

## Research Article

# System Architecture and Mobility Management for Mobile Immersive Communications

Mehran Dowlatshahi and Farzad Safaei

*Telecommunications and Information Technology Research Institute, University of Wollongong,  
Level 1, Building 4, Northfields Avenue, Wollongong, NSW 2522, Australia*

Received 29 October 2006; Accepted 19 December 2006

Recommended by Guobin (Jacky) Shen

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.

Copyright © 2007 M. Dowlatshahi and F. Safaei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

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 (VoIP) telephony, are based on a single point-to-point communication channel between the end users. Apart from increasing the wireless access bandwidth, a key technical challenge for these services is to develop fast hand-off schemes at layer 2 and layer 3 to enable uninterrupted voice and video communication in mobile scenarios [1].

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 (or distributed) virtual environments (NVEs). NVEs 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 pop-

ularity of networked virtual environments (NVE) 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 [2].

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 characterized 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 overlay multicast flows from the source to others within the area of interest (current IP-based multicast mechanisms due to lack of a fast dynamic reconfiguration capability does not seem

to be appropriate for multicasting of media streams to frequently changing groups of clients). 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 overlay multicast trees must undergo change. The immersive communications, therefore, is characterized by a large number of overlay multicast flows that are subject to rapid reconfiguration.

Wireless access to multimedia immersive communications presents additional challenges.

- (A) The *access bandwidth* is more likely to be a bottleneck as the wireless link capacity is unlikely to match that of wired access in near future. This would be especially important for the downstream bandwidth where the required number of streams received would depend on how “crowded” the participant’s surrounding environment is.
- (B) The *energy resources in wireless devices* for the battery operated mobile devices are likely to be scarce. Hence, it would be desirable to minimize transmission by mobile clients.
- (C) The *mobility* of wireless devices across networks will create new challenges for mobility management and as will be shown require new functions in addition to layer 2 and layer 3 handovers to control latency.

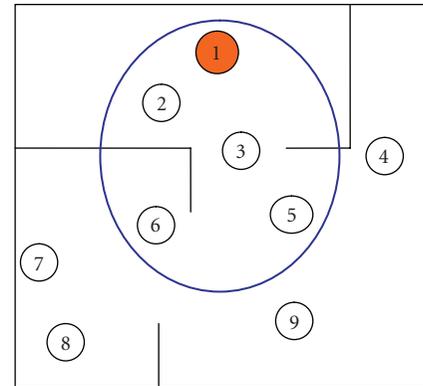
One approach is to tackle these problems head-on by developing higher speed wireless links and more powerful batteries. The second approach, which is the subject of this article, is to perform certain functions in the fixed infrastructure to relieve the wireless devices from excessive transmissions of media streams. Naturally, these two approaches are not mutually exclusive and can work together to offer the best outcome based on a given wireless technology.

In our earlier works [3, 4] a distributed proxy model for media streaming to fixed clients of virtual environments has been proposed. There, in order to minimize end-to-end delay, every fixed client is assigned to its topologically closest proxy. Here the same distributed proxy architecture for media streaming to mobile nodes has been adopted. Network layer mobility management (e.g., [1, 5]) may transparently change a mobile client’s network at any instant. In this paper it will be shown that in a distributed proxy model mobility management at the network layer without mobility management at the application layer is likely to deteriorate end-to-end delay performance. Main focus of this paper is therefore *application layer* mobility management techniques for the distributed proxy model.

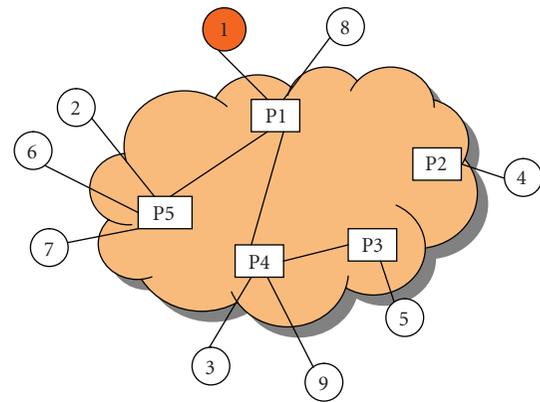
The rest of this paper is organized as follows. Section 2 describes our proposed system architecture. Section 3 provides a possible solution for managing physical mobility and simulation results on its effectiveness. Section 4 presents the concluding remarks.

## 2. SYSTEM ARCHITECTURE

A conceptually simple model for wireless immersive communications is to use a peer-to-peer model for transmission of



(a) Avatar locations in virtual world



(b) Clients and their proxy servers

FIGURE 1: Distributed proxy architecture for immersive communication.

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 overlay multicast to this subset.

While simple, this model has some drawbacks. First, the downstream wireless link is a shared media and would limit the number of flows that can be received by each participant. Second, in addition to their locally captured streams the clients may have to participate in forwarding (transmission) received media stream(s) from other clients towards the clients that need them.

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 overlay 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 analyzing its loudness and the characteristics of the environment (e.g., 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 overlay 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 voices. Consequently, at each instant of time, there will be  $N$  active overlay multicast trees between the proxy servers, where  $N$  is the number of talking avatars. By talking avatars we mean those who have subscribed to the audio service and have active voice signals at this moment. On the downstream side, the proxy should send the audio streams of relevance to its connected clients.

### 2.1. Coping with limited wireless link capacity

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 is equal to  $Kr$  bits per second (bps) for a given wireless device, 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 for ensuring 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.

In the first case, the proxy merely performs a simple "filter and forward" operation together with silent suppression at the clients. An important characteristic of immersive communications is the fact that usually only a limited number of talking avatars can be heard in each virtual scene (avatars in the same room, vehicle, etc.) and therefore each avatar will need to receive only certain number of audio streams at any instant. For constant bit rate streams, this number may still exceed the wireless access capacity. However, if voices are modeled 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. The client will then only send the "active" audio packets to its assigned proxy. The proxy multicasts these voice packets to all other proxies that require the audio signal from this avatar for any of their respective clients. On the receiving side, each proxy receives a number of voice streams from its attached clients and the other proxies. The proxy has information on the access bandwidth lim-

itation of its clients. It uses a priority-based ordered list of all avatars within the hearing range of each client and sends (at most)  $Kr$  bps of the active streams to each 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 the talk spurt periods of the closest avatars. In [6] 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 4 or more avatars have simultaneous talk spurts. Whether this loss is subjectively significant requires further study. The impact on listener, who is already receiving three active audio streams could be small. According to the same analytical results, more than 80% of the talk spurts of the fifth talking avatar, if any, will also be heard.

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 "center 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 center 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 (see [7] for details).

### 2.2. Coping with mobility in the virtual world

In the distributed proxy architecture, the wireless devices need not participate in formation and reconfiguration of overlay multicast trees. Instead, each proxy will become the root of overlay multicast for all of the streams sourced by its attached clients and will have to join all those overlay 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 overlay multicasts has another important advantage: the reconfiguration of overlay multicast trees due to movements in the virtual world happens 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 the movement of both the speaker and the listeners. This, in turn, may lead to a new multicast tree if any of the proxy leaves 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 recent years many techniques for construction of overlay multicast trees and networks have been proposed [8–10]. In [3, 4] we have proposed an algorithm for construction of overlay 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 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.

Using simulation we have investigated the effect of physical movements on round trip time delay of client to proxy nodes. In our simulation experiments we use a transit-stub topology [11] 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. Each stub domain is assumed to have a single proxy attached to one of its routers. 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

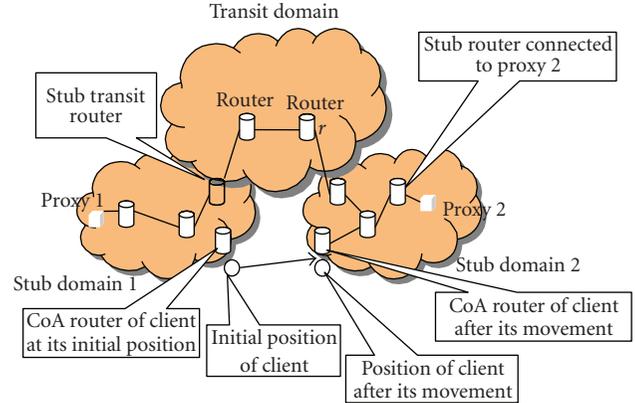


FIGURE 2: Movement of wireless node from one stub domain to another.

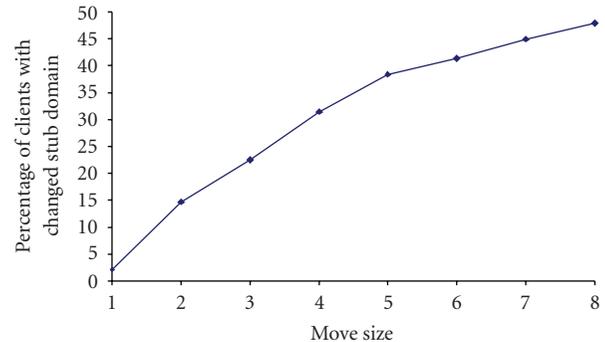


FIGURE 3: Percentage of stub domain changes versus move size.

scale carriers that operate across a vast geographical area (such as USA or Australia).

In Figure 3 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 a previous location. However, compared to the overall size of the network, relatively small moves could lead to significant probability of stub domain change.

Figure 4 shows the impact of movement size on the average delay penalty ratio between the wireless node and its assigned proxy. Delay penalty ratio is defined as the ratio of network delay from the wireless node to its previous proxy over this delay to the topologically closest proxy, see (1):

$$\begin{aligned} \text{Delay Penalty Ratio} \\ = \text{Delay (from previous proxy)} / \text{Delay (from closest proxy)}. \end{aligned} \quad (1)$$

As shown in the Figure 4, 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 its assigned 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.

Summarizing the above observations, there are three possible cases: (A) 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; (B) 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 stub 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; (C) as in case (B) above, but the new CoA router resides in a different stub domain. In this case, (assuming that each stub domain has its own proxy) it is highly probable that a proxy update would be beneficial.

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 (e.g., 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) 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.

### 3.1. 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. 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 network 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

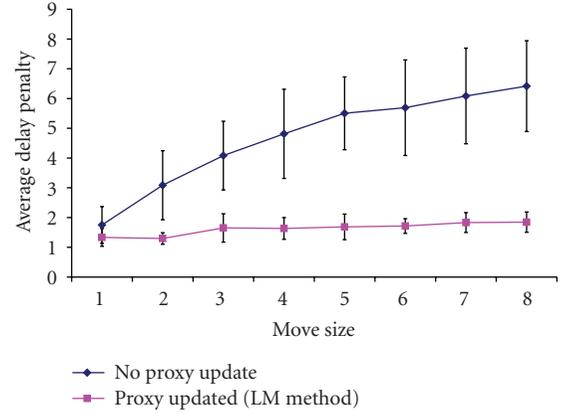


FIGURE 4: Average delay penalty increase versus move size (in kilometers).

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 equal to the distance between those two in the network geometric space [12]. 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 [13]. The PLR will then choose the candidate proxies by selecting those proxies with minimum distance (from client) 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

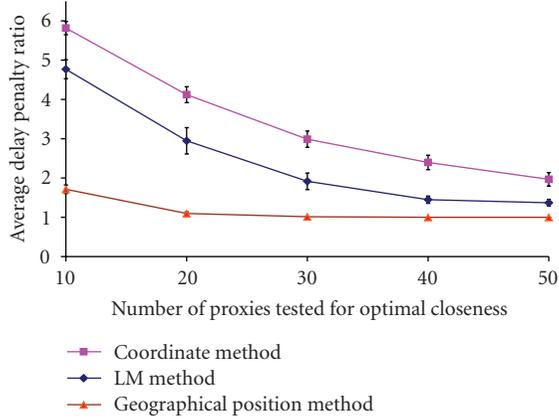


FIGURE 5: Average delay penalty with respect to the size of the candidate set (20% delay measurement error).

proximity order of landmarks. Distances between any two hosts when proximity orders are not accurate are calculated using weighted landmark closeness order distance as in (2):

$$D(N_1, N_2) = \sum_{i=0}^{LMN-1} (LMN - i) \times d_i(N_1, N_2). \quad (2)$$

Here  $d_i(N_1, N_2) = 1$  if  $i$ th closest landmarks of nodes  $N_1$  and  $N_2$  are different, and  $d_i(N_1, N_2) = 0$  if  $N_1$  and  $N_2$  have the same  $i$ th closest landmark. In (2)  $LMN$  is the number of landmarks and  $(LMN - i)$  is the weight of  $i$ th closest landmark. According to (2) closest landmark has the largest weight.

(3) Geographical position: the third method is based on knowing the 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 5 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 distance 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. Delay measurement between a mobile client and any other node is prone to error. A random error is therefore added to measured delay between

each mobile client and each landmark node. Delay measurement error reduces mobile nodes' ability to correctly determine their landmark proximity order and network geometric coordinates. In Figure 5 mobile nodes are assumed to have an average of 20 percent delay measurement error.

As shown in Figure 5, 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 performance of landmark closeness order method and coordinate method are inferior but for sufficiently large set sizes the same average delay penalty can be achieved.

### 3.2. 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 proxy location register completes proxy update of a mobile client, the PLR will update the client's entry in the CPA server with the new proxy. All proxies are required to consult CPA in constructing or reconfiguring their multicast trees after new client allocations as well as on regular time intervals.

After selection of a new proxy, old proxy of a moved client should forward required flows (of the moved client) to the new proxy of the client until a handover from the old proxy to the new proxy is complete. This policy assures uninterrupted forwarding of the media streams to the moving clients.

Handover delay for the proposed architecture comprises of the following components:

- (i) delay from client to PLR: this is the delay for the mobile client to send a request for allocation of a new proxy after it detects a change in its CoA node;
- (ii) maximum RTT delay from PLR to nominated proxies: this is the delay for the PLR to send a delay measurement request and receive a reply from all nominated proxies;
- (iii) maximum RTT delay from nominated proxies to the new CoA of the mobile client (for finding closest proxy to CoA);
- (iv) delay from PLR to the newly allocated proxy of the mobile client;
- (v) delay from new proxy to the mobile client;
- (vi) delay from new proxy to receiver proxies of the mobile client's media stream(s): this is the delay for new client's proxy to update all receiver proxies that need client's media stream.

Figure 6 shows the cumulative distribution of the handover delay after a physical move. For this measurement we

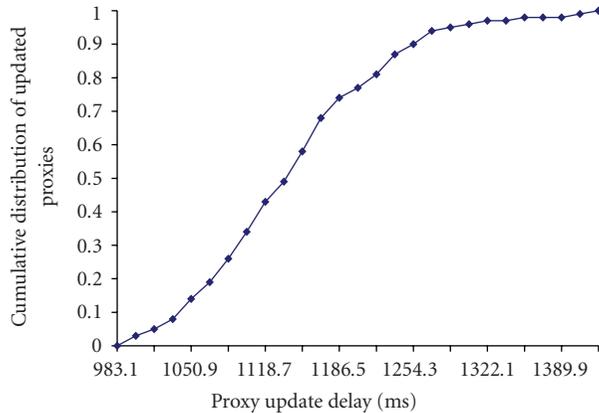


FIGURE 6: Cumulative distribution of handover time (update time) of proxies.

have scaled up average delay of the simulated network to that of the real networks. As can be seen, in majority of cases the handover is completed in less than 1.3 seconds. This may initially seem large, but it must be noted that the voice communication has not been interrupted during this period as the old proxy is still sending the voice streams to the client. This handover is to connect to a new proxy to improve delay performance of interactive communication and its perceptual impact may not be significant.

#### 4. CONCLUSIONS

Immersive communication 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 therefore the impact on the wireless link usage and power consumption remains negligible.

#### ACKNOWLEDGMENT

This work was funded by the Cooperative Research Centre (CRC) for Smart Internet Technology, Australia.

#### REFERENCES

- [1] D. Johnson, C. Perkins, and J. Arrko, "Mobility Support in IPv6," RFC 3775, June 2004.
- [2] Online game market, DFC Intelligence, "The Online Game Market 2004," August 2004, California, USA.
- [3] M. Dowlatshahi and F. Safaei, "A recursive overlay multicast algorithm for distribution of audio streams in networked games," in *Proceedings of the IEEE International Conference on Networks (ICON '04)*, Singapore, November 16th–19th, 2004.
- [4] M. Dowlatshahi and F. Safaei, "Overlay multicasting of real-time streams in virtual environments," in *Proceedings of IEEE Globecom*, San Francisco, 2006.
- [5] C. Perkins, "IP Mobility Support," IETF RFC 2002, October 1996.
- [6] M. Dowlatshahi and F. Safaei, "Audio distribution and rendering in network games," in *Proceedings of ATNAC*, 2004.
- [7] P. Boustead, F. Safaei, and M. Dowlatshahi, "DICE: Internet delivery of immersive voice communication for crowded virtual spaces," in *Proceedings of IEEE Virtual Reality*, 2005.
- [8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM SIGCOMM*, 2002.
- [9] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicasting with Delaunay triangulation overlays," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, 2002.
- [10] E. Brosh and Y. Shavitt, "Approximation and heuristic algorithms for minimum delay application-layer multicast trees," in *Proceedings of IEEE INFOCOM*, 2004.
- [11] K. Calvert, E. Zegura, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of IEEE INFOCOM*, 1996.
- [12] T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *INFOCOM*, pp. 170–179, 2002.
- [13] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *INFOCOM '02*, New York, 2002, IEEE.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

