

1-1-2007

System architecture and mobility management for mobile immersive communications

Mehran Dowlatshahi
University of Wollongong, mehran@uow.edu.au

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

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Dowlatshahi, Mehran and Safaei, Farzad: System architecture and mobility management for mobile immersive communications 2007.
<https://ro.uow.edu.au/infopapers/1659>

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

System architecture and mobility management for mobile immersive communications

Abstract

We propose a system design for delivery of immersive communications to mobile wireless devices based on a distributed proxy model. It is demonstrated that this architecture addresses key technical challenges for the delivery of these services, that is, constraints on link capacity and power consumption in mobile devices. However, additional complexity is introduced with respect to application layer mobility management. The paper proposes three possible methods for updating proxy assignments in response to mobility management and compares the performance of these methods.

Keywords

System, architecture, mobility, management, for, mobile, immersive, communications

Disciplines

Physical Sciences and Mathematics

Publication Details

Dowlatshahi, M. & Safaei, F. (2007). System architecture and mobility management for mobile immersive communications. *Advances in Multimedia*, 2007 (1).

System Architecture and Mobility Management for Mobile Immersive Communications

Farzad Safaei, Mehran Dowlatshahi
Smart Internet Technology Cooperative Research Centre
University of Wollongong
{farzad, mehran}@uow.edu.au

Abstract:

In this article we propose a system design for delivery of immersive communications to mobile wireless devices based on a distributed proxy model. It is demonstrated that this architecture addresses key technical challenges for the delivery of these services, that is, constraints on wireless link capacity and computational resources in mobile devices. However, additional complexity is introduced with respect to mobility management. The article proposes three possible methods for updating proxy assignments in response to mobility and compares the performance of these methods.

1- BACKGROUND

The new generations of wireless technology are poised to relieve the user from the severe bandwidth constraints of the earlier systems, paving the way for true multimedia communications. Most current services over the Internet, such as Voice over IP telephony, are based on a single point-to-point communication channel between the end users. We are interested in the next generation of communication services that facilitate natural *multiparty* interaction and can be called immersive communications. In immersive communications, the visual and aural scenes of each user create a sense of being in the presence of a group of people. One possible instance where immersive communications can be useful is within the Networked Virtual Environments (NVE). NVE's are likely to form the basis of a large class of applications for education, entertainment and collaboration. If complemented with immersive communications, these environments can transform the Internet from a medium primarily used for search and retrieval of information to one that facilitates human interaction, collaboration and play.

There has been a significant increase in popularity of Networked Virtual Environments in recent years. For example, reliable estimates indicate that by 2009 more than 230 million people will be playing multiplayer network games and, in particular, mobile games show significant growth [1].

Natural human communication within an NVE requires creating a suitable multimedia scene (voice, video, gestures and haptics) for each participant to mimic the real world sensory information of being in the presence of a group or a crowd. The audio scene, for example, must include the voices of all avatars in the participant's hearing range, spatially placed at a suitable distance based on the participant's perspective.

Unlike the current person-to-person communication services, which are characterised by more or less static point-to-point traffic flows, immersive communication involves a myriad of point-to-multipoint flows with highly dynamic changes in their connectivity arrangements. For example, the voice of each participant has to be included in the audio scene of everyone within the audible range of this voice. Likewise, other multimedia content (visual, gesture and haptics information) sourced from a given participant should reach everyone who is 'interested' in this information.

Conceptually, one might view several parallel multicast flows from the source to others within the area of interest. This ‘area of interest’ may differ for different types of media. Voice, for example, could propagate through walls while visual information does not. As avatars move within the virtual environment, these multicast trees must undergo change. The immersive communications, therefore, is characterised by a large number of multicast flows that are subject to rapid reconfiguration.

Wireless access to multimedia immersive communications presents additional challenges.

1. The *wireless access bandwidth* is more likely to be a bottleneck compared to wired access. This would be especially important for the downstream capacity where the required number of streams would depend on how ‘crowded’ the participant’s surrounding environment is.
2. The computational *resources in wireless devices*, especially the battery operated ones, are likely to be scarce or expensive to use. Hence, it would be desirable to relieve these clients from excessive low level flow management functions (such as continuous changes in multicast trees in response to mobility in the virtual environment).
3. The *mobility* of wireless devices across networks will create new challenges for mobility management and require new functions in addition to layer 2 and layer 3 handovers to control latency.

One approach to tackle these problems is to focus on developing higher speed wireless links, more powerful computational resources, and better batteries. The second approach, which is the subject of this article, is to perform certain functions in the fixed infrastructure to relieve the wireless devices from excessive computation and reduce the necessity of having a very complex wireless transmission system. Naturally, these two approaches are not mutually exclusive and can work together to offer the best outcome based on a given wireless technology.

In this article, a system architecture is introduced that can cater for scarcity of access bandwidth and computational resources in wireless devices (challenges 1 and 2 above). This enables the system to scale to crowded virtual spaces and cope with rapid changes in the virtual world. It is then shown that this architecture leads to new requirements for mobility management to control the latency after movement of wireless devices. Section 3 provides a possible solution for managing physical mobility and simulation results on its effectiveness (challenge 3). Section 4 presents the concluding remarks.

2- SYSTEM ARCHITECTURE

A conceptually simple model for wireless immersive communications is to use a peer-to-peer model for transmission of multimedia content between the participants. To use a concrete example, let us consider immersive voice communications for the following discussions. In a peer-to-peer model, each client captures the voice of its user and must identify the subset of participants who would be interested in this voice stream. The voice stream is then multicast to this subset. Alternatively, multiple unicast flows may be used if the underlying network cannot support multicast.

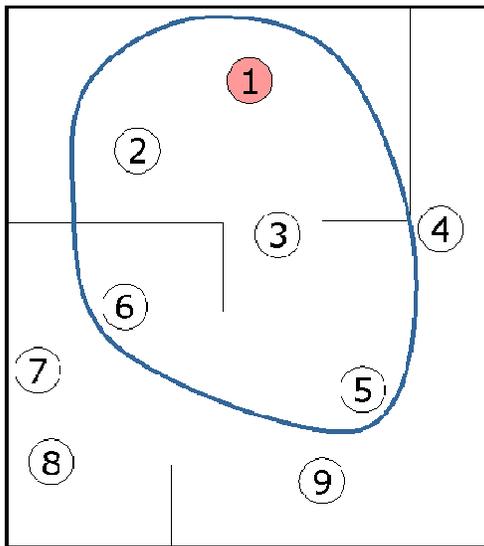
While simple, this model has some drawbacks. First, the downstream wireless capacity is a constraint and would limit the number of flows that can be received by each participant. Second, the clients have to participate in rapid reconfiguration of multicast flows in response to application dynamics (such as movement of avatars in the virtual world), which is a processing burden on wireless devices.

To overcome these difficulties, we propose to use a set of distributed servers – referred to as proxies – to aid in the delivery of multimedia streams to clients. Each proxy is responsible for a group of clients and, in essence, performs the necessary functions of these peers on their behalf.

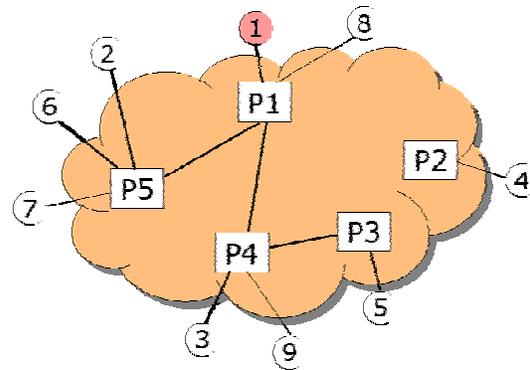
Figure 1 shows this architecture for a small NVE. Every wireless client is connected to a proxy server. To improve latency, it would be best to connect the client to its closest (in terms of network delay) proxy.

On the upstream side, the client will send its voice packets to its proxy. It is the responsibility of the proxy to forward/multicast this voice stream to other proxies who might need this information for the creation of their clients’ audio scenes. This is shown in Figure 1. Proxy P1 receives voice packets from one of its clients (avatar 1). P1 will determine the audible range of this signal by analysing its loudness and the characteristics of the environment (for example presence of sound barriers such as walls). The audible range is shown as a closed area in this Figure which includes several avatars, namely avatars 2-6. P1 will then determine the proxies for avatars 2-6 which happen to be P3, P4 and P5. Avatar 1’s voice packets are then multicast to P3, P4 and P5 with P1 as the root of the overlay multicast tree. Similarly, the proxies associated with all other (talking) avatars will create overlay multicasts for the purpose of communicating their client’s voice.

Coping with limited wireless link capacity



(a) Avatar Locations in the Virtual World



(b) Clients and their proxy servers

Figure 1: Distributed Proxy Architecture for Immersive Communications

On the downstream side, the proxy should somehow send the audio streams of relevance to its connected clients without exceeding the wireless link capacity. Let us assume that the downstream link capacity allocated to audio for a given client is equal to Kr bits per second (bps), where r is the required rate for a constant bit rate mono audio stream (in bps) and K is an integer. The proxy is responsible to ensure that the amount of voice information sent to this client does not exceed this

value regardless of how crowded the avatar's surrounding environment is. We have developed two possible mechanisms for this purpose.

- 1- In the first case, the proxy merely performs a simple “filter and forward” operation together with silent suppression at the clients. If voices are modelled as independent on-off audio sources (talk spurts and silence periods), by filtering out the silence periods from each stream, it should be possible to pack more voice streams within the downstream flow. The filtering out of the silence periods can be done at the originating client by running a silence detection algorithm. Each proxy uses a priority-based ordered list of all avatars within the hearing range of its client – for example, based on proximity in the virtual environment - and will send (at most) Kr bps of the active streams to the client. In this case, the first K high priority voice streams will always be received but there is also a good chance for packets from further away voices to also get through during the silence periods of these. In effect, these voices experience some ‘packet loss’ but only during periods of talk spurts by the closest avatars. In [2] it has been shown that despite its simplicity, this filter and forward operation can lead to good performance in reasonably crowded spaces even when the access bandwidth limit (value of K) is rather low. For example for $K=3$, obviously the first three talking avatars closest to the listener are heard with no packet loss. The packet loss of the fourth talking avatar, if any, would be below 10%. However, the losses only happen in bursts during those moments when the 4 avatars have simultaneous talk spurts. Whether this loss is subjectively significant requires further study. According to the same analytical results more than 80% of the talk spurts of the fifth talking avatar, if any, will also be heard.
- 2- In a second implementation, when the number of avatars in one's hearing range is greater than K , the proxy will group these into K separate clusters and perform a partial audio mixing operation for each cluster. The proxy will also calculate the ‘centre of activity’ of each cluster. This is the location of an imaginary audio source from which the cluster mix should emanate. The client will be able to render the audio scene by spatially rendering placing each mixed stream at its centre of activity. In this case, the whole audio scene is represented without any loss of voice packets. However, there may be some error in the spatial location of far away voices and proxies have to perform more computations (see [3] for details).

Coping with mobility in the virtual world

In the distributed proxy architecture, the wireless devices need not participate in formation and reconfiguration of multicast trees. Instead, each proxy will become the root of multicast for all of the streams sourced by its attached clients and will have to join all those multicast trees associated with streams needed by its clients. Apart from the fact that this will make the task of wireless devices much simpler, using proxies as opposed to peers for formation and reconfiguration of multicasts has another important advantage: the reconfiguration of multicast trees due to movements in the virtual world happen less often.

To illustrate this point consider Figure 1 once again. The movement of avatars will change the composition of crowds and the proximity of avatars to each other within a crowd. Consequently, the list of avatars in one's audible range will change due to movement of both the speaker and the listeners. This, in turn, may lead to a new multicast tree if any of the proxy leafs are different. Given that proxies are participating in multicast trees on behalf of all their attached clients, this change happens less often than a peer-to-peer model. For example, in Figure 1 if avatar 2 moves out of the audible range of avatar 1, the multicast tree from P1 will not change because there is still another avatar (6) connected to the same proxy that needs avatar 1's voice stream. In [4] we have proposed a

recursive overlay multicast algorithm for construction of multicast trees that is scalable to a large number of highly dynamic trees and will allow rapid reconfiguration on the time scales which are consistent with movements within a virtual environment.

3- MOBILITY MANAGEMENT

In the previous section we demonstrated that using a distributed proxy architecture can significantly improve scalability and robustness of immersive communication services for wireless nodes. The use of distributed proxies, however, creates an additional complexity - mobility management. This is particularly pertinent if the underlying network topology is hierarchical, which is of course very common.

To illustrate this point, consider the network of Figure 2 where a portion of a hierarchical infrastructure comprised of two stub domains interconnected using a transit domain is shown. A wireless client is initially connected to stub domain 1 through one of the routers associated with this domain as its care of address (CoA) router [5]. Let us assume that the client is mobile. Figure 2 shows a case when the mobile node has moved outside the range of its stub domain and connected to a new CoA router. Given timely handovers using layer 2 and layer 3, it should be possible for the wireless node to continue its multimedia communication session. However, the location of its proxy

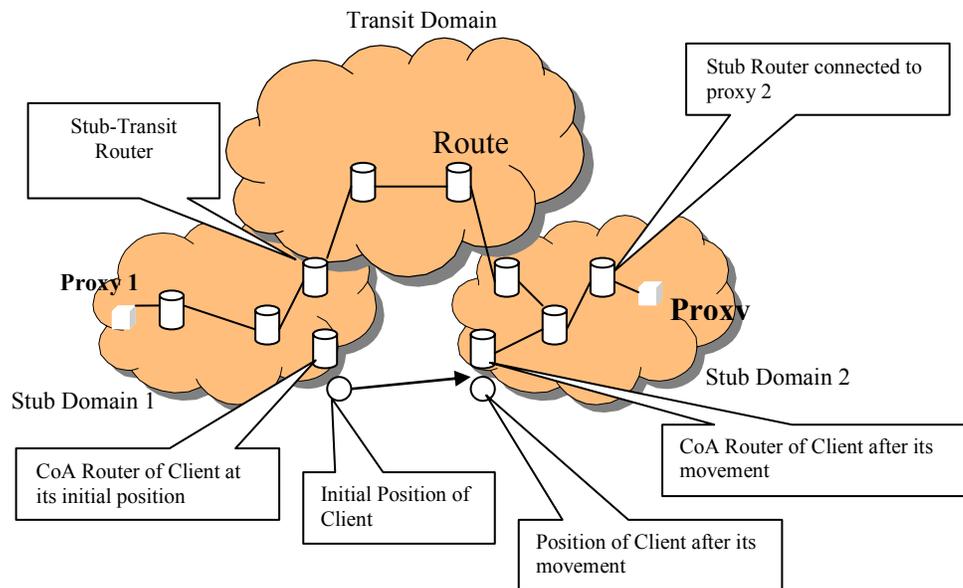


Figure 2: Movement of wireless node from one stub domain to another

may no longer be suitable. For example, the proxy may be connected to the old stub domain and the communication between the new CoA router and this proxy may have to go through one or more transit domains and experience significant increase in delay (although in terms of geographical distance the change may not be as significant). In this case, it may be important to reduce the latency by assigning the client to a closer (in terms of network delay) proxy such as the proxy in stub domain 2. The key issue is that a change of stub domain could happen as a result of moderate movement but may lead to significant increase in network delay from the wireless device to its proxy.

In our simulation experiments we use a Transit-Stub topology [6] to simulate a two-layer hierarchical topology. The network consists of six transit domains, each with an average of 10 routers. Each transit router is connected to an average of 3 stub domains, and each stub domain consists of 8 routers. Routers at any of the transit or stub domains have an average of 3 physical links to the network. The network is assumed to represent a 5000 by 5000 km geographical area. Note that we are not implying that different domains are owned by different network providers. A hierarchical infrastructure is common for large scale carriers that operate across a vast geographical area (such as USA or Australia).

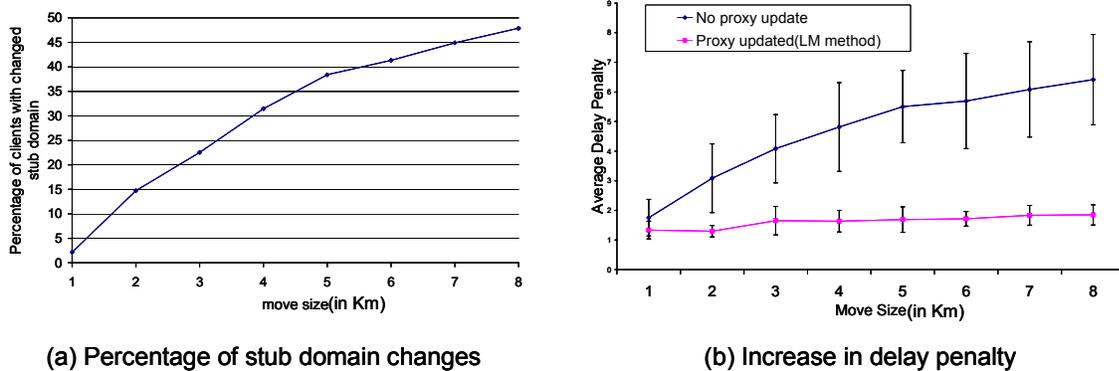


Figure 3: Effect of mobility on delay penalty

In Figure 3-(a) the percentage of moves that lead to a change of stub domain is shown for various ranges of movement from 1 to 8 Km. As can be seen, the chances of changing stub domain would increase as we move further from previous location. However, compared to the overall size of the network, relatively small moves could lead to significant probability of stub domain change. Figure 3-(b) shows the impact of movement size on the average Delay Penalty ratio between the wireless node and its assigned proxy. The Delay Penalty ratio is defined as the ratio of network delay from the wireless node to its previous proxy over this delay to the closest proxy. As shown in the Figure, without a proxy update mechanism, the latency perceived by the immersive communication service can significantly increase. Since all immersive communication flows to/from the wireless node are sent/received through the proxy, the increased delay penalty also implies wastage of network resources by using a longer than necessary path. For comparison, the delay penalty after a proxy update using the Landmark method (to be described later) is also included.

Motivation for Proxy Update

We are assuming that Layer 2 and 3 handover mechanisms are operational. Consequently the immersive communication session is continuing without interruption. However, in some cases, the latency perceived by the user may increase significantly if the proxy is not updated. Summarizing the above observations, there are three possible cases:

- 1- The range of movement of the wireless node is small and Layer 2 handover mechanisms are sufficient to maintain connectivity to the CoA router. In this case, there is no change in the CoA router and no proxy update would be needed.
- 2- The wireless node moves to such an extent that the CoA router changes and a layer 3 handover is triggered. However, the new CoA router is in the same stub domain as the previous CoA router. If there is only one proxy within this domain, a proxy update is

unlikely to be required. However, if the stub domain covers a vast geographical area and contains multiple proxies (in other words, the network topology is more or less flat in this region), an update may still improve the service delay.

- 3- As in case 2 above, but the new CoA router resides in a different stub domain. In this case, it is highly probable that a proxy update would be beneficial.

In short, a proxy update mechanism is triggered only after a change of CoA router and provided the application latency increases significantly.

The following observations are relevant to determine the most suitable proxy when an update is required:

- 1- There is only a very loose relationship between geographical and network proximity. This is particularly true for a hierarchical network topology as shown in Figure 2 where small changes in geographical proximity may translate to large variations in network delay. Nevertheless, it is quite likely that the optimal proxy (in terms of network latency) is not too far away.
- 2- We do not wish to burden the wireless node to carry out an exhaustive search for finding the optimal proxy (for example conducting a statistically reliable set of ping time measurements to all proxies after each move).
- 3- The wireless link (and in particular its MAC layer) could add significant jitter to delay measurements carried out by the wireless devices themselves.

Both observations 2 and 3 above suggest that we need to develop some form of support by the fixed infrastructure for the proxy update mechanism that requires minimal functions from the client.

Proxy Update Mechanism

In this article we propose a mechanism to identify the closest proxy by providing a Proxy Location Register (PLR) facility and a set of known landmarks [7]. The steps in identifying the nearest proxy is summarised below:

- Client notices a change in its CoA and therefore conducts a measurement of its round trip time (RTT) from the landmarks. The client sends its new care of address and the results of its delay measurements from landmarks (and possibly its derived coordinates in the Coordinate method to be described later) to the PLR.
- If the PLR already knows the closest proxy to this CoA router, it informs the client of the new proxy and the procedure ends.
- Otherwise, the PLR determines the closest proxy by the following steps:
 - PLR creates a set of what it considers to be the closest proxies to the CoA router as potential candidates. In this article, we compare three different methods for creating this *candidate set* and present comparative results on the effectiveness of these.
 - The PLR sends a request message to each of the proxies in the candidate set to measure their RTT from the new CoA router. These measurements are then returned by the candidates to the PLR and may be cached for future use.

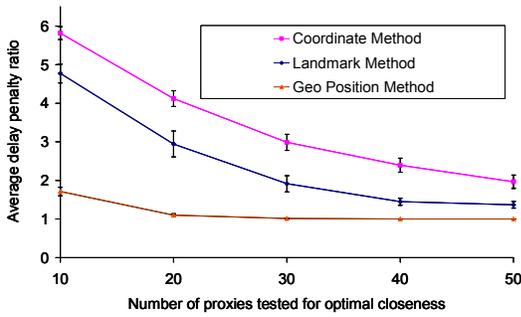
- The PLR determines the closest proxy for the client by selecting the candidate with minimum delay from the CoA router and informs the new proxy and its newly associated client about the update.

Note that the above procedure can also be used at the time when a new client joins the immersive communication service and repeated thereafter in response to mobility.

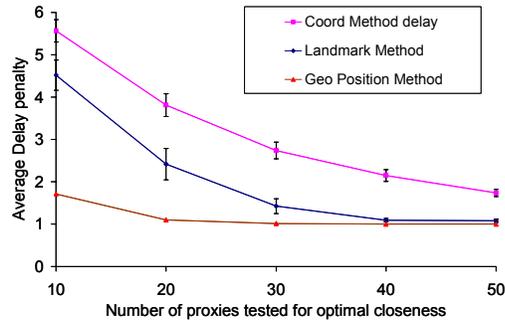
The key step in the above procedure is producing the candidate set of closest proxies based on the client's CoA and its measured delays from the landmarks. Here, we compare three possible methods for this purpose:

- 1- **Coordinate Method:** This method is based on modeling the network by a multidimensional geometric space where measured delay between any two nodes is assumed to be proportional to the distance between those two in the geometrical space [8]. The Proxy Location Register in this case has the coordinates of all proxies. Each host (whether proxy or client) based on its RTT measurement from landmarks and coordinates of landmarks finds the optimal coordinates for itself in this space such that the distances in the same space match the measured delays from landmarks as closely as possible. Due to approximation, it is usually not possible to correctly determine topologically closest proxy to a client in the network. Hence, the PLR will choose a fixed number of closest proxies in the geometrical space as the candidate set for further measurement of their delay from the CoA router.
- 2- **Landmark Closeness Order:** The second method is based on comparing the closeness order of proxies and the client to a set of well known Landmarks [7]. The PLR will then choose the candidate proxies by selecting those proxies with minimum distance in terms of their landmark closeness order. The set of determined proxies is then likely to include the closest proxy to the client. Landmark proximity order is most effective when RTT measurements from landmarks are rather accurate. For wireless hosts, the inaccuracy in RTT measurements may affect the proximity order of landmarks. We have therefore have devised a method for derivation of distances between any two hosts when proximity orders are not accurate.
- 3- **Geographical Position:** The third method is based on knowing exact geographical position of the client and proxies. In this method the candidate set is comprised of a number of geographically closest proxies to the client, which is expected to include the topologically closest proxy as well. This method can only be used if the client knows its geographical position, for example using GPS or by receiving its approximate geographical location from its CoA router. The geographical locations of proxies are also assumed to be known.

Figure 4 shows the delay penalty ratio after the proxy update is completed compared to the optimal proxy (if it could be found). Recall that in all three methods, the candidate proxies are requested to measure their RTT from the CoA router. The proxy having the minimum delay is then assigned to the client. It is possible that the optimal proxy is not within this candidate set. In order to increase the probability of finding the closest proxy, the number of proxies in the candidate set has been increased from 10 to 50 on the horizontal access (i.e. from 5.5% to 27.7% of all proxies in the simulated network model). Clearly, having a large number of proxies in the candidate set increases the accuracy but also raises the computation and bandwidth overhead and delay associated with proxy handover.



(a) Wireless delay variation of 20%



(b) Wireless delay variation of 5%

Figure 4: Average Delay Penalty with respect to size of the candidate set

As shown in this Figure, by increasing the size of the candidate set the delay penalty ratio decreases. In other words, the likelihood of finding a better proxy increases. The geographical position method can almost always find closest proxy for a candidate set size of 20 or more (11% of all proxies). The Landmark Closeness Order method and Coordinate method performance is inferior but for sufficiently high values same average delay penalty can be achieved.

For a wireless link, contention in accessing the shared wireless channel could cause significant variations over the propagation delay during RTT measurements. We have changed the delay variation introduced by the wireless link from 5% to 20% of the propagation delay to see the impact of this inaccuracy on the performance of the above methods. Clearly, the third method is not sensitive to inaccuracies in delay measurements for the client to the landmarks but the performance of other two methods is improved with lower delay variation values in the wireless link. However, the Landmark technique always outperforms the Coordinate method.

Updating Multicast Trees After a Proxy Update

Ideally, the proxy update in response to mobility should be seamless and without any disruption to the service. The main purpose of update is to improve delay performance and cost of delivery. It is therefore important to reconfigure multicast trees affected by this update as quickly as possible but without disruption. To this end, it is proposed here to have another facility referred to as Client Proxy Association (CPA) server. (This server may be running on the same hardware as the Proxy Location Register if appropriate.)

The CPA server maintains a list of every client/avatar and their assigned proxies. After each proxy update is completed by the Proxy Location Register, the PLR will update the CPA entry with the new proxy. All proxies who are the root of a multicast tree are required to consult CPA in constructing or reconfiguring their tree as well as on regular time intervals. For all multicast trees that the new proxy is a leaf, the transitions will happen gradually. Meanwhile the old proxy should forward these flows to the new proxy if needed. For the multicast trees that the new proxy is the root, the old and new proxies should coordinate the transfer to the new root.

4- CONCLUSIONS

Immersive communications is likely to form the basis of many future services for collaborative work, education and play. Access to these services with mobile devices will be of significant commercial interest. It is important to design a system architecture that can cater for both wired and wireless access. In this article, we have proposed an architecture that can achieve this goal. The distributed proxy model is suitable for both fixed and mobile clients and reduces the required functionality performed by the wireless nodes. We have also proposed possible methods for updating proxies in response to mobility. The role of the wireless device in managing mobility is still minimal and the impact on the wireless link capacity is negligible.

REFERENCES

- [1] DFC Intelligence, "The Online Game Market 2004", August 2004, California, USA.
- [2] Dowlatshahi, M., Safaei, F., "Audio Distribution and Rendering in Networked Games", proceedings of Australian Telecommunication Networks and Applications Conference ATNAC, Sydney, Australia, 8-10 December 2004.
- [3] Boustead, P., Safaei, F., Dowlatshahi, M., "DICE: Internet Delivery of Immersive Voice Communication to Crowded Virtual Spaces", to appear in the Proceedings of IEEE International Conference on Virtual Reality (VR 2005), Bonn, Germany, March 2005.
- [4] Dowlatshahi, M., Safaei, F., "A Recursive Overlay Multicast Algorithm for Distribution of Audio Streams in Networked Games", proceedings of the IEEE International Conference on Networks (ICON), November 16th-19th, 2004, Singapore.
- [5] C.E. Perkins 'Mobile IP' IEEE Communications Magazine • May 2002
- [6] K. Calvert, E. Zegura, and S. Bhattacharjee, "How to Model an Internetwork," in *Proc. IEEE Infocom*, 1996.
- [7] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection" In *INFOCOM'02*, New York, 2002. IEEE.
- [8] T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches", *Infocom 2002*, P170-179