2007

# The impact of avatar mobility on distributed server assignment for delivering Mobile Immersive Communication Environment

Ying Peng Que
*University of Wollongong*, ypq01@uow.edu.au

Farzad Safaei
*University of Wollongong*, farzad@uow.edu.au

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

# The impact of avatar mobility on distributed server assignment for delivering Mobile Immersive Communication Environment


## Abstract

In our previous work, we proposed a distributed server architecture to deliver multi-party immersive voice communication service to mobile clients (e.g. Sony PSP) accessing a distributed virtual environment (DVE). We refer to such immersive voice communication service as mobile immersive communication environment (MICE). We also proposed solutions to assign server to create the auditory scene for each avatar. In this work, we ascertain the necessary update period of server reassignment in order to cope with avatar mobility in the virtual world. In our simulations, we measure the percentage differences in delay deviation and bandwidth cost between the case of no server reassignment update since t=l second (the initial time instant) and the case of ideal server reassignment at every second. Our results show that the impact of avatar mobility leads to increases in both delay deviation and bandwidth cost as the time elapse further from t=l second. Such increase in delay deviation and bandwidth cost is more significant at the low avatar densities than at the higher avatar densities. The optimal server assignment algorithm is found to be too computationally intensive to be executed within the required update periods. A much faster greedy heuristic has been devised to perform server assignment within the required update periods.



## Disciplines

Physical Sciences and Mathematics

## Publication Details

Que, Y., Safaei, F. & Boustead, P. A. (2007). The impact of avatar mobility on distributed server assignment for delivering Mobile Immersive Communication Environment. IEEE International Conference on Communications (pp. 1576-1581). USA: IEEE.

*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.*

# The Impact of Avatar Mobility on Distributed Server Assignment for Delivering Mobile Immersive Communication Environment

Ying Peng Que, Farzad Safaei, Paul Boustead,
Smart Internet CRC,
Telecommunications and Information Technology Research Institute,
University of Wollongong, Australia
Email {ying, farzad, paul}@titr.uow.edu.au

***Abstract*-In our previous work, we proposed a distributed server architecture to deliver multi-party immersive voice communication service to mobile clients (e.g. Sony PSP) accessing a Distributed Virtual Environment (DVE). We refer to such immersive voice communication service as Mobile Immersive Communication Environment (MICE). We also proposed solutions to assign server to create the auditory scene for each avatar. In this work, we ascertain the necessary update period of server reassignment in order to cope with avatar mobility in the virtual world. In our simulations, we measure the percentage differences in delay deviation and bandwidth cost between the case of no server reassignment update since t=1 second (the initial time instant) and the case of ideal server reassignment at every second. Our results show that the impact of avatar mobility leads to increases in both delay deviation and bandwidth cost as the time elapse further from t=1 second. Such increase in delay deviation and bandwidth cost is more significant at the low avatar densities than at the higher avatar densities. The optimal server assignment algorithm is found to be too computationally intensive to be executed within the required update periods. A much faster greedy heuristic has been devised to perform server assignment within the required update periods.**

***Keywords: Mobility Management for Wireless Multimedia, Network Applications & Services, VoIP Services for Online Multimedia and Gaming Applications***

## I. INTRODUCTION

Recently, networked voice communication services in Distributed Virtual Environments (DVE) have attracted much research interest [1] [2] [3]. One typical example of DVE is Multi-player Online Games (MOG) such as Lineage II which had 2.1 million subscribers in January, 2005 [4]. Each DVE user is represented by an avatar in the virtual world. In a DVE, multiple users concurrently explore the high quality visual scenes of the virtual world. However, close interactions and co-operations among avatars are likely to be improved with the addition of a multi-party immersive voice communication service, which immerses each avatar in a personalised *Auditory Scene*. The auditory scene of a particular avatar is a mixture of the voice streams from all the surrounding avatars in the hearing range, with each voice stream directionally rendered and distance-attenuated with respect to the corresponding avatar position in the DVE. One special class of DVE is mobile DVE in which users access the DVE using mobile client devices such as laptops, Palmtops and mobile gaming platforms (e.g. SONY PSP and Nintendo DS). We refer to the immersive voice communication service for mobile DVEs as Mobile Immersive Communication Environment (MICE). In [5], we identified two important scalability constraints for the provisioning of MICE. The first constraint is that the access bandwidth of the wireless mobile devices is scarce in comparison to the wired devices due to the limited transmission spectrum and propagation path loss. Although the onboard processing power of mobile devices continue to improve (e.g. Pentium-M processor technology), the computation resources on mobile devices can be still regarded as scarce. The limited battery capacities on the mobile devices restrict the duration and complexity of the on-board processing operations.

The Computation Reduction Scheme and the central server architecture proposed in [5] address these scalability challenges with respect to bandwidth and processing. A distributed server architecture and two server assignment algorithms are proposed in [6] to further improve the delay performance of the central server architecture. However there lies the third challenge in the fact that our distributed servers assignment solutions are sensitive to avatar mobility in the virtual world which can occur frequently as avatars explore around the virtual world. Our distributed servers are assigned on the basis of virtual world avatar distribution. As discussed in Section III. A., avatar distribution changes significantly with virtual world avatar mobility. Therefore in this work, we investigate the impact of virtual world avatar mobility on the quality of static server assignment solution and the necessary update period for server reassignment. There is an interesting prior work [8] which deals with the management of virtual world mobility and physical world mobility. However, the work in [8] is concerned with the reconfiguration of multicast trees in response to avatar/client movements. The distributed server architecture studied in [8] is a distributed proxy model which means server assignment is only affected by physical world mobility. In the case of our distributed server architecture, both virtual world and physical world movements can necessitate changes in server assignments. Out work herein focuses on the impact of virtual world mobility. The management of physical world mobility will take advantage of the Mobile IP [10] and require server

**1-4244-0353-7/07/$25.00 ©2007 IEEE**

reassignment on a large time scale. Such issues will be addressed in our future publications.

The rest of the paper is organised as follows: Section II briefly reviews the previously proposed distributed server architecture for delivering MICE. The two server assignment algorithms are briefly reintroduced in Section II. The relationship between avatar mobility in virtual world and the update period of server reassignments is analysed in the simulation results presented in Section III. The final conclusion is given in section VI.

## II. System Overview

### A. Distributed Server Architecture

As elaborated in [6], we proposed a distributed server architecture depicted in Fig. 1 to improve the delays of the Optimally Placed Central Server architecture. We define avatar density as the average number of avatars in direct communication with each avatar (in its auditory scene). We assume that for a given virtual world application, users would have a certain threshold of tolerable voice delay [6]. The quality of voice communications between avatars is impaired when the delay between avatars exceeds or violates the defined threshold. Our Auditory Scene Creation (ASC) server assignment algorithm seeks to find a set of servers for all the avatars that minimises the sum of *delay deviations* from the pre-defined *voice delay thresholds*. We find in [6] that over the range of avatar densities from 10 to 50, the improvement in delay deviations by our distributed server architecture over the optimally placed central server architecture is between 54% and 58%.

### B. Bandwidth Unconstrained Minimal Delay Deviation Server Assignment Algorithm

The first ASC server assignment algorithm described in [6] is the *Bandwidth Unconstrained Minimal Delay Deviation* Server Assignment Algorithm (Bandwidth Unconstrained Algorithm for short). The Bandwidth Unconstrained Algorithm minimises the total upload access bandwidth cost which is measured as the total number of voice uploads from all the avatars to the ASC servers, while being subject to the constraint that the individual delay deviations are zero or at the minimum feasible according to the node-to-node delays of the underlying network. In essence, the Bandwidth Unconstrained Algorithm returns the maximum bound on the upload access bandwidth cost of a distributed server MICE system. If we relax the bandwidth cost constraint above this bound, we will not achieve any further reductions in the level of delay deviation. The Bandwidth Unconstrained algorithm is very scalable. For example, on a PC running Linux with AMD Athlon64 Dual-Core 2.0 GHz processor and 2.0 GB of RAM, the Bandwidth Unconstrained Algorithm can be rerun at once every 2 seconds to cope with avatar mobility for an avatar population of 20,000 and density of 50.

### C. Bandwidth Constrained Minimal Delay Deviation Server Assignment Algorithm

As found in [6], the upload access bandwidth cost incurred by the Bandwidth Unconstrained Algorithm is much larger than the central server case and is too large for the access bandwidth of the mobile clients, especially at high avatar densities. To reduce this upload access bandwidth cost, we proposed the *Bandwidth Constrained Minimal Delay Deviation Server Assignment algorithm* (Bandwidth Constrained Algorithm for short). The objective of the Bandwidth Constrained Algorithm is to minimise the sum of *delay deviations* from the pre-defined *voice delay thresholds* while the total upload access bandwidth cost is reduced significantly (e.g. by 50%) from the maximum bound returned by the Bandwidth Unconstrained Algorithm. The Bandwidth Constrained Algorithm achieves bandwidth cost reduction by choosing fewer servers than the Bandwidth Unconstrained Algorithm. Such bandwidth reduction is achieved at the expense of increasing delay deviations from the Bandwidth Unconstrained Algorithm. However the extent of such increase in delay deviations is minimised by the objective function of the Bandwidth Constrained Algorithm.

### D. Greedy Heuristic of the Bandwidth Constrained Server Assignment Problem

As found in [6], the Optimal Bandwidth Constrained Algorithm is NP-hard and computationally intensive as subsequently shown in Table 1. We have thus devised a greedy heuristic to find suboptimal solutions to this problem in a reasonable time. This greedy heuristic is devised to follow the behaviour of Optimal Bandwidth Constrained Algorithm and reduces the number of voice uploads by clients through assigning only a limited number of ASC servers capable of meeting the delay constraints of a large number of avatar pairs. Let $P$ denote the number of potential ASC Server Locations. Let $N$ denote the number of avatars. The complexity of this heuristic is less than $O(NP)$. Let $K$ denote the constraint of *maximum number of ASC servers* and $K<P$. Before the server number constraint is reached, the cost per iteration is $P$ from the linear search through all the potential server sites. After reaching that constraint, the cost per iteration is $K$ from the linear search through the set of $K$ servers already chosen from previous iterations. The Pseudo-Code is given in Appendix A.

## III. Simulations

In our simulation experiments we use a 5000 node Transit-Stub graph generated by the GT-ITM topology generator [7] to algorithm the Internet topology. The Transit-Stub network consists of five transit domains (each representing a geographic region). Potential servers are randomly chosen from the 5000 nodes and hence their placements are spread across these five regions. Each transit domain has an average of ten transit routers. Each transit router is connected to on average, nine stub domains (each representing an Autonomous System). Each stub domain consists of eight stub routers. We randomly place two Internet Service Provider (ISP) Point Of Presences (POPs) across the stub ASs connected to each transit node ($5 \times 10 \times 2$), which results in 100 ISP POPs widely spread across the network. The topology generator parameters are chosen such that the maximum propagation delay in the shortest path between two nodes is 160 ms. The size of avatar population studied is 200.

We measure the quality of server assignment in terms of the delay deviations and bandwidth cost incurred. In our simulations, we study the impact of avatar mobility on server assignment quality in two scenarios. In the first scenario, the server assignment solution obtained at the initial time of t=1 second is reapplied to handle the avatar distributions at subsequent time instants between 2nd second and 1000th second. In the second scenario, the servers are reassigned at the update period of once every second. While such high update frequency may not be practical for real implementation, this should provide a benchmark for the best (or ideal) performance that we could expect from a practical mobility management system. We implement the random waypoint motion algorithm [9] to simulate the movements of avatars in the virtual world. We discarded the first 1000 seconds of avatar movement data to avoid the avatar density fluctuation problem during early time instants [9]. The speed of avatar movement is between 0.1 meter per second and 1 meter per second. Such avatar speed range is reasonable for walking pace and can have impact on avatar distribution which has a hearing range set to 30 meters. The pause times for all avatars are set to be zero.

### *A. Impact of Avatar Mobility in the Virtual World*

Avatar mobility in the virtual world can significantly change the distribution of avatars. Avatar mobility can change the voice delay thresholds between a pair of communicating avatars. We measured the Cumulative Distribution Function (CDF) of such change in delay thresholds. For example for avatar densities between 5 and 25, across all the time instants we studied, we found that 30% of avatars experience changes in delay threshold between 28 ms and 46 ms which are significant considering the maximal delay threshold is set to be 100 ms. On the other hand, pairs of avatars originally in communication with each other can move out of communication (referred to as separated avatar pairs). Fig. 1 shows the percentage of avatar pairs originally in communication at t=1second that go out of communication due to mobility at different time instants after t=1 second. Over different avatar densities, the percentage of separated avatar pairs increase consistently as the time elapsing from t=1 second increases. Moreover, at the corresponding time instants, the percentages of separated avatar pairs are less at high avatar densities than at low avatar densities. Because of the close proximities of avatars in a dense virtual world, it takes longer for an avatar to move out of the communication zone of another avatar. These two types of mobility-induced changes in avatar distribution generate the need for regular update period of server reassignment.

### *B. The Impact of Avatar Mobility on the Quality of Server Assignments*

Fig. 2 and 3 show, over different avatar densities, the respective percentage increase in delay deviations and bandwidth costs, when comparing the case of no server reassignment since t=1 second against the case of ideal server reassignment at every second. In the case of Fig. 2 and 3, the bandwidth cost constraint of the server assignment algorithm is reduced by 50% from the maximum bandwidth cost found by the Bandwidth Unconstrained Server Assignment Algorithm as discussed in Section II. C. From Fig. 2, for all

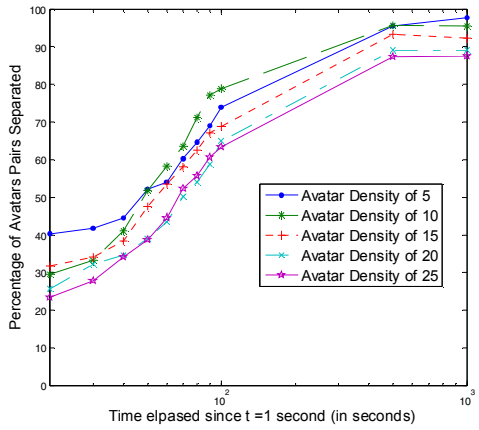


**Fig. 1 The percentage of pairs of avatars separated (out of communications) due to avatar mobility over different time instants.**

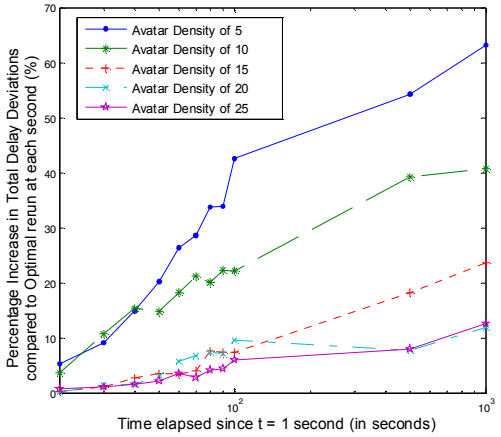


**Fig. 2 Percentage difference comparison in Delay deviations for 50% reduction from maximum Bandwidth Cost.**

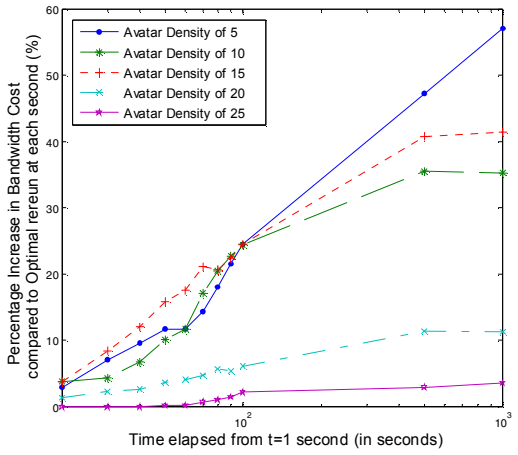


**Fig. 3 Percentage difference comparison in Bandwidth for 50% reduction from maximum Bandwidth Cost.**

densities studied, the additional delay deviations incurred by reusing the server assignment solution from t=1 second increases as the time elapsed from t=1 second increases. For example, at avatar density of 5, the total delay deviations incurred at 100 seconds when there is no server reassignment since t=1 second is 43% larger than the case of ideal server reassignment at every second. Fig. 3 reveals a similar deteriorating trend in the additional bandwidth cost incurred as tine elapses further from t =1 second. As shown in Fig. 1, due to avatar mobility, the changes in avatar distribution grow larger as time elapses further from the t=1 sec. Another interesting observation from Fig. 2 and 3 is that the extent of deterioration of server assignment quality with reducing update frequency is less at higher avatar densities than lower avatar densities. For instance, at avatar density of 25, at 100 seconds, the additional total delay deviations incurred in the case of no server reassignment since t=1 second is only 9% compared to the case of ideal server reassignment at every second. This is much smaller gap compared to the 43% observed at 100 seconds for avatar density of 5. This observation can be explained by the trend shown in Fig. 1, that the percentage of separated avatar pairs due to mobility is higher at lower avatar densities than at higher avatar densities.

We also studied the impact of avatar mobility at two other bandwidth cost constraints lower than the 50% reduction case shown in Fig. 2 and 3. Fig. 4 and 5 show the percentage difference comparisons in bandwidth cost and delay deviations when the bandwidth cost constraint is reduced by 60% from the maximum bandwidth cost.

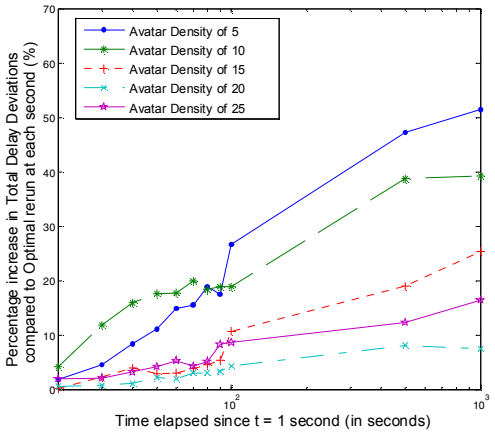


**Fig. 4 Percentage difference comparison in Delay deviations for 60% reduction from maximum Bandwidth Cost.**

Moreover, Fig. 6 and 7 show the percentage difference comparisons of bandwidth cost and delay deviations when the bandwidth cost constraint is reduced by 70% from the maximum bandwidth cost. Fig. 4 and 6 show that, when the bandwidth cost constraints are really low, the avatar density has a less significant effect on the increase in the additional delay deviations incurred over elapsing time instants, when there is no server reassignment since t=1 second. Furthermore, by comparing Fig. 4 and 6 against Fig. 2, we can see that at the corresponding time instants, the lowering of bandwidth cost constraint does not lead to a greater extent of increases in delay deviations. Both observations are due to the fact that at very low bandwidth cost constraints, even the delay deviations returned by the ideal server assignment algorithm are very large. By comparing Fig. 5 and 7 against Fig. 3, we find that the lowering of the bandwidth cost constraint, lead to larger extent of increases in bandwidth cost at corresponding update frequencies. This is especially true at low avatar densities of 5 and 10 avatars per communication zone. Such observations suggest that as time elapses further from t=1 second, the quality of the previous server assignment results deteriorates significantly and can not achieve comparable bandwidth cost to the very low bandwidth cost (e.g. 60% and 70% reduction from maximum) obtained by the once per second rerun of the optimal server assignment algorithm.

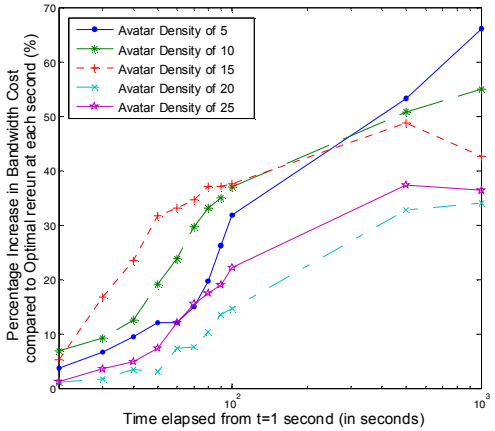


**Fig. 5 Percentage difference comparison in Bandwidth for 60% reduction from maximum Bandwidth Cost.**

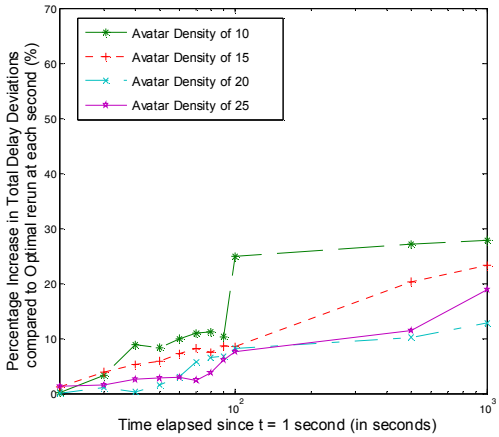


**Fig. 6 Percentage difference comparison in Delay deviations for 70% reduction from maximum Bandwidth Cost.**

### *C. Performance comparison between the Optimal Server Assignment Algorithm and the Greedy Heuristic*

Figures 8 and 9 illustrates the percentage increase in terms of total delay deviations and average upload access bandwidth cost of the greedy heuristic over the optimal Bandwidth Constrained algorithm, averaged over different time instants at various avatar densities. The performance gap between the optimal and the heuristic could be considered acceptable in all cases studied. The largest percentage difference in total delay deviations is 24%. In this case, the delay deviation of the

heuristic is still 40% lower than using an optimal central server. The largest percentage difference in bandwidth cost is 37.55%. In this case, on average, each avatar uploads 3.26 streams for an avatar density of 15 which is reasonable.

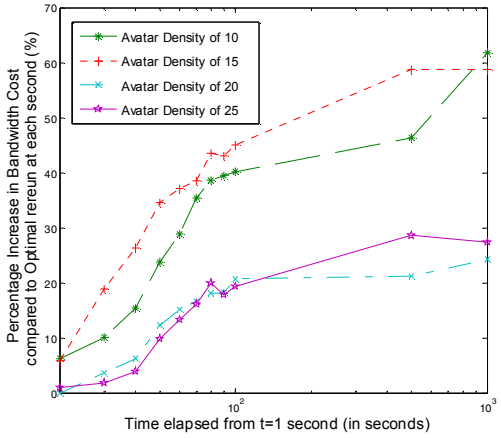


**Fig. 7 Percentage difference comparison in Bandwidth for 70% reduction from maximum Bandwidth Cost.**

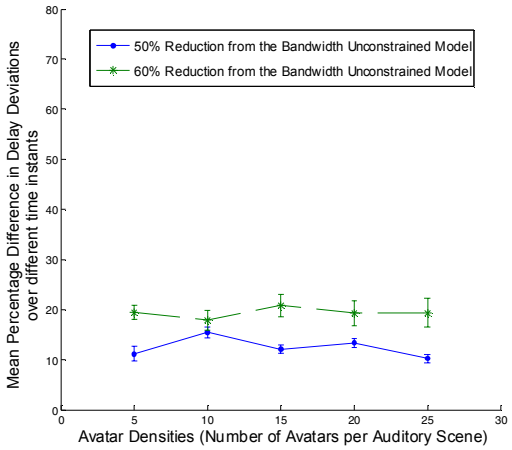


**Figure 8. Delay Deviation comparison between the Optimal Server Assignment Algorithm and the Greedy Heuristic with different bandwidth cost constraints.**

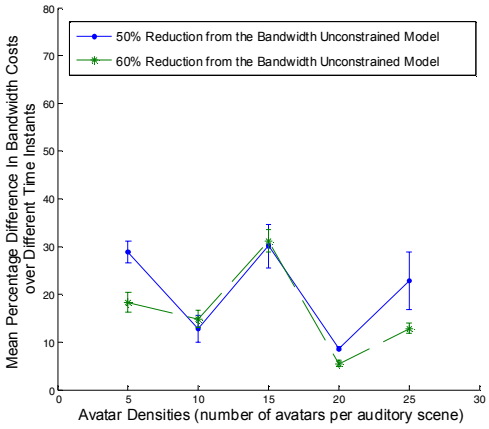


**Figure 9. Bandwidth cost comparison between the Optimal Server Assignment Algorithm and the Greedy Heuristic with different bandwidth cost constraints.**

From Figures 8 and 9, it can be observed that the rising avatar densities do not have any significant impact on the performance of the greedy heuristic. Figures 8 and 9 also show that the performance of the greedy heuristic does not change significantly when applied to different bandwidth cost constraints. We implemented our LP-based Bandwidth Constrained server assignment algorithm using the CPLEX software package [17]. The greedy server assignment heuristic was implemented in Matlab. We obtained our execution times for both the optimal solution and the heuristic on a PC running Linux with AMD Athlon64 Dual-Core 2.0 GHz processor and 2.0 GB of RAM. As shown in Table 1, in all cases studied, the execution time of the greedy heuristic is much faster than the execution time of the optimal server assignment algorithm, especially at high densities and/or at low bandwidth cost constraint (e.g. 60% reduction).

**Table 1.**

**Execution Time Comparison between the Optimal Bandwidth Constrained Server Assignment Algorithm and the Greedy Heuristic.**

| Avatar Density | 50% BW Reduction Optimal | 50% BW Reduction Heuristic | 60% BW Reduction Optimal | 60% BW Reduction Heuristic |
|---|---|---|---|---|
| 5 | 67.21s | 0.21s | 173.38s | 0.15s |
| 10 | 95.12s | 0.40s | 134.84s | 0.30s |
| 15 | 277.74s | 0.81s | 937.02s | 0.53s |
| 20 | 488.96s | 0.96s | 1127.48s | 0.74s |
| 25 | 533.56s | 1.47s | 2370.09s | 1.08s |

## *D. Summary of results and recommendations*

If we choose to never exceed the threshold value of 15% additional delay deviation over the benchmark case of server reassignment at each second, the following rules of thumb can be derived for the appropriate update period of server reassignment for the mobility model studied. If we assume that the 15% rise occurs only in the level of delay deviation violation in each avatar pair, this figure of 15% means at most each avatar pair incurs an additional delay of 15 ms on top of the 100 ms of maximal delay threshold. For low avatar densities equal to or below 10, the update period should be once every 30 seconds. For avatar densities from 15 to 20, the update period should be 100 seconds. For avatar densities higher than 20, the update period should be 1000 seconds. When the bandwidth constraint is set to 50% reduction from maximum (or a more relaxed constraint), the additional bandwidth cost incurred compared to the benchmark case is also about 15% or lower for these rules of thumb, similar to the delay deviation threshold value. However, as the bandwidth cost constraint is lowered to 60% or 70% reduction from maximum, the additional bandwidth cost constraint incurred will be much higher at these recommended update periods for the specified range of avatar densities. As shown in Table 1, for avatar densities less than 20, the heuristic must be applied as the optimal server assignment algorithm is not fast enough for all cases studied. For avatar densities above 20, under more relaxed bandwidth cost constraint of 50% reduction, the optimal server

assignment solution can be executed within the 1000 seconds update period. However, the avatar population size studied in Table 1 is only 200, for real world scenario with thousands of avatars, the heuristic is still required instead of the optimal solution at avatar densities above 20. Furthermore, the required update periods at low avatar densities below 10 are too fast when considering the complicated process of server handover and synchronisation. This issue leads to future studies.

## IV. CONCLUSION

In this paper, we study the relationship between the avatar mobility in the virtual world and the update period of server reassignments for Mobile Immersive Communication Environment. We find that avatar mobility can significantly change the composition of the communication zone of avatars. This in turn leads to the requirement of reasonable fast update periods between 20 second and 1000 seconds for different avatar densities. In our simulations, we measure the percentage difference in delay deviation and bandwidth cost incurred, between the case of no server reassignment since t=1 sec and the case of optimal server reassignment at every second. We find that in all cases studied, the percentage differences in both delay deviations and bandwidth costs increase as the time elapses further from t =1 second (corresponding to longer update periods). Moreover, we observe that this deteriorating trend is more significant at low avatar densities than at high avatar densities. This implies we can run the server assignment algorithm at a slower frequency for higher avatar densities (e.g. once every 1000 seconds for avatar densities >= 20) than for low avatar densities (e.g. once every 30 seconds for avatar densities <= 10). Furthermore, decreasing bandwidth cost constraint does not result in greater deteriorations in delay deviations but lead to greater deteriorations in bandwidth cost. We find that the greedy heuristic offers much faster execution time than the optimal server assignment algorithm. The performance gap between the heuristic and the optimal solution is not affected by rising avatar density. When matching the delay deviation between the optimal algorithm and the heuristic, we incur larger bandwidth cost than the optimal case but the bandwidth cost is still acceptable.

## ACKNOWLEDGEMENT

This work is supported by the Co-operative Research Centre for Smart Internet CRC (SITCRC) and the University of Wollongong (UOW), Australia.

## REFERENCES

[1] Paul Boustead and Farzad Safaei, "Comparison of Delivery Architectures for Immersive Audio in Crowded Networked Games", in *Proc. of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV),* Ireland, 16-18 June, 2004.

[2] J. Bolot and F. Parisis, "Adding Voice to Distributed Games on the Internet", in *Proc. of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies,* pages 480-487, 1998.

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

[4] MMOGCHART.com, http://www.mmogchart.com/ (9 Aug. 2005).

[5] Ying Peng Que, Paul Boustead, and Farzad Safaei, "Minimising the Computational Cost of Providing a Mobile Immersive Communication Environment (MICE)", in *Proc. of the 2nd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME 06) at Consumer Communications and Networking Conference (CCNC)*, Las Vegas, Nevada, U.S.A, 7-10 Jan. 2006.

[6] Ying Peng Que, Paul Boustead, and Farzad Safaei, "Distributed Server Architecture for the Over-lay Network Delivery of Immersive VoIP", submitted to *2007 Annual Conference of the ACM Special Interest Group on Data Communications (ACM SIGCOM 2007)*, Kyoto, Japan, 27-31 August 2007, available from http://www.uow.edu.au/~ypq01/.

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

[8] Mehran Dowlatshahi and Farzad Safaei, "Managing virtual and physical mobility for mobile immersive voice communications", in *Proc. of the 4th International Conference on Mobile and ubiquitous multimedia 2005 (MUM 05)*, Christchurch, New Zealand, 08-10 December 2005.

[9] Tracey Camp, Jeff Boleng and Vanessa Davies, "A survey of mobility models for ad hoc network research", in Wireless Communications and Mobile Computing: Special Issue on Mobile Computing, Vol 2, pp. 483-502, John Wiley & Sons, 2002.

[10] Charles E. Perkins, "Mobile IP", in Communications Magazine, Vol. 35, Issue 5, pp. 84-99, ISSN : 01636804, May 1997.

## APPENDIX A.

**Variables:**

$P$ – Total number of potential ASC Servers.

$N$ – Total number of avatars.

$S_{max}$: Maximum number of ASC Servers Allowed.

$L_s^R$: The List of ASC Servers Ranked including server Id and ranks determined in pre-processing.

$L_s^u$: The List of ASC Servers Used.

$J$: The current avatar Id to be processed.

$S_{cur}$: The current ASC server ID offering the minimal sum of Delay Deviations to $J^{th}$ avatar.

**Pseudo code:**

$J = 1$;

Initialise $L_s^u$ to be empty

While $J <= N$, (jump out if all avatars have been assigned an ASC Server)

If length of $L_s^u < S_{max}$

Iterate $L_s^R$ to find $S_{cur}$ offering the min sum of Delay Deviations;

If a tie situation in Delay, choose the server with higher rank;

If $S_{cur}$ is not found in $L_s^u$

Add $S_{cur}$ to $L_s^u$;

End

Else (Reached the Maximum Number of Servers Allowed)

(Limit the Search to $L_s^u$ instead of $L_s^R$)

Iterate $L_s^u$ to find $S_{cur}$ offering the min sum of Delay Deviations;

If a tie situation in Delay, choose the server with higher rank;

End

Assign J to $S_{cu}$;

$J=J+1$;

End