x$, the avatar is from a random chosen POP. If $r \leq x$, the avatar is from the POPs that are correlated to that clan/crowd. Therefore, the higher the correlation parameter is, the more people in a particular real-world geographical region group together in the same virtual location in the game.

Each avatar has an interactive zone, denoted as a circle of diameter D_I, and a background zone, denoted as a circle of diameter D_Z ($D_Z > D_I$). The positions of avatars determine a subset of participants who are in the hearing range of each other.

III. Optimization Algorithms for the Distributed Locale Server Architecture

A. Problem description

It is assumed that a game service provider can have access to a number of servers located over the Internet. The objective is to design a server assignment algorithm to minimize the total interactive communication delays, which we call *delay cost*. We partition the virtual world into N square locales. Let P be the number of potential processing servers. Each locale is processed by only one server. It is possible to assign one server to more than one locale. If avatars in two adjacent locales, which are assigned to two different servers, are in the hearing range of each other, the two servers need to exchange the required audio streams. The inter-server delays introduces additional delays for the group communication.

Our aim is to find the optimal way to partition the virtual world into locales and then choose the locale servers in such a way that reduces the total interactive communication delay perceived by all avatars. We assume that each server has enough capacity to create audio scenes for the whole game. The distribution of servers therefore will be solely based on latency requirements as opposed to resource limitations (LDD as opposed to RDD). Server resource limit can be considered later by putting capacity constraints in the algorithms described in the next section.

B. Mathematical programming formulation

We use the following mathematical programming to model the problem. Fist of all, the known parameters are defined as follows. Let c_i^s be the sum of delays from server s to all avatars in locale i and d_{st} be the delay between server s and server t. Let m_{ij} be the number of avatars in locale i that are in interactive communication with avatars in the neighbor locale j. If locales i and j are assigned to two different servers, say s and t, the inter-server delay cost due to this assignment is denoted as k_{ij}^{st}, and $k_{ij}^{st} = m_{ij} d_{st}$.

Decision variables:

$$x_i^s = \begin{cases} 1 & \text{if server } s \text{ is chosen for locale } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Objective function:

Minimize:

$$\sum_{1 \leq i \leq N; 1 \leq s \leq P} x_i^s c_i^s + \sum_{1 \leq i,j \leq N; 1 \leq s,t \leq P; i \neq j; s \neq t} x_i^s x_j^t k_{ij}^{st} \tag{2}$$

subject to

$$\sum_{s=1}^{P} x_i^s = 1 \quad \forall i : 1 \leq i \leq N \tag{3}$$

This mathematical programming is non-linear. We use a simple method to linearize it as follows. Let y_{ij}^{st} be a binary decision variable that has the following property.

$$y_{ij}^{st} = \begin{cases} 1 & \text{if } x_i^s = x_j^t = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The former problem is transformed to the following linear programming (LP).

Minimize:

$$\sum_{1 \leq i \leq N; 1 \leq s \leq P} x_i^s c_i^s + \sum_{1 \leq i,j \leq N; 1 \leq s,t \leq P; i \neq j; s \neq t} y_{ij}^{st} k_{ij}^{st} \tag{5}$$

subject to

$$\sum_{s=1}^{P} x_i^s = 1 \quad \forall i : 1 \leq i \leq N \tag{6}$$

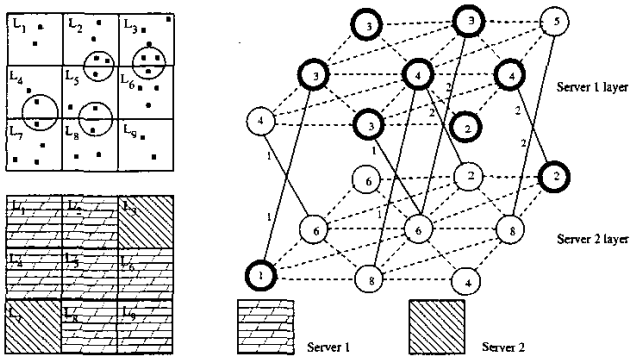


Fig. 3. Graph representation and optimal server assignment

$$y_{ij}^{st} \leq x_i^s \tag{7}$$

$$y_{ij}^{st} \leq x_j^t \tag{8}$$

$$x_i^s + x_j^t \leq 1 + y_{ij}^{st} \tag{9}$$

The constraints in equation 6 ensure that each locale is processed by one server. The constraints in equations 7, 8, and 9 are to ensure the condition in equation 4 is satisfied. Basically, these constraints ensure that if two adjacent locales are assigned to two different servers, the delay cost due to inter-server communication is taken into account.

Since there are N locales and P servers, the number of x_i^s variables is NP. Each locale may have from three to eight adjacent locales depending on position of the locale in the virtual world. For each locale that is already assigned to a server, there are P different server assignment possibilities for each of its adjacent locales. As a result, the number of y_{ij}^{st} decision variables is in the order of NP^2. Therefore, the number of binary decision variables for this problem is in the order of $O(NP+NP^2)$. Consequently, when the problem size is large, it is crucial to devise a heuristic to solve the problem efficiently. In the next section, we provide a greedy heuristic based on a graph algorithm.

C. *Greedy heuristic algorithm*

C.1 Graph representation

In order to solve this problem, we propose a multi-layer graph $G = (V, E)$, where $|V| = PN$ and $V = \{v_i^s : 1 \leq i \leq N; 1 \leq s \leq P\}$. This graph has P layers corresponding to P servers, and each layer has N vertices corresponding to N locales. Each vertex v_i^s (vertex i in layer s) has a delay cost c_i^s. If locales i and j are adjacent, there is an edge connecting any pair of v_i^s and v_j^t, denoted as e_{ij}^{st}, which has delay cost k_{ij}^{st}. The problem is to find a subgraph $G' \subseteq G$ that covers N vertices corresponding to N locales and has the minimum total vertex costs and edge costs. G' should have the property that if any two vertices of G are in G', the edge connecting these vertices is also in G'.

This graph representation is similar to graph partitioning algorithms in distributed virtual environment (DVE) in [6][7]. However, these algorithms are designed for load balancing, therefore, the cost of each vertex (the computation cost of each locale) is the same for any server since this cost only depends on the number of avatars in that locale. Therefore, only a single-layer graph is used. In this work, the delay cost of each vertex depends on the spatial location of a server with respect to physical distribution of participants in that locale. In addition, the cost of an edge connecting two vertices also depends on the servers that are assigned to each vertex. As a result, a multi-layer graph is required to capture this behaviour.

Fig. 3 shows a virtual world consisting of nine locales and the resulting 2-layer graph representation of the virtual world when two servers are used. The figure also presents the optimal server assignment solution. The vertex cost in each layer denotes the delay cost from each server to all avatars in the locale corresponding to that vertex. The circles denote a set of avatars in adjacent locales who communicate interactively. This results in the inter-layer edge costs in the graph. The zero cost inter-layer edges are not shown to avoid cluttering the figure. In each layer, all edges have zero cost (since adjacent locales are served by the same server) and are denoted as broken lines in the figure. As shown in the figure, the costs from server 2 to all locales are higher than those from server 1 except for locales L_2, L_3, L_7. Therefore, locale L_3 and L_7 are assigned to server 2. Locale L_2 is not assigned to server 2 since the reduction in vertex cost is 1, while the additional edge cost is 2. All other locales are assigned to server 1.

C.2 Greedy algorithm

In this algorithm, we aim to select a sub-graph $G' \subseteq G$, that covers N vertices and has the smallest sum of vertex costs and edge costs. First of all, we choose a starting vertex for which the sum of the vertex cost and its average outgoing edge costs is minimum. Then, we repeat the following procedures: choose the vertex that has the minimum of its vertex cost and costs to all vertices in the existing sub-graph that are adjacent to it. When a vertex in one layer is added to the sub-graph, the corresponding vertices in all other layers as well as all edges connecting to these vertices are deleted. This is to make sure that each locale is processed by only one server. The algorithm finishes when N vertices are covered. The complexity of this algorithm is similar to Minimum Spanning Tree problem [13], which is $O((NP)^2)$.

Pseudo code:

1. Initialize
$G'(V_{sub}, E_{sub}) = 0$;
$Neighbor(V_{nb}) = 0$;
2. Choose the first vertex v_i^s s.t (c_i^s + mean(k_{ij}^{st})) is minimum. Add v_i^s to G'.
3. While $size(G') < N$ do
   4. Update the set of *Neighbor*: $V_{nb} = \{v_j^t\}$ s.t $\{v_j^t \notin V_{sub}$ and $\exists\, e_{ij}^{st}$ s.t $v_i^s \in V_{sub}\}$.
   5. For each $v_j^t \in V_{nb}$, compute $cost_j^t = c_j^t + \sum e_{ij}^{st}$ s.t $v_i^s \in V_{sub}$
   6. Add v_j^t to G' s.t $cost_j^t$ is minimum
   7. Remove v_j^t from V_{nb}, delete all v_j^u s.t $\{u \neq v, 1 \leq u \leq P\}$ and all edges connected to these vertices.
8. End

As an example in Fig. 3, this algorithm chooses vertex v_7^2 (vertex corresponding to locale L_7 and server 2), then, v_4^1, v_1^1, v_2^1, v_5^1, v_9^1, v_8^1, v_6^1, and finally, v_3^2.

If we need to consider the maximum load of each server, additional constraints are required for the LP model. For the greedy

heuristic, a constraint can be put in step 4. A vertex is added to the *Neighbor* set only when the server corresponding to that vertex is not overloaded. In addition, the server load is incremented each time a vertex is added to G' in step 6.

D. Impact of avatar movements on optimal server assignment

Movements of avatars in the virtual world change the composition of locales. This means that as time goes by, the current partition and assignments of locales to servers drifts from optimal. In this situation, re-partitioning and re-assignment of locales to servers are necessary to reduce the total delay cost.

While avatar movements may occur constantly, joining and leaving the game by participants tend to occur on a longer time scale. Especially, game players in different geographical regions tend to join and leave the game in six different 4-hour blocks during a day [10]. In addition to avatar movements, this behavior may also affect optimal server allocation.

IV. Simulations

GT-ITM topology generator [15] is used to model the Internet topology. Specifically, we use a transit-stub graph of 600 nodes, comprising of three transit domains, which reflect three main geographic regions: North America, Europe, and Asia. Each domain has on average eight transit nodes, each transit node connects to three stub Autonomous Systems (AS), representing the connectivity of different ASs in each region. We randomly place 24 potential servers and 100 ISP POPs in these three regions. Each ISP POP has a uniform random number of game clients connected. Unless otherwise stated, the simulation results are generated with an average of 50 game players per POP. The topology generator parameters are chosen such that the maximum propagation delay in the shortest path between two nodes is 300ms. The interactive zone diameter D_I is 10m, the background zone diameter D_Z is 40m and the average number of avatars in each avatar's interactive zone in crowd/clan based virtual world is 2.5. The number of crowds/clans in a crowd/clan based virtual world is 50 and 250, respectively. In each delivery architecture, the interactive delay is defined as the average delay from an avatar to all other avatars it communicates with in the interactive zone. The bandwidth requirement is calculated based on the total number of links required by all audio flows.

A. Impact of server assignment algorithms on delay

In the following simulations, we compare our proposed greedy heuristic with the optimal results from the LP model. The optimal results are obtained from Cplex optimization software [14]. We partition the virtual world into different number of locales, run the two server assignment algorithms and calculate the total interactive communication delays for both cluster distribution (25 crowds) and uniform distribution. In these simulations we assume the correlation equal to 0.5. In uniform distribution, we divide the virtual world to 25 squares, and the correlation parameter specifies how people aggregate in each of these square based on their real-world geographic locations.

As shown in Table I, the results from the greedy heuristic are within 5% of optimal. Fig. 4 shows the close match between the two algorithms in crowd based distribution. The numbers in each square represent the server number that is assigned to that locale.

Locales	Optimal	Time	Greedy	Time	Gap(%)
16	371726	0.47(s)	382684	0.01(s)	2.9
36	355397	1.32(s)	356825	0.05(s)	0.4
64	340345	0.74(s)	343153	0.18(s)	0.8
100	324486	3.8(s)	329845	0.45(s)	1.6
400	321532	670(s)	324305	8.94(s)	0.8
900	312306	12h10m	320773	77.1(s)	2.7

(a) Uniform

Locales	Optimal	Time	Greedy	Time	Gap(%)
16	375154	0.23(s)	375463	0.01(s)	0.08
36	360761	0.46(s)	360852	0.04(s)	0.04
64	360392	0.60(s)	360542	0.17(s)	0.04
100	356572	5(s)	357042	0.43(s)	0.13
400	346074	77(s)	349397	7.88(s)	0.9
900	337340	3840(s)	343556	62.7(s)	1.8

(b) Cluster

TABLE I

Comparison between optimal results and greedy heuristic

1	1	5	5	1	1	1	1
1	1	5	5	1	1	1	0
1	1	1	3	6	4	0	0
1	1	1	1	3	0	2	2
1	3	5	0	0	0	2	2
1	5	5	7	0	7	6	0
9	4	1	7	1	5	2	7
9	4	4	1	9	5	5	3

Greedy heuristic

1	1	5	5	1	1	1	1
1	1	5	5	1	1	1	0
1	1	1	3	4	3	0	0
1	1	1	1	3	0	2	2
1	3	5	0	0	0	2	2
1	5	5	7	0	7	6	0
9	4	1	7	1	5	2	7
9	4	4	1	9	5	5	2

Optimal result from Cplex

Fig. 4. Server assignment results from Cplex and Greedy heuristic in cluster distribution

The difference between the greedy heuristic and optimal result is highlighted by shading locales for which the greedy heuristic provides a different result. In addition, it is noted from Table I that the greedy heuristic is scalable while the running time for the LP model is exponential, which would not be suitable for large problems.

It is also shown that increasing the number of locales reduces the delay cost. As indicated in Table I, the delay cost is reduced by nearly 20% in uniform distribution and about 10% in cluster distribution when the number of locales is increased from 16 to 900. This is expected since reducing the size of locale would improve granularity of assigning servers based on delay requirements of each avatar.

B. Effect of changes in the number of server and physical/virtual world correlation

In this experiment, we investigate the effect of varying the ratio of the number of distributed locale servers to the number of POPs and virtual/physical world correlation on the interactive delay in a crowd based game. An optimal central server is chosen among 24 potential servers. As shown in Fig. 5, at high correlation parameters, increasing the number of distributed servers would reduce the interactive delays considerably. When all 24

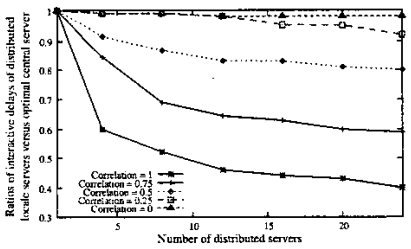
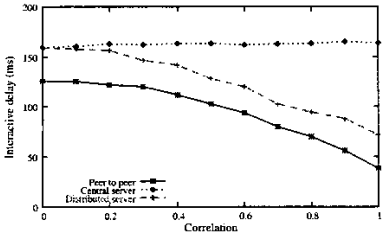
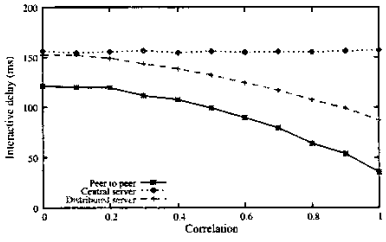


Fig. 5. Effect of changes in the number of server and virtual/physical world correlation on interactive delay in a crowd based game.

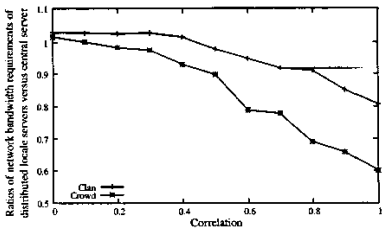


(a) Crowds

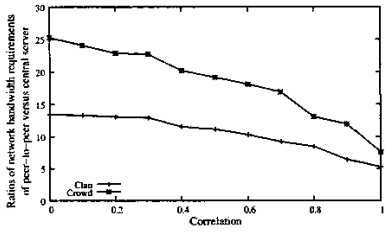


(b) Clans

Fig. 6. Effect of virtual/physical world correlation on interactive delays in a clan/crowd based game.



(a) Distributed locale server



(b) Peer-to-peer

Fig. 7. Effect of virtual/physical world correlation on resource usage of distributed server and peer-to-peer versus central server in a crowd/clan based game.

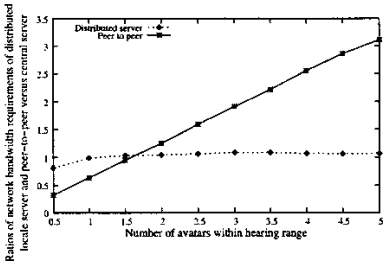


Fig. 8. Network bandwidth requirements in a loner based game

servers are used, the interactive delay of the distributed locale server architecture is reduced by approximately 60% and 20% at correlation of 1 and 0.5, respectively. However, at low correlation, this improvement is less significant. In addition, when the number of servers goes above a certain value, approximately one eighth of the number of POPs, further improvements in the interactive delays are very small.

C. Comparisons of delivery architectures for different player grouping behaviors

In the following experiments, the effect of changes in physical/virtual world correlation on the interactive delays and network bandwidth requirements of the distributed locale server architecture is investigated. Simulations are carried out to evaluate the performance of peer-to-peer, central server and distributed locale server architectures in different avatar grouping behaviors. In the delay performance comparison, the peer-to-peer architecture, which has the lowest delay, is used as a benchmark. In the network bandwidth resource usage comparison, the central server architecture is used as a benchmark.

As shown in Fig. 6a, in a crowd based game, the interactive delay of the central server architecture does not depend on correlation, while the interactive delay of the distributed locale server architecture is reduced when the correlation increases. In particular, this delay is reduced by about 60% and 20% at correlation of 1 and 0.5, respectively, and it closely follows the "best case" peer-to-peer delay. A similar behaviour is shown for a clan based game in Fig. 6b.

As shown in Fig. 7a, the distributed locale server architecture uses equal or less network bandwidth resources than the central server architecture. In particular, when the correlation increases from 0 to 1, the bandwidth usage of the distributed locale server architecture is reduced to 40% and 20% below that of the central server architecture in a crowd based game and a clan based game, respectively. The peer-to-peer architecture has large bandwidth usage in this case. As indicated in Fig. 7b, the ratios of network bandwidth requirements of this architecture versus the central server increase from above 5 at correlation of 1 to about 14 (in clan based game) and 25 (in crowd based game) at correlation of 0. It is noted that a crowd based game requires nearly twice the network bandwidth resources of a clan based game due to a larger number of avatars in each avatar's hearing range.

In the final experiment, we simulate a loner based virtual

world, in which, the average number of avatars in hearing ranges is varied from 0.5 to 5, and the correlation parameter is equal to 0. As indicated in Fig. 8, the ratio of network bandwidth requirement of the peer-to-peer architecture versus the central server architecture increases linearly from under 0.5 to about 3. The network bandwidth requirement of the distributed locale server architecture is approximately equal to that of the central server architecture. Therefore, in loner based game, the bandwidth usage of the peer-to-peer architecture is low and not much different from the other architectures.

D. Summary of results and recommendation

In this study, we have investigated the performance of the distributed locale server architecture based on a range of game player grouping behaviours and physical/virtual world correlation. The following recommendations are proposed:

- For loner based games, the peer-to-peer architecture would be favored due to low delays in direct paths between avatars and acceptable levels of bandwidth usage, however, security/anonymity issues must be addressed.
- For games consisting of crowds and clans, the distributed locale server architecture is most appropriate due to improvements in interactive delays over the central server, especially when there is a correlation between the virtual world and the physical world.
- In many games, where all different player characteristics may be apparent in different parts of a virtual world, a hybrid architecture would be most suitable, in which, parts of the virtual world that mainly consist of loners would use the peer-to-peer architecture, while crowds or clans would use the distributed locale server architecture.

V. Related work

There are few papers in the literature that have considered audio communication in virtual environments. The work in [2] discusses adding voice to the Mimaze based on multicast. Another work in [3] proposes an architecture called "distributed partial mixing" that effectively provides audio communication in collaborative virtual environment by adapting audio mixing functions to network congestion. In all these work, audio flows are sent in peer-to-peer or peer-to-peer multicast. Our work is different by defining an "immersive audio environment" that creates a personalized audio scene for each participant. This requirement has significant effect on the choice of delivery architectures. When the number of avatars within hearing range is large, the peer-to-peer architecture may not be appropriate due to limited access bandwidth at end users. Distributed server architectures has been proposed in several papers with the purpose of designing scalable state information servers for large multiuser virtual environments including network games in references [4][5]. Research results in load balancing and load sharing for distributed server architectures in DVE are presented in [6][8] and references therein. These works are not related to audio processing server as described in this paper. Moreover, the distribution of servers in this paper is driven by latency requirement as opposed to load balancing and load sharing. Finally, research results on spatial audio rendering techniques [11][12] can be applied to our immersive audio scene creation servers.

VI. Conclusions

In this paper, we propose a distributed server architecture to improve the communication delay perceived by users and increase scalability of an immersive audio communication service. We also propose algorithms to optimize the performance of this architecture. We present the result of our analysis and simulation and discuss various factors that might improve the group audio communication delay. In particular, we investigate the effect of changes in the number of servers and the correlation between distribution of avatars and game participants on communication delays and network resource usage. From a simulation study of different avatar aggregation behaviours, it is demonstrated that our distributed locale server architecture can reduce the overall delay by around 20% compared to an optimally located central server and can have lower network bandwidth usage than the central server architecture. Our optimization algorithms and simulation study will be of benefit to future immersive audio communication service providers in the design of a cost effective delivery architecture for this service.

Our future work will investigate the impact of avatar movements and game player joining/leaving on the distributed locale server architecture. In this situation, the optimization algorithm can be rerun to dynamically assign servers to locales in order to reduce interactive communication delays between players.

Acknowledgement

This work is supported by Co-operative Research Centre for Smart Internet Technology (SITCRC), Australia.

References

[1] T. A. Aas, S. Brown, B. Green, S. Motte, "Show me the money! Revenue models for massively multiplayer games," in *Game developers conference*, 2003, available at www.gdconf.com/archives/2003/

[2] J. C. Bolot, S. Fosse-Parisis, "Adding voice to distributed games on the Internet," in *Proc. IEEE Infocom*, 1998, pp. 480-487.

[3] M. Radenkovic, C. Greenhalgh, S. Benford, "Deployment issues for multiuser audio support in CVEs," in *ACM Symposium on Virtual Reality Software and Technology*, 2002, pp. 179-185.

[4] D. B. Barrus, R. C. Waters, D. B. Anderson, "Locales: Supporting large multiuser virtual environments," *Computer Graphics and Applications, IEEE*, vol. 16, no. 6, pp. 50-57, 1996.

[5] C. Diot, L. Gautier, "A distributed architecture for multiplayer interactive applications on the internet," *IEEE Networks magazine*, July/August 1999, pp. 6-15.

[6] J. Lui and M. Chan, "An efficient partitioning algorithm for Distributed Virtual Environment Systems," *IEEE Transactions on Parallel and Ditributed Systems*, vol. 13, no. 3, pp. 193-211, 2002.

[7] John C.S. Lui, Oldfield K.Y. So, M.F. Chan, T.S. Tam, "Dynamic partitioning for a Distributed Virtual Environment," in *Proceedings of the 3rd High Performance Computing Asia Conference (HPC Asia'98)*, 1998.

[8] Ta Nguyen Binh Duong, Suiping Zhou, "A dynamic load sharing algorithm for massively multiplayer online games," in *Proc. IEEE ICON*, 2003, pp. 131-136.

[9] Cong Nguyen, Farzad Safaei, Don Platt, "On the provision of immersive audio communication to massive multi-player online games," in *IEEE Symposium on Computers and Communications*, 2004, pp. 1000-1005.

[10] Wu-chang Feng, Wu-chi Feng, "On the geographic distribution of on-line game servers and players," in *Proc. Netgames*, 2003, ACM Press.

[11] C. Kyriakakis, P. Tsakalides, T. Holman, "Surrounded by sound," in *IEEE Signal processing magazine*, vol. 16, no. 1, pp. 55-66, Jan 1999.

[12] Martin Naef, Oliver Staadt, Markus Gross, "Spatialized audio rendering for immersive virtual environments," in *ACM Symposium on Virtual Reality Software and Technology* 2002, pp. 65-72.

[13] Douglas B. West, "Introduction to graph theory," Prentice Hall, 2001.

[14] Cplex optimization software, available at http://www.ilog.com

[15] GT-ITM: Georgia Tech internetwork topology models, available at http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html