2006

Multipoint Interactive Communication for Peer to Peer Environments

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: Multipoint Interactive Communication for Peer to Peer Environments 2006.
<https://ro.uow.edu.au/infopapers/503>

Multipoint Interactive Communication for Peer to Peer Environments

Abstract

In this paper a scheme for peer-to-peer (p2p) multiparty voice communication is proposed. The paper considers realistic constraints on the upstream and downstream bandwidth of peers and their processing capacity. We propose a novel mechanism for distributed mixing of the audio streams that can improve the scalability of this service while meeting the capacity constraints. The paper also proposes an algorithm to form a multipoint-to-multipoint spanning tree (MMST) among the peers within a communication zone. The simulation results show that the performance of algorithm in terms of delay stretch and its impact on bandwidth reduction in comparison to multiple source rooted overlay multicast trees is quite acceptable.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Dowlatshahi, M & Safaei, F, Multipoint Interactive Communication for Peer to Peer Environments, IEEE International Conference on Communications 2006, Istanbul, Turkey, June 2006, 317-322. Copyright IEEE 2006.

Multipoint Interactive Communication for Peer to Peer Environments

Mehran Dowlatshahi

Smart Internet Technology Cooperative Research Centre
University of Wollongong
Australia
mehran@uow.edu.au

Farzad Safaei

Smart Internet Technology Cooperative Research Centre
University of Wollongong
Australia
farzad@uow.edu.au

Abstract— In this paper a scheme for peer-to-peer (p2p) multiparty voice communication is proposed. The paper considers realistic constraints on the upstream and downstream bandwidth of peers and their processing capacity. We propose a novel mechanism for distributed mixing of the audio streams that can improve the scalability of this service while meeting the capacity constraints. The paper also proposes an algorithm to form a multipoint-to-multipoint spanning tree (MMST) among the peers within a communication zone. The simulation results show that the performance of algorithm in terms of delay stretch and its impact on bandwidth reduction in comparison to multiple source rooted overlay multicast trees is quite acceptable.

Keywords—peer-to-peer; distributed mixing; overlay multicast; multiparty voice over IP;

I. INTRODUCTION

Voice communication over the Internet is gaining momentum and attracting business and residential customers. The current Voice over IP services are primarily designed for one-to-one communication. Voice chat rooms and conferencing services are also on the rise; however, in majority of cases, these services are supported by conferencing servers (or bridges) to perform audio mixing operations for the users. Peer-to-peer conferencing applications are often limited to a small number of participants to meet the bandwidth and processing constraints of the peers.

Multiparty voice communication is also becoming important within the Networked Virtual Environment such as online games. In these environments, it would be desirable to allow seamless voice communication for those avatars who are within each other's hearing range. As such, the membership of multiparty communication is governed by proximity and the existence of sound barriers (such as walls and rooms) in the virtual world. There has been significant research and commercial activity in providing proximity based voice communication capability to multiplayer network games. Immersive voice communication with spatial audio has been described in [1], [2], and [3], but requires servers to perform partial audio mixing or filtering of audio packets. Many commercial games use a peer-to-peer communication mechanism, but are limited to one or very few voice channels. This limitation is, once again, due to bandwidth and processing constraints of the clients.

Recently p2p voice communication has become popular. As an example, the number of users of Skype p2p service has increased significantly [4]. In Skype's multiparty communication, the peer client with largest processing and access link capacity automatically becomes a mixing node for other peers. This mixing client then receives streams from all other clients, calculates a mixed stream for each client and sends the calculated mixed stream back to them. Obviously a mixing node must be able to accommodate all incoming and outgoing streams from other clients and therefore may face scalability issues.

We are interested to design a multiparty voice communication system that can scale to reasonable number of participants and is purely based on a peer-to-peer model. This system could be used to provide conferencing services over the Internet (for example, by joining a conference or chat room from a drop down menu or based on invitation) as well as addition of voice capability to Networked Virtual Environments (such as games or collaborative environments). In the latter category, the location of avatars in the virtual environment and the characteristics of the environment will determine the composition of the multiparty communication peers. We can assume, therefore, that the virtual environment is partitioned into a number of *communication zones* based on the audible range of speakers. In essence, the avatars within the same communication zone would form a conference or voice chat group. The information pertaining to composition of communication zones can be obtained from a central/distributed state information server or using a peer-to-peer exchange of state information depending on the design of the virtual environment. In both cases, however, we require the actual voice communication to be based on a peer-to-peer model.

The key technical challenges in developing a peer-to-peer multiparty voice communication service are bandwidth and processing limitations of peers on one hand and the communication delay associated with overlay multicast among the peers on the other hand. In this paper we propose a novel distributed audio mixing architecture that can provide all the required voice streams to a participant regardless of how crowded the conference is while meeting the bandwidth and processing constraints of the peers. This architecture is described in Section 2. In Section 3, we propose a model for

creation of a multipoint-to-multipoint spanning tree with due consideration to capabilities of peers and their proximity to each other to minimize the communication delay among the peers. Many of the existing overlay multicast schemes are focused on delivery of media streams from a single source to a large number of end users (for example NRADA [5], NICE [6] and ZIGZAG [7]). ALMI [8] as another overlay multicast scheme uses a centralized algorithm to create a minimum delay spanning tree for multi source overlay multicasting. ALMI however has not considered actual link capacity constraints of the nodes when constructing an overlay tree.

Here our aim is to form a multipoint-to-multipoint tree for interactive multiparty communication amongst the peers. Section 4 presents the simulation results on the performance of our proposed architecture. Overlay multicast approaches are further discussed in section 5 and concluding remarks are presented in Section 6.

II. MIXING AND FORWARDING OF AUDIO STREAMS

A conceptually simple model for the peer-to-peer voice communication is for every peer to send the captured voice of its user to every other peer belonging to the same communication zone. This could be done using IP multicast (in rare cases when it is supported), multiple unicast flows or overlay multicast among the peers (such as SCRIBE over a Pastry overlay [9]). While simple, this model is not scalable with respect to number of participants in the multiparty voice conference. The portion of downstream bandwidth of each node that is allocated for voice communication is required to be at least equal to the product of number of peers in the communication zone and the bit rate of each voice stream. Likewise the upstream bandwidth may be excessive as each peer may have to relay several voice streams as part of the overlay multicast operation.

In practice, we believe that the following assumptions are reasonable:

- 1- Incoming (downstream) and outgoing (upstream) capacity of peer nodes are rather small and therefore can accommodate only a few (say, two or three) voice streams in upstream and downstream directions. In the case of asymmetrical access bandwidth, we consider a minimum duplex capacity that is feasible for both directions.
- 2- The voice of each peer must be included in the mixed audio received by every other peer in the same communication zone.
- 3- The communication delay from any peer to any other peer node within the same communication zone must, as much as possible, remain less than an acceptable upper threshold (e.g. 150 ms).
- 4- Behavior of peer nodes is unpredictable; i.e. peer nodes may join or leave the communication zone or fail at any instant and therefore it is necessary that the proposed scheme be able to handle these events properly.
- 5- Due to resource limitation at each peer node, the control overhead at each peer node should remain small.

We achieve goals 1 and 2 above by using a distributed mixing of the voice streams. Goals 3, 4 and 5 above are addressed by the multipoint-to-multipoint spanning tree algorithm presented in the next section.

Figure 1. Scenario for mixing and forwarding of voice streams using peer nodes in the overlay multicast tree

To clarify the distributed mixing operation, let us assume that all peers in a communication zone have formed a multipoint-to-multipoint minimum delay tree as shown in Fig. 1. This tree has been formed with due consideration to the capacity limitations of peers. In the case of Fig. 1, the node degrees of peers P1 and P2 are two. In other words, these peers can maintain at most two duplex voice flows simultaneously. The other two peers (P3 and P4) have lower access bandwidth and can only support one voice stream in either direction. Clearly, this is a minimum requirement to be able to participate in the communication zone; however, such peers will not be able to contribute to the distributed mixing operation.

Any peer with the node degree of two and above will perform selective audio mixing as shown in Fig. 1. The algorithm is very simple. Every outgoing voice stream is a linear mix of all the streams arrived at the peer except the stream arrived from the same direction. Note that all the outgoing streams include the voice generated by the local client. For example, P1 receives the voice stream S3 from the direction of P3, receives a linear mix of S4+S2 from the direction of P2 and the captured voice of its own client S1. The outgoing voice stream from P1 in the direction of P2 is S1+S3 and in the direction of P3 is S1+S4+S2. It is clear that all peers will be able to receive all the audio streams of interest in their communication zone. Every peer with the nodal degree of k (that is, a peer that can support k duplex voice streams) should also be able to perform k linear mixing operation, which is not excessive.

III. MULTIPOINT TO MULTIPOINT SPANNING TREE

The minimum communication delay between the peers can be achieved by using multiple unicast shortest path flows (assuming no congestion on the shortest path). As mentioned before, this model does not scale due to excessive bandwidth wastage.

The second alternative is to use multiple source rooted overlay multicast trees from each source of audio to all other peers (listeners). These multicast trees are formed to minimize the delay stretch and therefore must take into account the proximity (in terms of network delay) of peers with respect to each other. One such technique has been developed by us to create overlay multicast trees among a set of nodes [3]. Our

method assumes the knowledge of the spatial coordinates of the nodes in the Internet. The spatial coordinates are obtained using the approach presented in [10], where the network distances (delays) are predicated based on measured bandwidth of nodes and measured round trip delay (RTD) from a set of landmarks. It is shown that in many cases, the accuracy is around 90%. Consequently, the geometric coordinates of nodes and their distances from each other within the geometric space formed by these measurements provide a map of network delay distances between the peers. A shortest delay multicast tree can then be created from each source to all other peers (see [3] for details). In general, shortest delay multicast trees from different sources of audio will be different and therefore, we need to support multiple trees (at the worst case, equal to the number of peers within a communication zone). This will violate the bandwidth constraints discussed before and also make the distributed mixing operation impractical.

In this paper, therefore, we propose a third alternative for overlay communication among the peers. Our intention is to create a single multipoint-to-multipoint spanning tree (i.e., spanning all the peers in a given communication zone) that provides minimum overlay delay and conforms to the nodal degree constraints of the peers. In our algorithm, the overlay delay is minimal with respect to a judiciously selected root node. The algorithm presented shortly is based on a modified version of the Dijkstra's algorithm in which overlay cost (delays) are minimized from a selected peer node referred to as the root. Note that minimization of delay from the root node to all other nodes cannot be assumed to result in a minimum delay between any two nodes in the tree. However, as shown by the simulation results, by appropriate selection of the root node, the extra delay compared to the second alternative based on source rooted multicast trees is not excessive. The advantage of the shared spanning tree, however, is that we can use the distributed mixing operation which allows the system to scale to reasonably large number of participants.

To form the multipoint-to-multipoint spanning tree (hereafter referred to as MMST), the peers must obtain either their round trip delays or geometric coordinate in the network delay space (based on RTD measurements from the landmarks as discussed above). It is also assumed that peers are aware of each other's upstream and downstream capacity constraints (in terms of nodal degree) using an appropriate control signaling.

TABLE I. DATA BASE USED FOR CONSTRUCTION OF MMST

Node to node delays/distances	A	B	C	D	E	F	Total delay	Node Capacity
A	0	2.5	3.5	2	2.2	2	12.2	2
B	2.5	0	3.2	3.6	5.6	2.2	17.1	2
C	3.5	3.2	0	3	5.1	5	19.8	2
D	2	3.6	3	0	2.2	4	14.8	3
E	2.2	5.6	5.1	2.2	0	3.6	18.7	2
F	2	2.2	5	4	3.6	0	16.8	2

For instance information about peer nodes within a communication zone can be obtained from a state information server or a P2P discovery service for the communication zones.

Figure 2. Formation of minimum delay spanning tree during construction

Every node locally executes the same spanning tree calculation algorithm for determining its neighboring nodes (parent and children) and given consistency of information, the calculations in individual peers would result in a consistent MMST.

Using the locally constructed tree every node is then able to find its neighbors, i.e. parent and children nodes. It should be emphasized that all nodes are assumed to have exactly similar database (see table 1 used in the given example) and therefore all nodes will derive the same tree. Fig. 2 shows the steps of the algorithm in formation of MMST.

These steps are described below:

1- Initialize the set of nodes in MMST (referred to as S) to an empty set. The algorithm ends when all the nodes are included in S. Here we assume that every node running these steps has already obtained the list of all peers in the communication zone and their related information.

2- Select a peer node (from the obtained database of peer nodes) that minimizes the average of maximum overlay delay derived based on estimation in (1):

$$DMax_{avg}(i) = D_i * Minimum\{((Log \frac{N-1}{C_i}) / Log C_{avg}), (N-1)/C_i\} \quad (1)$$

Here $DMax_{avg}(i)$ is the estimated average of maximum overlay delay when node i is the tree root, $D_i = (\sum_{all..j} RTD(i, j)) / (N-1)$ is the average round trip delay of node i

from all other nodes, $C_{avg} = (\sum_{all..j} C_j) / N$ is the average in/out

degree of peer nodes, C_i is the in/out degree of node i and N is number of peer nodes in the communication zone. In (1), $Minimum\{((Log \frac{N-1}{C_i}) / Log C_{avg}), (N-1)/C_i\}$ gives an estimate for average maximum depth of the overlay tree with node i as its root. In (1) a product of average tree depth and average RTD

delay of the tree root from all other nodes (in the tree) is used as an estimate for average maximum overlay delay of that tree. According to the obtained results maximum overlay delays of the trees rooted at nodes minimizing (1) are larger from the maximum overlay delays of trees with optimally selected root nodes by a small percentage (i.e. by less than 10 percent). Overlay delay of the selected root node is then set to zero and its capacity (max nodal degree) is initialized. In Table 1, D is the node that minimizes (1) and therefore is chosen as the root of the tree, Fig.2 (a).

3- Sort all peer nodes in ascending order of their RTD from the selected root node and their in/out degree. Nodes with in/out degree of 1 cannot have children nodes and therefore are placed at the end of the list. Additionally for two nodes with similar delay from root node the node with larger degree is inserted first. If there is still a tie between 2 nodes, the one with higher ID (e.g. p2p ID) will be inserted first. In this example, the sorted nodes according to their delay/distance from D are: A, E, C, B and F. Call this sorted list T.

4- While the sorted list of unattached nodes, T, is not empty remove the first peer node from it. In Fig. 2 nodes A, E, C, B and F are removed one at a time from the sorted list T, (created in step 3).

5- From the set S, find the first attachable node (i.e. node with unallocated capacity larger than voice stream bit rate) that gives minimum overlay delay from root to the removed peer node from T. In Fig. 2,(e), the next removed node from T is B and the first attachable node in S that gives minimum overlay delay from the root is A i.e. $RTD(D,A)+RTD(A,B)$ is minimum, where $RTD(A,B)$ is round trip delay or distance between nodes A and B. In this case A and B will become parent and child in the tree. In Fig. 2 numbers inside the nodes show the overlay delay of the nodes from root.

6- Insert the selected node in the children list of the parent and record the child's overlay delay from the root. Reduce the capacities of both parent and child by one degree (i.e. by one voice stream).

Repeat steps 5 and 6 until all elements of T are added to S (MMST tree). Fig. 2 shows the tree after each insertion starting from (a) and ending with (f). The inserted nodes to S are A, E, C, B and F and their parents are respectively D, D, D, A and E.

As earlier mentioned all nodes must use same version of database for constructing MMST. It is therefore necessary that nodes include version of the used database in their control messages. Control packets are then forwarded through established TCP connections along the constructed overlay tree while voice streams are transmitted along the same tree using UDP packets.

A. Joining of a Node

As mentioned before, the membership of multiparty voice communication service is dynamic. In the case of voice chat rooms and conferences, joining and leaving a conference is often initiated by an explicit action of the user. In the case of networked virtual environments, movements of avatars will change the composition of communication zones. In both cases, it is important to have efficient methods for

joining/leaving a peer to/from a communication zone. The steps for the insertion of a new node in the MMST are as follows:

1- The joining peer finds the zone database server (e.g. using p2p service, game or virtual environment state information server, etc.). Here the main emphasis is on the join process after a database server for the communication zone/s has been found.

2- The joining node gets the list of all peers in the communication zone and their related information (e.g. information similar to table 1).

3- Using the same MMST construction algorithm, the joining node locally constructs a tree and finds its parent and children (if any), establishes a connection with them and finally sends a join message to them. A join message contains all details of the joining node.

4- Using the same construction algorithm in the previous section a node receiving a join request recalculates its position (neighboring nodes) in the MMST.

5- After local derivation of overlay tree, a node first establishes a connection to its new neighbors (if any) and later sends join message to its uninformed neighbors.

6- A node, with new neighbors, later has to send a disconnect request to end its connection with previous neighbors.

In the above insertion process, joining node follows steps 1 to 3, and other nodes follow steps 4 to 6. Based on the above algorithm, join message traverses the modified MMST and therefore update delay from joining node to other nodes is roughly proportional to overlay delay between nodes and as a result, a larger node-to-node overlay delay causes a larger update delay.

In a departure process steps 4 to 6 are similarly used for forwarding of the leave request of a node. A departing node only has to send a leave request to its neighbors (parent and children). Remaining nodes then follow steps 4-6 to reconstruct the new overlay multicast tree and advertise (forward) the leave message to their neighbors along the locally derived new tree.

One remaining issue is to determine when to start using the new MMST. This can happen after the farthest node from the joining node has received the join message. Based on the overlay delays/distances it is possible for each of the nodes to locally predict the time by which farthest node will receive the join message.

Despite the policy for using new tree after the calculated time, it is still quite likely that during the transitional phase different member nodes use different versions of MMST. In this scheme therefore each voice packet includes a tree version which is used by the receiver node for mixing and forwarding of the voice packets. It should be mentioned that tree version, (incarnation), has been earlier proposed and used in [8]. Similar to ALMI each node keeps a copy of the previous MMST tree of its zone for forwarding of the packets with a previous version. As a result of possible forwarding of voice packets along two different versions of overlay tree, this policy may

cause congestion and packet loss on network interfaces of the nodes during overlay tree replacement phase.

B. Coordinate Update

A member peer node sends a coordinate update request to other member nodes in its zone if it detects a noticeably different coordinates for itself which affects the structure of the tree. Here each node follows a similar approach to the *Join* process for forwarding of the coordinate update request message. Steps 4-6 in the *join* process are therefore used by all nodes for locally constructing of the new tree and forwarding of the coordinate update request.

IV. SIMULATION EXPERIMENTS

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 and each stub domain is connected via a single stub-transit link to a transit router. Each peer node is assumed to be collocated with a stub router, i.e. access delay from a host to its stub router is ignored (as it cannot be controlled by us). In these experiments every node is assumed to have exactly similar set of information about all other nodes in the same communication zone.

Two important measures used for performance study of the peer-to-peer overlay trees are delay stretch and link stress. Delay stretch is defined as ratio of overlay delay along the multicast tree between 2 nodes over shortest unicast delay between the same 2 nodes. Link stress is defined as the number of times copies of the same packet traverse a link for an overlay tree. Here each node is likely to change the application layer contents (i.e. speech samples) of multicast packets prior to forwarding them. Link stress therefore does not seem to have a meaningful interpretation in here and is not presented in this paper.

Fig. 3 shows cumulative distribution for peer-to-peer (p2p) delay stretches for any pair of nodes in the shared overlay tree. In this experiment all nodes can receive and transmit up to 3 voice streams.

Figure 3. Cumulative probability distribution of delay stretch for host-to-host delay in the overlay multicast trees (Each Peer node has a link capacity (nodal degree) = 3 times voice stream bit rate)

Figure 4. Average delay stretch for peer nodes in the overlay multicast tree with 26 and 14 nodes for both delay and coordinate based minimum delay / distance trees versus average degree of the nodes, (error bars show 95% confidence intervals)

Based on these results with 14 nodes in the same zone, more than 90 percent of the pairs of nodes have delay stretches less than 3 which can be acceptable if unicast delays are rather small (e.g. less than 50 milliseconds). The main reason for rather large number of p2p delay stretches is the fact that the same overlay tree is used for sending the voice streams.

Fig. 4 shows delay stretch versus different average node link capacities (node degrees). In this experiment nodal degree of the peer nodes are uniformly distributed between 2 to 5. According to the presented results in Fig. 4 with only a small number of nodes having a capacity for accommodating 3 or more (duplex) voice streams, a rather acceptable average delay stretch may be achievable. Results also show that delay performances of the constructed trees based on coordinates in network geometric space are close to the performance of constructed MMST trees using exact p2p unicast delays and therefore coordinates instead of exact p2p delays can be used. Assuming that global landmarks are available in the Internet then a node may reuse its derived coordinates as many times as needed before another recalculation of its coordinates is necessary, [10].

In another experiment we compare the overlay delay of each member node from other member nodes of the MMST with measured overlay delays of the same node from other nodes in its own optimum source rooted overlay spanning tree, Fig.5.

Figure 5. Average ratio of overlay p2p delay in the single overlay tree over overlay delay of source rooted tree versus average normalized link capacities (error bars show 95% confidence intervals)

Assuming $D(k,j)$ as overlay delay from node k to node j

along MMST, and assuming $D_s(k,j)$ as overlay delay from node k to node j along (optimum) source rooted tree of node k then Fig. 5 shows average ratio of $D(k,j) / D_s(k,j)$. Obtained results show that overlay delay between any two member nodes in the constructed MMST tree is on average around two times the overlay delay between the same two nodes in their source rooted minimum overlay delay trees. It is interesting to notice that regardless of the average link capacities of the nodes the measured ratio remains approximately equal to 2. This can be attributed to the fact that an increase in the average link capacity of the nodes improves node-to-node overlay delays with a similar factor in both source rooted and MMST trees.

V. DISCUSSION

Overlay multicast as a means of streaming real time information has been recently studied extensively. The goal of overlay multicast is to construct and maintain efficient distribution trees between the multicast session participants, minimizing the performance penalty involved with application-layer processing [5]. Many of the overlay multicast mechanisms which aim at reducing the overall delay [5,6,7] construct a minimum height (or minimum diameter) tree using constrained degrees for controlling bandwidth usage or link stress at the physical layer. Recently in [12] message distribution delay and communication delay of end hosts have been used as a single cost to characterize the performance of multicast trees. Based on this characterization, approximation and heuristic methods for construction of efficient overlay trees have been developed.

Seemingly none of the existing schemes consider all of the key properties required for the p2p multipoint communication. Key properties of the proposed overlay multicast approach are: A) partitioning of the peer nodes into communication zones, B) proper selection of a root node for each communication zone, C) complete reconstruction of an efficient shared overlay tree after join/departure of the nodes without a need for a pre-existing mesh [13], D) observation of the nodal degrees during the tree construction and therefore preventing any bottleneck along the constructed trees.

Here root node of the overlay tree for each communication zone is selected such that the average maximum overlay delay as the cost of the tree remains close to its minimum, (Section III).

Another important aspect of the overlay tree approaches is their construction time. Construction and modification time of overlay trees is of significant importance when peer nodes change their communication zone frequently. Here based on using a delay prediction method [10] instead of actual RTD measurements it is possible to reduce overlay tree construction and modification time.

VI. CONCLUSION

In this paper a distributed mixing scheme for realization of multiparty voice communication in peer to peer environments has been proposed. Distributed mixing scheme has been shown to be an effective approach for streaming of voice streams to

end nodes with constrained link capacities. In this scheme a single overlay spanning tree as the basis for distributed mixing of the audio signals has been used. An algorithm for construction of an efficient multipoint to multipoint spanning tree of nodes with constrained network link capacities has been simulated. Results have shown that node to node overlay delay in the constructed trees is on average around two times the overlay delay in their source rooted (minimum overlay delay) spanning trees which seems to be acceptable. Simulation results have also shown the effectiveness of using geometric coordinates of the nodes for construction of overlay multicast trees.

Finally Interactive voice communication imposes severe end-to-end delay constraints which may limit number of clients within each communication zone to a few tens of peer nodes (depending on the RTD of nodes, their link and processing capacities this number can change significantly). This limit on number of clients however is not expected to reduce usefulness of the proposed multipoint communication approach for many networked applications since there is still no limit on number of communication zones.

VII. ACKNOWLEDGMENT

The support of the Co-operative Research Centre for Smart Internet Technology for this work is hereby acknowledged.

REFERENCES

- [1] 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.
- [2] Boustead, P., Safaei, F., Dowlatshahi M., "DICE: Internet Delivery of Immersive Voice Communication to Crowded Virtual Spaces", Proc. of IEEE International Conference on Virtual Reality (VR 2005).
- [3] 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 (ICON2004).
- [4] S.A. Baset, H. Schulzrine, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol,"
- [5] Y.-H. Chu, S. G. Rao, H. Zhang, "A case for end system multicast," in ACM SIGMETRICS 2000, 2000.
- [6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in Proceeding Of ACM SIGCOMM, 2002.
- [7] Duc A. Tran, Kien A. Hua, Tai Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," Infocom 2003, P1283-1292.
- [8] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "Almi: an application level multicast infrastructure," in Proceedings of the 3rd USENIX Symposium on internet technologies and systems (USITS 2001), Mar. 2001.
- [9] M. Castro, P. Druschel, A. M. Kermarrec, A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," IEEE JSAC, VOL. 20, NO. 8, OCTOBER 2002 P100.
- [10] T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches", Infocom 2002, P170-179.
- [11] E.W. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. Infocom 1996.
- [12] E. Brosh, Y. Shavitt, "Approximation and Heuristic Algorithms for Minimum Delay Application-Layer Multicast Trees", Infocom 2004, P2697-2707
- [13] S. Wu, S. Banerjee, X. Hou, R.A.Thompson "Active Delay and Loss Adaptation in Overlay Multicast Tree", ICC 2004, P2014-2018