

2004

A recursive overlay multicast algorithm for distribution of audio streams in networked games

M. Dowlatshahi

University of Wollongong, mehran@uow.edu.au

F. Safaei

University of Wollongong, farzad@uow.edu.au

Publication Details

This paper originally appeared as: Dowlatshahi M and Safaei, F, A recursive overlay multicast algorithm for distribution of audio streams in networked games, Proceedings of the 12th IEEE International Conference on Networks, 16-19 November 2004, vol 2, 689-693. Copyright IEEE 2004.

A recursive overlay multicast algorithm for distribution of audio streams in networked games

Abstract

This paper aims to provide a fast multicast algorithm among a set of distributed proxies to enable the creation of realistic audio for networked games. We assume that each proxy is responsible for a group of clients. The proxy receives the audio packets from its clients and multicasts this audio to every other proxy that requires this audio for any of its attached clients. In crowded virtual environments, the number of multicast trees could be quite large and these undergoes continuous change as the avatars move in and out of each others hearing range. To reduce the effect of overhead traffic caused by a large number of unicast flows, we propose the use of (i) coordinate-based clustering of the servers to reduce the number of packet replications required at each server; and (ii) a recursive multicast tree construction with minimal control overhead and reconfiguration complexity. We present simulation results for a crowded networked environment and evaluate the effectiveness of our algorithm in terms of path delay in comparison with the shortest unicast delay paths.

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper originally appeared as: Dowlatshahi M and Safaei, F, A recursive overlay multicast algorithm for distribution of audio streams in networked games, Proceedings of the 12th IEEE International Conference on Networks, 16-19 November 2004, vol 2, 689-693. Copyright IEEE 2004.

A Recursive Overlay Multicast Algorithm for Distribution of Audio Streams in Networked Games

Mehran Dowlatshahi, Farzad Safaei
Smart Internet Technology CRC
University of Wollongong, Australia

Abstract - This paper aims to provide a fast multicast algorithm among a set of distributed proxies to enable creation of realistic audio for networked games. We assume that each proxy is responsible for a group of clients. The proxy will receive the audio packets from its clients and will multicast this audio to every other proxy that requires this audio for any of its attached clients. In crowded virtual environments, the number of multicast trees could be quite large and these will undergo continuous change as the avatars move in and out of each others hearing range.

To reduce the effect of overhead traffic caused by a large number of unicast flows, we propose the use of (i) coordinate-based clustering of the servers to reduce the number of packet replications required at each server; and (ii) a recursive multicast tree construction with minimal control overhead and reconfiguration complexity. We present simulation results for a crowded networked environment and evaluate the effectiveness of our algorithm in terms of path delay in comparison with the shortest unicast delay paths.

1. INTRODUCTION

Creation of realistic audio scenes for the avatars in a networked virtual environment presents significant challenges with respect to scalability of network and server infrastructure. In [1], a comparison of several delivery architectures for such a service is presented. One particularly attractive option is to use a set of distributed servers – referred to as proxies – over the Internet to aid in the delivery of audio streams to clients. Each proxy (which typically may reside at the Point of Presence of an ISP) is responsible for its own group of clients. When a given participant is talking, the audio packets are captured by the client and sent to its respective proxy. The proxy will then identify all other avatars that might require this audio stream and will forward the audio packets to their proxies either as multicast or multiple unicast flows. After receiving all the audio streams of interest, each proxy will have to send either a subset or a partially mixed set of audio streams to every one of its clients so that the access bandwidth limit is not exceeded. The clients can then render the audio scene based on the information received.

This architecture is attractive because it provides a natural partitioning of tasks among the ISP owned servers. As new participants are added to the networked environment, their

ISP is responsible to provide computational resources in the proxy to cater for their audio needs. The ISP is also able to add security, anonymity and billing functions in each proxy. However, the distributed proxies need to maintain and dynamically reconfigure a large number of multicast trees between themselves if the service is to be scaled to large crowded spaces. The number and size of these multicast trees depend on the density of the virtual world, perceptually acceptable hearing range for the avatars, and the game's genre (how critical the audio communication is and how much it is being used by participants).

IP-based multicast capability is now available in many routers. However, a number of difficulties have prevented ISPs from deploying multicast enabled networks widely. Application layer multicast has become the focus of a number of research projects and a number of different schemes for construction of overlay multicast trees [2, 3, 4] have been proposed. Main goals of these schemes are: (a) - To limit the degree or number of outgoing streams from each server, (b) - To optimize the depth of overlay multicast trees and (c) - To reduce distance penalty as defined later in this paper. Servers' degrees must be limited according to both the outgoing capacity of the servers' links as well as their duplication capability.

In [3], overlay topologies for application multicast are categorized into 3 groups. First group [5] is suitable for construction of single tree topology. The second group [6] constructs spanning trees embedded in mesh graphs and requires running of a multicast routing protocol. The third group [7] builds overlay networks based on logical addresses assigned to the member nodes.

Here we propose a scheme for construction of multicast trees that is scalable to a large number of highly dynamic trees. Our approach assumes knowledge of the spatial coordinates of the servers in the Internet. In [8], a relatively accurate approach for predicting network distances of nodes based on measured bandwidth of nodes and measured round trip delay (RTD) from a set of landmarks have been devised and developed. It is shown that in many cases, the accuracy is around 90%.

In our proposed approach, the target servers are clustered based on their angular separation by the root of the multicast

tree. The number of clusters is at most equal to the number of duplications that the root server can perform. In the construction phase the root server will then send a list of destination servers in each cluster to the nearest server in that cluster. The nearest servers, in turn, repeat the same clustering method if needed for these other destinations and send to their nearest server in each cluster. The process will continue until all the destinations have received the packet.

The rest of this paper is organized as follows: In Section 2 we provide a description of the distributed proxy model and the need for maintaining a large number of dynamic multicast trees for audio scene creation. In Section 3, our proposed multicast approach is described and its strengths and weaknesses discussed. In Section 4, we present some of the simulation results and discuss their implications. Conclusions and future work are presented in Section 5.

2. DISTRIBUTED PROXY ARCHITECTURE AND MULTICAST TREES

Figure 1 shows the distributed proxy architecture for creation of immersive audio scenes for crowded virtual spaces. Every client is connected to a proxy server for the duration of service. This connection can be assumed to be pre-determined – e.g. based on geographical proximity or the preferred ISP.

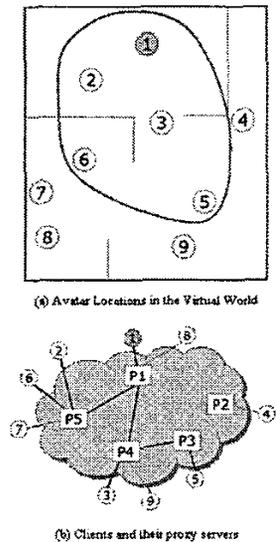


Figure 1 distributed proxy architecture for creation of immersive audio scenes

On the upstream side, the client will send its audio packets to its proxy. It is the responsibility of the proxy to forward/multicast this audio 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 audio 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 (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 audio 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 audio. Consequently, at each instant of time, there will be N active 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 activated the service at this moment.

Figure 1 also demonstrates another important property of the multicasts in the distributed proxy model. The audio stream received by P5 is needed by two of its attached clients (2 and 6) while the other two proxies (P3 and P4) need the stream for only one of their clients. It should be clear therefore that if the majority of avatars in the audible range of a given audio signal are in a close physical proximity (that is, connected to the same proxy) then the size of the multicast trees between the proxy servers become smaller. (Here we define the size of multicast tree by the number of proxy leaf nodes.) In [1] a 'correlation' parameter has been defined to model the relationship between the distribution of avatars in the virtual world and the distribution of their corresponding clients in the physical world. Figure 2 shows the cumulative probability distribution of the size of multicast trees when the correlation parameter is zero and avatars are uniformly distributed in the virtual space. As can be seen for this case most of the multicast tree sizes are between 9 and 14.

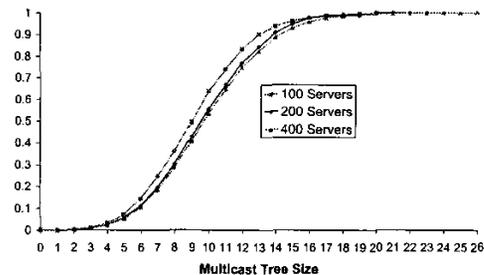


Figure 2 Multicast Tree size when a server is allowed to send up to 4 copies of each packet

2.1 Application Dynamics

The multicast trees discussed above are not static and will have to change in response to the application dynamics. 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 leaves are different.

Figure 3 shows the cumulative distribution of changes in the set of multicast trees after a given period of time and based on a random waypoint motion applied to all avatars. Percentage of changes due to movement of avatars for each server is found by comparing the set of required audio streams before and after movements of the avatars for each server. As can be seen more than 90% of servers undergo more than 60 percent change in their set of required audio streams. These results are based on a zero correlation in distributions in the virtual and physical worlds. If the correlation is high, the multicast tree changes will be less.

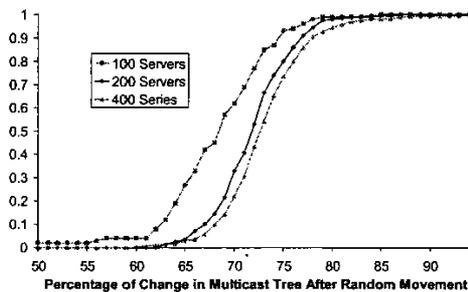


Figure 3 Percentage of change in multicast tree after a random movement of all the avatars in the virtual environment

3. APPLICATION LAYER MULTICAST

In the proposed approach only proxies participate in overlay multicasting and clients' hosts receive (mixed or non-mixed) audio streams directly from their allocated proxies.

Each client directly sends position, orientation and talking/silence status of its associated avatar to its allocated proxy, (called *local proxy*). Assigned clients to a proxy and their associated avatars are called local clients and local avatars of that proxy. Also each proxy receives information about other avatars associated with the clients of other proxies, (called *non-local avatars, non-local clients*).

Each proxy also receives necessary information to be able to determine proxies of non-local avatars. Dissemination of information about non-local avatars and their associated proxies are outside the scope of this paper.

For the sake of clarity in explanations, in this study it is assumed that the virtual world is partitioned into grid zones called *cells*. By suitably choosing the size of these cells, it would be safe to assume that avatars can only hear talking avatars within their *home cell*, where home cell of an avatar is the grid cell that the avatar resides. In essence, each cell determines the hearing range of its avatars. This assumption only reduces the average number of listeners for talking

avatars and as a result reduces the average multicast tree sizes.

Each proxy can construct audio multicast trees for each of its talking avatars based on the above information as described before.

3.1 Multicast Tree Construction

The main goals in designing the algorithm for construction of multicast trees are: (a)- Limited duplications: Because of limited bandwidth of the proxies the algorithm must be able to construct trees with limited number of copies determined independently for each proxy; (b)- Small computational overhead: In a large virtual environment with many talking avatars construction and maintaining of a large number of small multicast trees will be necessary; (c)- Small distance or delay penalty: Distance penalty (DP) is the extra traveled distance due to deviation from the direct line connecting the source and destination proxies when forwarding a packet through an overlay multicast tree; (d)- Small depth trees: A small multicast tree depth reduces number of middle proxies between any leaf node proxy and multicast root proxy and therefore decreases end-to-end delay and is desired.

In the following different stages of the proposed clustering algorithm for construction of a multicast tree are explained and later some of the advantages of the proposed algorithm reviewed.

In the first stage, the local proxy of the intended talking avatar has to determine all the proxies that require the audio stream sourced from its clients. We refer to this as the source proxy. Each proxy is assumed to have the spatial coordinate of other proxies based on the model developed in [8].

In the second stage, the source proxy will cluster the recipient proxies into K segment based on angular separation. This is referred to as *Angular clustering* where K is the maximum allowed duplication for the source proxy. Each cluster is formed by two radial rays emanated from the proxy towards the edges of the proxies' network plane and contains a subset of proxies requiring the multicast, see Figure 4. Angular clustering is used to reduce the distance penalty and is explained in the next section. Maximum duplication limit for proxy P1 is 3, and C1, C2 and C3 are its angular clusters created based on the coordinates of all the recipient proxies only.

In the third stage, source proxy determines closest proxy host (minimum Euclidian distance) in each cluster and sends a list of all proxies in that cluster to the determined closest proxy. In figure 3 proxy P2 in cluster 1 is closest proxy and P1 sends a list containing P2, P3 and P4 to proxy P2.

After receiving a list of proxies, recipient proxy repeats steps 2 and 3. Steps 2 to 3 are executed while two or more are in the received list. The proxy that receives a list of size one will assume itself as a leaf node. It should be noticed that list of proxies is actually all the underlying proxies in the multicast

tree. Here proxy P2 receives a list of size 3. After deleting itself from the list because of duplication limit of 1, P2 sends a list of proxies P3 and P4 to its closest in the list which is proxy P3. It is obvious that this approach leads to loop free overlay multicast trees.

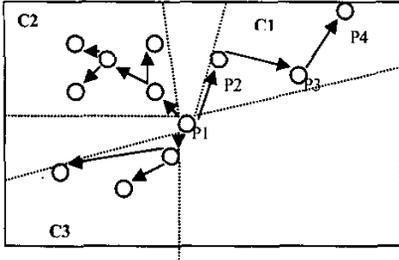


Figure 4 Recursive constructions of multicast trees

Limited number of copies, loop free construction, simplicity, small processing load, reduction of travel distance to acceptable level as proved by simulation and a depth proportional to $\log(N)$ are main advantages of the proposed algorithm. In this algorithm every middle proxy can independently and dynamically change its maximum number of clusters to avoid congestion on its network links.

The main disadvantages of this algorithm are its dependence on accuracy and consistency of the coordinates and assumption of triangular inequality in the network plane.

3.2 Angular Clustering

As previously mentioned *angular clustering* algorithm puts destination proxies into K angular clusters. Each cluster is formed by two radial rays emanated from the intended source proxy towards the edges of the network plane and contains a subset of destination proxies, see Figure 4. The algorithm for angular clustering from the perspective of proxy i and with K angular clusters can be described as follows:

If the total number of proxies in the underlying multicast tree is less than or equal to K , then send a list of size one to each of them. Each of those proxies will then directly receive audio stream from the source proxy in the upper tier.

Otherwise, select one of the proxies randomly and choose the radial ray that emanates from proxy i and goes through this destination proxy as the reference for calculating of angular position of other destination proxies.

Calculate the *relative angle* of every other destination proxy with respect to this reference.

Order the destination proxies in ascending order of their relative angles from the reference radial ray.

Calculate *angular separation* of each two successive proxies in the angularly ordered list by subtracting their respective relative angles.

Largest angular separation is then subtracted from 360 degrees. The remainder of this subtraction is then divided by the allowed number of clusters K . The result is the maximum angular separation allowed between the first and any other proxy in its cluster.

Clustering is started from the proxy with a larger relative angle in the largest separated pair unless proxy with larger relative angle is the last in the sorted angular list. In the later case the proxy with the smaller relative angle (angle zero) in the largest separated pair will become the starting point for clustering.

Proxies are inserted in the cluster while the separation from the first in the cluster is less than the maximum found, otherwise a new cluster is started and clustering continues until all proxies are inserted.

As an example, assume that there are 10 proxies in the underlying multicast tree of proxy i . Also assume that $K=4$. We select one of these destination proxies in random and calculate the relative angles from the perspective of talking avatar's proxy i . Let the ordered list of relative angles measured in degrees be:

0, 10, 16, 48, 67, 120, 143, 170, 222, 253.

The associated proxies are named according to the same order A_0 to A_9 . The angular separation list will be:

10, 6, 32, 19, 53, 23, 27, 52, 31, 107,

Corresponding to the angular separation between:

$A_0-A_1, A_1-A_2, A_2-A_3, A_3-A_4, A_4-A_5, A_5-A_6, A_6-A_7, A_7-A_8, A_8-A_9, A_9-A_0$

The largest angular separation is 107 between A_9-A_0 and the maximum allowed angular separation between the first and any proxy in its cluster is obtained as follows: $(360 - 107)/4$ or 63.5 degrees. Since the proxy with the larger relative angle from the largest separated pair is the last in the sorted list, the proxy with smaller angle in that pair is the starting point for clustering. The four clusters are therefore found to be:

$\{A_0, A_1, A_2, A_3, A_4, A_5\} \{A_6, A_7\} \{A_8, A_9\}$

4. SIMULATION EXPERIMENTS

We have studied performance of our proposed algorithms through simulation experiments. We have assumed similar link capacities for proxies. Proxies are spread in a square plane. Clients are randomly assigned to proxies.

Avatars are uniformly spread on a grid plane and randomly associated with clients. Virtual world comprises of 70 by 70 grid cells, i.e. 4900 grid cells. 50000 clients in the network participate in the virtual world are randomly associated with 50000 avatars in the virtual environment. In all of the experiments 20 percent of the avatars are assumed to be talking, i.e. 10000 avatars are talking in all experiments. We

have assumed similar link capacities and similar maximum number of duplications per packet (i.e. $K=4$) for all proxies.

4.1 Simulation Experiments

Figure 5 shows cumulative distribution for number of incoming audio streams of audio proxies.

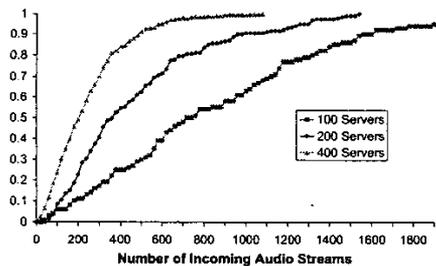


Figure 5 Cumulative distribution for number of incoming audio streams to proxies

These results can be used in scaling the link capacities of proxies to the network. As expected results show that with an increase in the number of proxies number of incoming audio streams decreases. Using simple calculations one can obtain an upper bound for average number of incoming audio streams to each proxy. With 50000 avatars in the virtual environment and 20 percent talking ratio, each proxy will receive an average of: $10000 / (\text{number of Proxies})$ audio streams from its allocated clients which we call m . Also assuming an average of $50000 / (\text{number of proxies})$ clients are allocated to each proxy, and assuming that no two avatars of the same proxy are in the same grid cell then number of incoming streams from other proxies will be on average less than $((10000 - m) / 4900) * (50000 / (\text{number of proxies}))$. By inserting different number of proxies it is easy to find an upper limit approximation for average number of incoming audio streams of each proxy. It has been observed that average number of incoming streams obtained by simulation is consistent with results of the above calculations.

Results show that cumulative distribution for number of incoming audio streams of proxies is similar to cumulative distribution for number of multicast trees in which each proxy participates. This similarity is intuitively correct because each incoming stream usually relates to a multicast tree. Only when a talking client has no non-local avatar in its home cell multicast tree is not associated with the incoming stream.

Here sensitivity of distance (or delay) penalty (DP) in percentage of the direct distance from the multicast root proxy to the proxies in the multicast tree when coordinates of the proxy nodes in the network plane are not accurate, have been obtained. For this purpose 2 set of coordinates have been assigned to each one of the proxies; A- inaccurate coordinates based on which clustering and forwarding have been performed and B- accurate coordinates based on which

traveling distance and DP have been calculated. It has been observed that when 40 percent of proxies have completely different (i.e. zero correlation) accurate and inaccurate coordinates, average DP is equal to 79% and 60% for 400 and 100 proxies respectively.

5. CLOSING REMARKS

A recursive multicast clustering approach for construction of numerous small size multicast trees has been proposed. Previous works in this area have focused in large overlay multicast trees with one or few media sources. These approaches usually require relatively large processing time and are not appropriate for virtual environments with a large number of dynamically changing multicast trees.

The proposed approach has all the necessary characteristics including support of numerous overlay multicast trees, limited duplication, manageable processing overhead, loop free overlay trees, acceptable delay penalty and small tree depths. Simulation results have also shown that even when a large percentage (e.g. 40%) of proxies have totally incorrect predicted coordinates, observed distance penalty remains acceptable.

6. ACKNOWLEDGEMENT:

This work was funded by the Co-operative Research Center (CRC) for Smart Internet Technology, Australia.

7. REFERENCES:

- [1] Boustead, P., Safaei, F., "Comparison of Delivery Architectures for Immersive Audio in Crowded Networked Games", Accepted for publication in the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video NOSSDAV, Ireland, June 16-18, 2004.
- [2] Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, Samir Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications", in *Proc. IEEE INFOCOM*, 2003.
- [3] Jörg Liebeherr, Michael Nahas, Weisheng, Si, "Application-Layer Multicasting With Delaunay Triangulation Overlays", *IEEE JSAC*, VOL. 20, NO. 8, OCTOBER 2002.
- [4] Duc A. Tran, Kien A. Hua, Tai Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming", *IEEE* 2003, 1283-1292.
- [5] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over a peer-to-peer network," Stanford Univ. Comput. Sci. Dept., Stanford, CA, Tech. Rep. 2001-30, June 2001.
- [6] Y. Chawathe, S. McCanne, and E. A. Brewer, "RMX: Reliable multicast for heterogeneous networks," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 795-804.
- [7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001, pp. 149-160.
- [8] T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches", *Infocom* 2002, P170-179.