International Journal of Natural and Engineering Sciences 5 (3): 61-64, 2011
ISSN: 1307-1149, E-ISSN: 2146-0086, www.nobel.gen.tr

# Multicast Tree Construction for Peer to Peer Video Streaming Systems

Müge SAYIT[1]* E. Turhan TUNALI[2]

[1] International Computer Institute, Ege University, 35100, Bornova, Izmir, TURKEY

[2] Department of Computer Engineering, Izmir University of Economics, 35330, Balcova, Izmir, TURKEY

***Corresponding Author**

**e-mail:** muge.fesci@ege.edu.tr

**Received:** August 19, 2011
**Accepted:** August 30, 2011

**Abstract**

In this work, a novel multicast tree construction algorithm for video streaming systems running on peer-to-peer networks is proposed. Both delay values between nodes and capacity of nodes are considered during multicast tree construction. The proposed algorithm aims at (i) constructing shorter trees in terms of hop number, (ii) optimal placement of nodes in the multicast tree so that received video bitrate is maximized. It has been observed through extensive simulations that the proposed approach achieves high performance in terms of received bitrate and tree height. The proposed approach is compared with another multicast tree construction algorithm from the literature and results are summarized in comparative graphs.

**Key Words:** Video streaming, multicast tree, peer to peer networks

## INTRODUCTION

In order to meet the growing demand for streaming video services over the Internet, many video streaming systems have been proposed by both academia and commercial industry. Recently peer-to-peer video streaming became one of the major alternatives in video streaming. Since a large number of nodes exist in a peer to peer system, optimal multicast streaming paths need to be constructed for delivering video packets to the requester nodes. IP multicast services are not quite adequate because of scalability problems and the absence of error and flow control algorithms. Thus, video multicast applications running on application layer are proposed in the literature. Multicast tree construction algorithms provide efficient video delivering paths and have a low message and algorithm complexity. Due to these reasons, various multicast tree construction algorithms were introduced for video streaming systems in the literature [1]. In this study, a novel multicast tree construction algorithm for peer to peer video streaming systems is proposed.

Considering the studies in the literature, the most important criteria in constructing parent - child relationships of multicast trees is the delay between the nodes, i.e. the packet delivery time from one node to another. It should be ensured that the distance between parent and child is limited to a determined value. Moreover, the number of hops between the tree root and the nodes should be minimized. This is due to the fact that more hops cause increase in latency and packet loss probability.

The tree algorithms proposed in [2, 3] are centralized in which the length of the constructed tree in terms of latency is guaranteed to be limited. [4, 5] are distributed algorithms that can be considered to be achieving the same goal. The multicast trees used for distributed systems are adapted to video streaming applications by considering the scalability, sense of the underlying network topology, node capacity and bandwidth.

In order to sense the underlying network topology, one approach used in the literature is placing the nodes that are topologically close to each other in the same cluster [6-9]. Therefore, the message delay between parent and child is limited in multicast trees constructed in each cluster. Another important criterion to be considered in tree construction is the capacity of the nodes in a peer to peer system. The capacity of a node is defined as the maximum number of nodes that can be connected to that node. However, the capacity limitation was not considered in the proposed systems of [6-9]. A disadvantage of multicast trees is that subtree of any node having a limited upload bandwidth will be negatively affected [10]. In order to prevent this disadvantage, we proposed placing the nodes with higher capacity near the root in our previous work [11]. [11] gives a hierarchical cluster approach to allocate the bandwidth budget. However, bandwidth-delay optimization is not elaborated. In [12], degree constraint multicast tree construction is proposed. Similarly, a peer to peer video streaming system based on degree constraint multicast tree construction is proposed in [13]. Although degree constraint multicast tree construction proposed in [12] and [13] consider capacity of nodes in the system, these works suffer from the number of tree levels constructed.

Our previous work [15] utilizes a greedy approach that first places the highest bandwidth node to upper layer and then chooses the next highest bandwidth node with lowest delay as a child. The policy of [15] may cause a high capacity node to be placed towards the root even if it has very high delay. This may cause extensively increased total delay on the broadcast tree. In this work, we propose a new algorithm to overcome this deficiency of [15]. The proceeding sections of the paper are organized as follows: In the following section, the details of the multicast tree construction algorithm are given. This is followed by the results section in which simulation results are presented. In the last section, concluding remarks are made

## MATERIALS AND METHODS

The bandwidth value between two nodes is more important than the delay between these nodes in video streaming applications. For video streaming on multicast trees, choosing higher bandwidth nodes closer to the root improves streaming performance. This is due to several reasons: First, we can connect more children to higher bandwidth nodes, thereby reducing the height of the tree, leading to improvement in delay. Second, a leaf –a node that is on the edge of the tree which has no children- of a tree cannot receive a video stream at a rate higher than the bandwidth of the bottleneck link between itself and the root. Therefore, by placing higher bandwidth nodes closer to the root, we improve the average utilized bandwidth. Because of these, bandwidth values have more importance than delay values and tree cost needs to be determined by considering this observation.

### Algorithm

A peer to peer network can be modeled as undirected graph. For a given graph *G(V,E)*, $e_{u,v}$ represents the cost of the edge between vertexes, i.e. nodes *u* and *v*. Following this terminology, we considered following properties to construct multicast trees:

- Let $c_u$ represents the node *u*'s capacity, i.e. the maximum number of children that can be connected to node *u* for all $u \in V$. Capacity of a node is limited by upload bandwidth of that node, hence, $d_u \le c_u$ where $d_u$ is the number of node *u's* children.
- To keep multicast tree height as short as possible, the relation between $d_u$ and $c_u$ must be as follows: if there are any nodes that are not located to the tree yet, $d_u$ should be equal to $c_u$, otherwise $d_u$ should be equal or less than $c_u$.

Algorithm starts with children selection of source node(s). If there exists more than one source in the system, the source nodes should choose one or more requester nodes with minimum edge cost ($e_{u,v}$). On the other hand, the number of children of a source node can not exceed the capacity of that source node.

```
1  S = { s1, s2, ..., sn}
2  R = { r1, r2, ..., rm}
3  Rs = ∅

4  while R≠∅ do
5      n = |S|
6      while ( ∑_{i=1}^{n} d_{c_i} < ∑_{i=1}^{n} s_{c_i} ) && (R≠∅) do
7              select ea,b with minimum cost where a ∈ S and b ∈ R
8              Rs = Rs ∪ {b}
9              R = R – {b}
10             da = da + 1
11             if (da==ca) then S = S - {a}
12     end while
13     S = Rs
14     Rs = ∅
15 end while
```

**Figure 1.** Multicast Tree Construction Algorithm Pseudo Code

According to the constraints above, our multicast tree construction algorithm is given in figure 1. The cost term of figure 1 will be explained in the next paragraph. Suppose that there are n source nodes and m requester nodes in the system. According to the algorithm given in figure 1, S is the set of source nodes and R is the set of requester nodes. $R_s$ represents the nodes located at the bottom layer of the tree and is initialized as empty set before algorithm starts. If the nodes in $R_s$ do not fulfill their capacity or if there are any nodes which are not connected to the tree yet, algorithm continues running the code block inside while loop at the 6th step. At the 7th step of the algorithm, the node having edge with minimum cost between it and any node *a* in $R_s$ are chosen and added to the set of $R_s$. The degree of node *a* is incremented (10th step), if node *a* fulfills its capacity, it is extracted from the source set (11th) since new children no longer can be connected to node *a*. At the 13th step of the algorithm, $R_s$ is assigned as a new set of virtual sources and while loop starts back at the top again.

According to multicast tree construction constraints, cost and height of the tree should be kept as small as possible. In this study, both the capacity of nodes and delay between two nodes are used to assign the cost values to bandwidth delay combinations. The cost values assigned are chosen in such a way that a balance between bandwidth and delay is constructed.

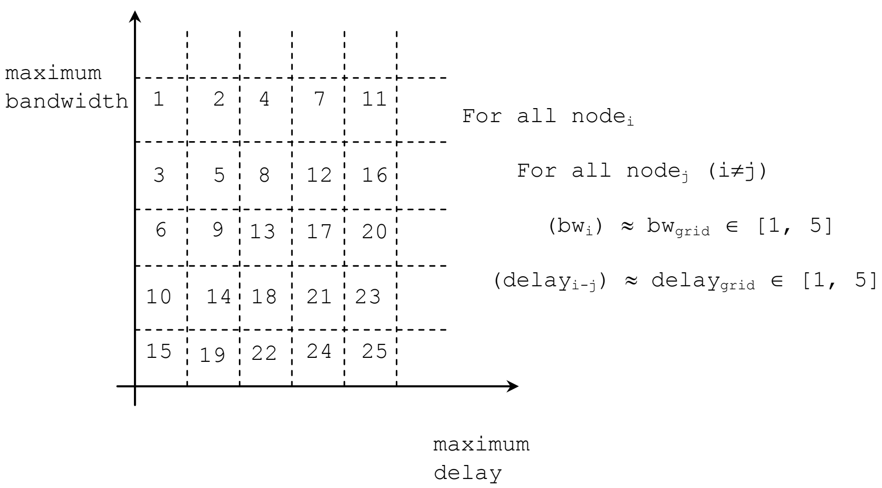


**Figure 2.** Weight Grid Table of Delay-Bandwidth Pairs

Both upload bandwidth and delay values categorized into five regions to form a five by five grid that is given in figure 2. When the cost values assigned in figure 2 are examined, it is observed that cost follows a specific pattern similar to a diagonal raster scan. This is the distinct policy that makes our approach different from all the studies in the literature. A node in a peer to peer video streaming system calculates the cost to any other node in the system by considering its upload capacity and the delay between itself and the other node. This cost represents the connection quality between these two nodes.

## RESULTS

In order to observe the performance of the proposed algorithm, network topology was constructed by GT-ITM [14] module. The topology consists of 500 nodes and all the nodes in the system are included in video streaming experiments. In all experiments, there is one source in the system. We have used the delays generated by the GT-ITM module. Generally, two nodes that share a logical edge in the overlay architecture may not actually be directly connected to each other. In this case, communication path between these nodes was chosen as shortest path in the network and the delay between these nodes was calculated by considering the shortest path between them.

In the experiments, hop count and bandwidth utilization of the proposed multicast tree algorithm was compared with similar multicast tree algorithm proposed in the literature. In the algorithm proposed by [13], multicast trees are constructed according to degree constraint multicast tree construction algorithm [12]. In the first set of experiments, capacities of all nodes in the system equal 3 as this value was used in the experiments given in [13]. According to this limitation, every node has one parent (except root) and can have maximum of 3 children. When the graph given in figure 3 was examined, cumulative statistical results show that the nodes were located on the multicast tree with 6 hops at most. It can be also seen from the graph that 70% of the nodes in the tree are 5 hops apart from the root. On the other hand, it was observed that leaf nodes in the tree proposed by Fei [13] can be 36 hops apart from the root, and only 4% of the nodes in the tree are 6 hops apart from the root. As it is mentioned in previous section, the height of the tree is an important criterion for video streaming applications since any error occurred in a node affects all subtrees of that node. Furthermore, if the number of hops from the source to destination node increases, the probability of packet loss before the packet reaching the destination also increases. In order to observe the effect in received bitrate of

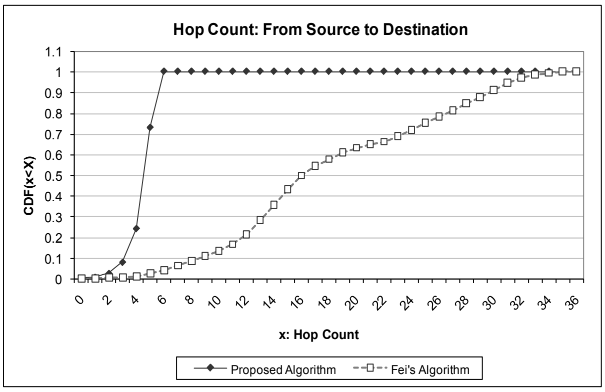


**Figure 3.** Comparative CDF Graph of Hop Count

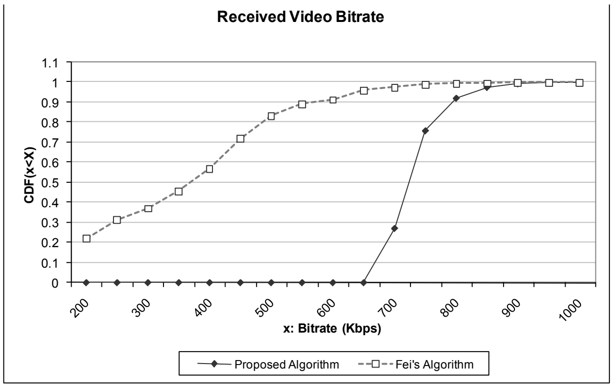


**Figure 4.** Comparative CDF Graph of Average Bitrate in Lossy Environment

tree height, we added packet loss probability of 5% to each link on the topology and measured received bitrate values by nodes in the system. Comparative results of received video bitrate in lossy environment are given in figure 4. It is seen from figure 4 that 90% of nodes receive video less than 600 Kbps in system proposed by [13] while all nodes in our proposed system receive video higher than 700 Kbps from the figure. Therefore, it can be said that shallow trees provide less packet loss during video streaming session.

One disadvantage of the proposed algorithm is that the total delay from source to each requester node is not minimized. The algorithm proposed by Fei [13] provides minimum delay since minimum spanning tree is constructed. According to the algorithm given in [13], average delay distance from root to the nodes on the tree equals 1.45 seconds while this value equals 2.53 seconds in the proposed tree construction algorithm. Therefore, considering the tree length in terms of latency, our algorithm has a penalty of 1.08 seconds. Nevertheless, this difference only increases the initial waiting time, i.e. the elapsed time from the beginning of video streaming session to the time which packets are started to be received by requester nodes. After every node start to receive video packets, initial waiting time is no longer important.

The reason of observed tree height with algorithm proposed in [13] is higher than our proposed tree is that nodes may not use their full capacity to connect children in [13]. In other words, in Fei's algorithm, the capacity of a node is an upper limit but not a lower limit and because of this approach some nodes may have no children. This turns out to be an increase in tree height in terms of hop count.

In another experiment set, a network topology with 500 nodes having different capacities was constructed with GT-ITM module and simulation results were observed with new test set. Every node that receive video packets, immediately forward them to their children. If the bandwidth capacity is not adequate for sending the video in received quality, then the node drops the video layers until reaching the compatible bitrate. 50% of the nodes have capacity of 1, 20% of the nodes have capacity of 2, 15% of the nodes have capacity of 3, 1% of the nodes have capacity of 4 and finally 5% of the nodes have capacity of 5. The upload bandwidth of the nodes having more than capacity 1 was set as 1000 Kbps per each child, for example, if a node has capacity of 3, then 3 children can be connected to it and its upload bandwidth equals 3000 Kbps. The upload bandwidths of nodes having capacity of 1 are classified as follows: the upload bandwidth of 50% of these nodes is 500 Kbps, 30% of them

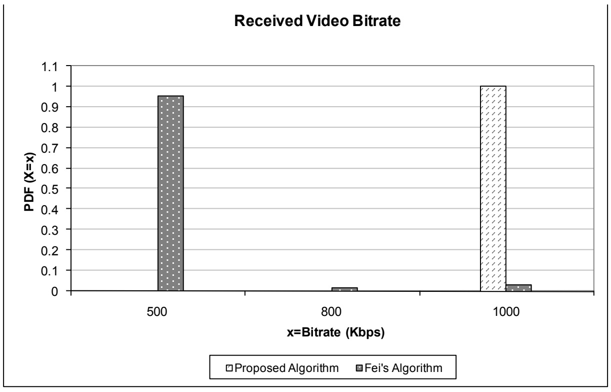


**Figure 5.** Comparative PDF Graph of Received Video Bitrate

800 Kbps and 20% of them is 1000 Kbps. In the experimental setup, the maximum number of transmission bitrate that a node can send to its children was set as 1000 Kbps. If a node has upload bandwidth of 5000 Kbps, it can send video packets at 1000 Kbps unless the upload bandwidth of its parent is less than 1000 Kbps. On the other hand, if the parent of a node has an upload bandwidth of 800 Kbps, then the node can receive video packet at a bitrate of 800 Kbps at most. This property is one of the disadvantages of multicast trees as it is mentioned in the first section.

Weight table values proposed in this study allows the nodes with higher capacity to be placed near root in the tree. This approach provides that the received bitrate of the nodes in the tree is maximized as it can be seen from the statistical graph given in figure 5. According to test results, all nodes in the systems received video at bitrate of 1000 Kbps, which is the maximum bitrate a node can receive in our system. But, with the same set of experiments and the network topology, 95% of nodes received video at 500 Kbps in the system proposed by [13]. This is due to the fact tha,t in [13], there are some nodes having limited upload bandwidth located near root in the tree due to their short delay to root. 2% of nodes received bitrate at 800 Kbps and 3% of nodes received bitrate at 1000 Kbps with the proposed system given in [13].

# CONCLUSION

In this study, we proposed a novel multicast tree construction algorithm for peer to peer video streaming applications. Simulation results show that the height of constructed multicast trees is optimized in the proposed system. Therefore, the probability of packet loss is reduced since the average number of links from source to destination nodes is optimized.

Proposed multicast tree construction algorithm provides higher bitrates since the higher capacity nodes are located near root. Comparative test results also show that our system has significant performance over another similar system proposed in the literature.

Future work plans include adding fault tolerance features to the system and observe the performance of proposed algorithm with peer churn.

**Acknowledgment**

This work is supported by Scientific and Technical Research Council of Turkey (TUBITAK) within the scope of the project no: 108M411. The authors thank to TUBITAK for supporting this project.

# REFERENCES

[1] Liu J, Rao SG, Li B, Zhang H. 2008. Opportunities and challenges of peer-to-peer Internet video broadcast. IEEE Special Issue on Recent Advances in Distributed Multimedia Communications. 9(1):11-24.

[2] Prim RC. 1957. Shortest connection networks and some generalizations. Bell Systems Technology Journal. 36:1389-1401.

[3] Kruskal JB. 1956. On the shortest spanning tree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society. 7:48-50.

[4] Chang EJH. 1982. Echo algorithms: Depth parallel operations on general graphs. IEEE Transactions on Software Engineering. 8:391-401.

[5] Gallager RG, Humblet PA, Spira PM. 1983. A distributed algorithm for minimum-weight spanning trees. ACM Transactions on Programming Languages and Systems, 5:66-77.

[6] Zhang X, Zhang Q, Zhang Z, Song G, Zhu W. 2004. A construction of locality-aware overlay network: mOverlay and its performance. IEEE J. Select. Areas Commun. Special Issue on Recent Advances in Service Overlay Networks, 22(1):18–28.

[7] Banerjee S, Bhattacharjee B, Kommareddy C. 2002. Scalable application layer multicast. ACM SIGCOMM, 205-217.

[8] Tran DA, Hua K, Do T. 2004. A peer-to-peer architecture for media streaming. IEEE JSAC Special Issue on Advances in Service Overlay Networks. 22(1):121-133.

[9] Birrer S, Bustamante FE. 2005. Resilient peer-to-peer multicast without the cost. MMCN. 113-120.

[10] Magharei N, Rejaie R, Yang G. 2007. Mesh or multiple-tree: A comparative study of live P2P streaming approaches. IEEE INFOCOM. 1424-1432.

[11] Sayıt MF, Tunalı ET, Tekalp AM. 2009. Bandwidth-aware multiple multicast tree formation for P2P scalable video streaming using hierarchical clusters. IEEE ICIP. 945-948.

[12] Narula SC. 1980. Degree constrained minimum spanning tree. Comput. Ops. Res. 7:239-249.

[13] Fei Z, Yang M. 2007. A proactive tree recovery mechanism for resilient overlay multicast. IEEE/ACM Transactions on Networking. 15(1):173-186.

[14] http://www.cc.gatech.edu/projects/gtitm/

[15] Tunali ET, Sayit MF. Optimal Backup Parent Pools for Resilient Multicast Trees on Peer to Peer Networks. Springer Peer to Peer Networking and Applications. Under review.