2006

Overlay multicasting of real-time streams in virtual environments

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

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

Publication Details
Overlay multicasting of real-time streams in virtual environments

Abstract
Multicasting of real-time streams (voice, video, etc.) for interactive communication in networked virtual environments (V.E.) is becoming an increasingly important service. Here as an example characteristics of interactive voice scenes for participants in a V.E. are presented. Proxy based overlay multicasting of real-time streams is a candidate architecture for this type of service. Here we propose a scheme for angular clustering of destination proxies in multi dimensional network geometric spaces as the core algorithm for construction of efficient overlay multicast trees. We show that using the proposed angular clustering scheme it is possible to construct source rooted overlay trees with acceptable delay and overhead traffic performances.

Keywords
Overlay, multicasting, real, time, streams, virtual, environments

Disciplines
Physical Sciences and Mathematics

Publication Details

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/1640
Overlay Multicasting of Real-Time Streams in Virtual 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—Multicasting of real-time streams (voice, video, etc.) for interactive communication in networked virtual environments (V.E.) is becoming an increasingly important service. Here as an example characteristics of interactive voice scenes for participants in a V.E. are presented. Proxy based overlay multicasting of real-time streams is a candidate architecture for this type of service. Here we propose a scheme for angular clustering of destination proxies in multi dimensional network geometric spaces as the core algorithm for construction of efficient overlay multicast trees. We show that using the proposed angular clustering scheme it is possible to construct source rooted overlay trees with acceptable delay and overhead traffic performances.

Keywords-overlay multicast trees; delay prediction; Overlay networks; clustering; interactive virtual networks

I. INTRODUCTION
Overlay multicast as a means of streaming real time information for different services has been recently studied extensively [1, 2, 3, 4 and 5]. Recent studies have shown that proxy based overlay multicasting could achieve comparable performance to IP multicast and therefore is becoming the preferred approach for real-time streaming, [6].

Networked virtual environments (V.E.) are also becoming increasingly important for many different types of services. Interactive communication in networked virtual environments can largely add to the usefulness and popularity of these environments. In virtual environments a large number of rather small and frequently changing overlay multicast trees may exist, which to the best of our knowledge is not commonly considered by other studies. In many of the overlay multicast tree construction schemes [2, 4, 5] performance measures of overlay multicast trees like delay stretch (defined as ratio of overlay delay over shortest unicast delay) and link stress (defined as number of copies of the same packet traversing the same physical link) are gradually improved. Another group of construction schemes use shared overlay trees [3] which are not as efficient as source rooted overlay multicast trees.

Recently a proxy based overlay multicasting for real-time streaming of captured voice of participants in Virtual Environments (V.E.) has been adopted [1]. In this architecture each proxy is responsible for delivery of audio streams to its assigned group of clients. When a given participant is talking, its audio packets are captured by the client and sent to its respective local proxy. Local proxy will then identify all other participants within the VE that might require this audio stream and will act as a source (or root) of multicast to forward the audio packets to their proxies (referred to receiver or destination proxies).

After receiving all necessary audio streams, 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 limits of the clients are not exceeded [7]. The clients can then render the audio scenes based on the received information (position, orientation, etc.) of the streams.

Here only source and receiver (interested) proxies participate in construction of the overlay multicast trees (OMT). The assumption is that, prior to the construction of the overlay multicast tree, root proxy of each stream knows all destination proxies (their coordinates in the network geometric space together with all other necessary information) for that stream. This assumption is based on the fact that VE state server/s can provide all of the necessary information for all proxies. VE state servers already have to disseminate the state information to a large number of clients and therefore it would be acceptable to assume that a separate or the same group of VE state server/s be responsible for derivation and dissemination of the necessary information among the proxies.

A better understanding of the target virtual environments will further clarify our intended overlay multicast scenarios. In this regard section II presents required characteristics of overlay multicast trees in virtual environments. Section III reviews a previously proposed recursive multicast tree construction algorithm [1]. Later in this section motivation for angular clustering as the core of the recursive construction algorithm is presented. Correctness of angular clustering largely depends on correctness of positioning of the nodes with respect to each other in a network geometric space. It has been shown that in a two dimensional network geometric space correct positioning of the nodes may not be possible [8]. Section IV is therefore devoted to the presentation of angular clustering in multi dimensional geometric space. Section V compares performance of overlay trees constructed using angular clustering and two other benchmark overlay tree construction mechanisms. Section VI concludes the paper.

II. VIRTUAL/PARTICIPATORY ENVIRONMENTS
Multicasting of real time streams for interaction in virtual environments needs a better understanding of requirements of these environments. Here as an example properties of real
time voice communication in interactive virtual environments are investigated.

![Figure 1. Average percentage of change in destination proxies after a random movement of all the avatars in the virtual environment](image)

It has been observed that multicast tree sizes depend on number of avatars in voice scenes and are much less dependent on total number of proxies; i.e. Regardless of total number of proxies, number of receiver proxies is always less than or equal to the number of avatars in the hearing range of any avatar. In the simulated VE average number of proxies within each tree is around 13 proxies.

Fig. 1 further shows average percentage of change in multicast trees after a random waypoint motion for different maximum move sizes given in percentage of the virtual world dimensions. Percentage of change in a multicast tree is found by comparing the set of destination proxies before and after the movement of avatars.

As can be seen even for relatively small movements of avatars percentage of change in multicast trees is rather large (more than 10 percent change after a maximum of 1 percent move). These results show that in rather large virtual environments interactive voice scenes need: A- construction of a large number of small multicast trees (here up to 5000 multicast trees each having up to 25 destination proxies); B- rapid modification or reconstruction of multicast trees; C- acceptable computational overhead for construction or modification of the multicast trees.

### III. REVIEW OF RECURSIVE MULTICAST TREE CONSTRUCTION ALGORITHM

This section reviews the proposed recursive multicast tree construction algorithm in [1]. This algorithm is devised for proxy based architecture and is reviewed by giving an example, see Fig. 2. Based on this algorithm required steps for construction of overlay multicast tree are as in the following.

1- Local proxy (also referred to as source proxy when clustering) of the source client receives a complete list of proxies that require the stream (e.g. voice stream) of its local source client. In Fig.2 each small circle represents a proxy that requires the stream of a local client of proxy P0. 2- Source proxy clusters N recipient proxies into K groups where K is the maximum allowed number of replications per incoming stream, see Fig. 2. In this example maximum replication limit for source proxy P0 is 3, and C1, C2 and C3 are the three clusters that P0 created based on angular clustering method as described in [1]. 3- Source proxy selects closest proxy in the network geometric space to itself within each cluster as its direct child proxy and sends a list of all proxies in that cluster to the determined child proxy. Here proxy P1 in cluster 1 (C1) is determined as closest proxy to P0 and P0 sends a list containing P1, P2 and P3 to proxy P1. 4- After receiving a list of proxies, if the received list contains more than one proxy recipient proxy repeats steps 2 and 3. The proxy that receives a list of size one will assume itself as a leaf node. Here proxy P1 receives a list of size 3. After deleting itself from the list because of maximum replication limit of 1, P1 sends a list containing proxies P2 and P3 to closest proxy (to itself) from the list, i.e. proxy P2. Obviously this approach always leads to loop free overlay multicast trees.

Construction of efficient overlay tree in a single iteration, independent replication control at each proxy, loop free construction, simplicity and an average overlay tree height close to $\log_k^N$ are main characteristics of the proposed...
algorithm. In this algorithm the middle proxy can independently and dynamically change its maximum number of clusters (replications) to avoid congestion in its network links. The set of actions for the join or departure of one or more proxies to an already existing overlay multicast tree are as in the following.

1- Local proxy of the source client (also referred to as source proxy in the next steps) receives a new set of receiver proxies requiring the client’s stream; 2- Source proxy clusters proxies and determines if there is a change in the set of proxies in any of the clusters.

For every changed cluster, as before source proxy forwards a list of proxies to the closest proxy within changed cluster and updates its direct child node in that cluster if necessary; 3- Recipient proxy as the elected source proxy for proxies in the received list continues step 2 while list size is larger than one.

The above approach only modifies the affected clusters of the OMT and therefore reduces overhead traffic. Reception of the list of destination proxies constitutes the main part of the control traffic overhead. This traffic overhead is directly proportional to the rate of changes in the set of required streams by proxies, i.e. traffic overhead increases with an increase in changing rate of the destination proxies of streams.

In [1] angular clustering algorithm has been proposed for use in the above OMT construction and modification schemes. Here motivation behind the angular clustering algorithm is presented.

A. Motivation for Angular Clustering

Angular clustering tries to minimize distance stretch (defined as ratio of overlay distance between two proxies over shortest direct distance between same pair of proxies) by minimizing separation angle between the destination proxies within the same cluster from the perspective of their source proxy (see Fig.2) in the network geometric space.

![Figure 3. Effect of angular separation](image)

Here motivation for this approach is explained by giving an example (see Fig.3). In this example a, b, c are respectively Euclidian distances between (P0, P1), (P0, P2) and (P1, P2) and C is the angle between (P0, P1) and (P0, P2). According to the trigonometric relation we have:

\[ c = \sqrt{a^2 + b^2 - 2ab \cos C} \]

In this example we assume that P0 as the source proxy has inserted P1 and P2 in the same cluster. We further assume that \( a < b \). Then according to the recursive construction algorithm P1 becomes the parent node for P2 and the overlay distance of proxy P2 from source proxy P0 is equal to \((a+c)/b\) and overlay distance stretch for P2 equals to \((a+c)/b\). This distance stretch is minimized if separation angle, C from the perspective of source proxy P0 is minimized.

B. Angular Clustering Algorithm in 2 Dimensional Space

In a two dimensional network geometric plane angular clustering puts destination proxies into K angular clusters where as before K is the maximum number of stream replications for a source proxy [1], see Fig. 2. Angular clustering forms clusters such that maximum separation angle between any two nodes within the same cluster is minimized. Further details of this clustering approach can be found in [1]. Unfortunately positioning of the proxies in a two dimensional network geometric space may not be always possible. Our simulation results have shown that using two dimensional geometrical positions of the proxies in a Transit-Stub network model [9] results in construction of overlay trees with rather large average delay stretches (>2.2 for overlay trees with 16 proxies). Therefore it is important to extend the proposed angular clustering to the multi dimensional network geometric space. Following section describes a proposed mechanism for this extension.

IV. ANGULAR CLUSTERING IN MULTI DIMENSIONAL NETWORK GEOMETRIC SPACE

As mentioned before correctness of angular clustering algorithm is largely dependent on correctness of positioning of the nodes with respect to each other in the network geometric space, i.e. it is crucial that positions of the nodes in the network geometric space rather correctly represent positions of those nodes in the physical network. In [8] it has been shown that a relatively accurate mapping of network positions of nodes to geometric positions in multi dimensional geometric space is possible only when number of coordinates is large enough (>4). It is therefore important to extend the angular clustering to multi dimensional network geometric space. Here we propose the criteria for this extension. Briefly in these criteria we first project all nodes onto (M-1) distinctive coordinate planes. Later we independently cluster projections of all proxies on each of the (M-1) planes. We define an objective function for selecting one of the (M-1) clustering options. In the following the proposed criteria is presented in more detail.

1- Find two dimensional angular positions of the projections of the nodes (from view point of projection of source proxy) on each of the (M-1) planes, \((M-1)O(N)\). Here j-th plane has \(X_{ij}\) as the common coordinate axis and \(X_j\) as its second coordinate axis \((1 \leq j \leq M-1)\) (i.e. \(X_{ij}\) is the shared coordinate by all used coordinate planes.)

2- Angularly cluster projection of all proxies in each of \((M-1)\) planes independently. Operation order: \((M-1)*O(N^2)+(M-1)*O(N)+O(N^2/K)\). This will produce \((M-1)\) clustering (overlay multicast tree) options.

3- Using \(O\) (defined as the objective function) calculate total worst case overlay distance for all clusters in each of \((M-1)\) planes, \((M-1)*O(N)\).

\[
\sum_{AIClusters} \sum_{i=1}^{ClusterSize} \sum_{j=1}^{J} D(P_{j-1},P_j) \tag{2}
\]

In \(O\) \(P_0\) is the source proxy, \(P_j\) is the j-th closest proxy to \(P_0\) in its cluster, \(cluster size\) refers to number of proxies in each cluster and \(D(P_{j-1},P_j)\) represents geometric distance.
between \( P_{j-1} \) and \( P_j \). This expression gives a sum of overlay distances of all proxies in all clusters when all proxies except source proxy have a maximum replication number equal to one.

In (2) expression \( \sum_{j=1}^{i} \sum_{j'=1}^{i} D(P_{j-1}, P_j) \) gives worst case overlay distance of destination proxy \( P_i \) from source proxy \( P_0 \). As an example in Fig.2 worst case overlay distance of destination proxy \( P_3 \) from proxy \( P_0 \) is the sum of distances from \( P_0 \) to \( P_1 \), from \( P_1 \) to \( P_2 \) and from \( P_2 \) to \( P_3 \).

Expression \( \sum_{i=1}^{\text{Cluster Size}} \sum_{j=1}^{i} D(P_{j-1}, P_j) \) gives a sum of worst case overlay distances of all destination proxies within a cluster. Expression (2) (objective function) gives a sum of worst case overlay delays for a clustering option in each of the \((M-1)\) coordinate planes.

4- From \((M-1)\) clustering options, choose the option which minimizes total worst case overlay distance as obtained in (2).

Above clustering approach is based on the following two observations: (a) - A group of nodes closest to each other are likely to fall in the same angular cluster in one or more of the \((M-1)\) planes and (b) - The plane in which most of the nodes are inserted in their optimal clusters is likely to have a minimum value of total worst case overlay distance.

Fig. 4 further clarifies angular clustering in a 3 dimensional geometric space. In this example \( P_0 \) as the source proxy is clustering \( P_1 \) to \( P_4 \) in XY and XZ planes (X is the shared coordinate axis). It is not difficult to observe that angular clusters in the XZ plane are \((P_1, P_4)\) and \((P_2, P_3)\), and angular clusters in the XY plane are \((P_1, P_2)\) and \((P_3, P_4)\). If according to (2) total worst case overlay delay for the clustering option \((P_3, P_4)\) and \((P_1, P_2)\) is less than total worst case overlay delay of clustering option \((P_1, P_4)\) and \((P_2, P_3)\), then \((P_1, P_2)\) and \((P_3, P_4)\) clustering option will be chosen for construction of the multicast tree.

![Angular clustering in a 3 dimensional geometric space](image)

After selection of the clustering option, source proxy \( P_0 \) sends lists of proxies within each cluster to the geometrically closest proxy in each of those clusters (for e.g. \( P_1 \) and \( P_4 \)) and registers \( P_1 \) and \( P_4 \) as its direct children within the constructed overlay tree.

Intuitively angular clustering is also expected to implicitly reduce network-layer link stress since it is likely that direct unicast paths from source proxy to the destination proxies in the same angular cluster share a large percentage of the underlying network-layer links.

Main disadvantages of angular clustering are: A) Dependence on accuracy of coordinates and positioning of the nodes and B) Unbalanced Clustering. In angular clustering it is quiet possible that a cluster remains empty or be filled by a few proxies and remaining cluster/s become significantly larger. This usually happens when from the viewpoint of the source proxy one or a few nodes compared to the rest of the nodes are on different sides of the network geometric space. Although this unbalanced clustering does not seem efficient at the application layer it usually reduces link stress in the network layer and has been observed to have a limited effect on increasing delay stretch. To reduce this imbalance one probable approach is to allow re-clustering of large clusters using empty clusters. (In this paper re-clustering has not been used.)

It should be mentioned that the above approach for clustering in \( M \) dimensional network geometric space can be further improved by angularly clustering the projections of all destination proxies on all possible coordinate planes i.e. on all \( M(M-1)/2 \) distinctive coordinates planes. This may further improve total worst case overlay distance of the nodes but will increase computational load of the algorithm. This improvement is more likely to be used for up to five or six dimensional network geometric spaces. In this paper we have not clustered destination proxies on all possible distinctive coordinate planes.

V. SIMULATION ENVIRONMENT AND EXPERIMENTAL RESULTS

Here in our simulation experiments we use a Transit-Stub topology [9] which generates a network with a two-tier hierarchy of transit and stub networks. The parameters for all simulated network topologies are as follows. The network consists of six transit domains, each with an average of 16 routers. Each transit router is connected to an average of 12 stub domains, and each stub domain consists of 9 routers.

Routers at any of the transit or stub domains have an average of 3 physical links to the network. Each stub domain is connected by a single link to a transit domain. Proxies are assumed to be directly connected to the LAN interfaces of routers in stub domains. In these experiments number of receiver (destination) proxies is increased from 16 to 256. Each packet is replicated by each proxy up to a fixed maximum number. For each multicast tree size performance measures are obtained by averaging the results of 50 different experiments; i.e. for each multicast tree size and replication number we use 5 different transit-stub graphs and in each graph ten different set of randomly chosen stub routers from the set of all stub routers are chosen as root and receiver proxies.

In all experiments 12 landmarks are assumed to be available. With twelve landmarks, coordinates of proxies in an eleven dimensional geometric space have been calculated. Coordinates for landmarks and proxies have been found using the Downhill Simplex optimization technique [10]. In these experiments we use source rooted minimum delay and source rooted minimum geometric distance overlay trees as benchmarks. The source rooted minimum geometric distance overlay tree is constructed based on geometric distances of the nodes in the network geometric space. Similar to angular clustering this algorithm does not use information about maximum replication number of any proxy other than the source proxy. The minimum delay degree constrained source
rooted overlay tree is constructed using actual delay between proxies and is based on the Dijkstra’s algorithm in which overlay delays from the root proxy of the overlay multicast tree to all proxies in the tree are minimized. This algorithm would be our benchmark for optimal result as it uses a global knowledge of the maximum replication numbers (of all proxies) and shortest unicast delays between all pairs of proxies.

Fig.5 presents average delay stretch versus multicast tree sizes when up to 2 replications per incoming stream are allowed. These results confirm that in comparison to optimal and minimum distance methods average delay stretch of overlay trees constructed using angular clustering method is acceptable.

Fig.6 displays average link stress for overlay trees constructed using optimal delay, minimum distance and angular methods. In these experiments maximum allowed replication number for each proxy is equal to 5. A lower replication number decreases average link stress for all methods. Results show that angular clustering has the smallest network-layer link stress.

Angular clustering approach is a rather computationally intensive method for large overlay tree sizes $O(M \times N^2)$. In interactive virtual environments this computational overhead seems to be acceptable since multicast tree sizes for many virtual environments are rather small, (i.e. $N$ number of receiver proxies is less than 25).

VI. CONCLUSION

In this paper an overlay multicasting scheme for real-time streaming in virtual environments has been investigated. In comparison to other overlay multicasting environments (Video on Demand, internet radio, etc) virtual environments possess fundamentally different characteristics.

Simulation results have shown that in virtual environments a large number of frequently changing small multicast groups may exist. It is therefore important to use overlay tree construction algorithms which do not need more than a single iteration for constructing overlay trees. A distributed recursive construction method is one such scheme previously proposed in [1]. The recursive scheme has been observed to construct inefficient overlay trees in more realistic transit-stub network models when using angular clustering in two dimensional network geometric spaces.

In this paper using a simple criteria angular clustering has been extended to multi dimensional network geometric spaces. Simulation results have shown that the extended angular clustering scheme can be used for construction of efficient overlay trees in terms of delay stretch and link stress.

VII. ACKNOWLEDGMENT

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

REFERENCES