Quality Assurance of Streaming Data Dissemination over P2P Network

Yao-Nan Lien, and We-Chung Chiu

Dept. of Computer Science, National Chengchi University, Taipei, Taiwan, R.O.C.

{lien, g9539}@cs.nccu.edu.tw

***Abstract*—**Peer-to-peer architecture is becoming more and more popular for real-time multimedia streaming services recently due to its scalability as compared with client-server model. Peer-to-peer networks are mostly formed in random fashion without a good control on its topology. As the size of network grows, packets may have to travel through numerous links to reach far-end receivers. The longer the path, the higher the packet loss rate and longer transmission delay. This research is trying to find a better topology for multimedia data multicasting which makes the cumulated delay of the most-far-end user be tolerable and the packet loss be minimized. The problem is modeled as a MLDST problem, which is a NP-Complete problem. To solve the problem, we propose a heuristic MLDST algorithm by modifying Dijkstra's single-source shortest-path algorithm. The proposed solution was evaluated on PlanetLab. Experiments show that our algorithm outperforms traditional minimum spanning tree and shortest path spanning tree algorithms.

***Keywords**: P2P, Multicast, IPTV.*

I. INTRODUCTION

As the advance of network technology, many new network services, such as real-time multimedia streaming service (e.g. IPTV), emerge rapidly. Conventional server-client model is no longer adequate to support these kinds of services because of the extremely heavy traffic load they generate and the stringent time requirement they ask. Server-client topology puts entire traffic on a single link connecting the server and all requesting clients. Under this circumstance, the link bears a great burden such that packets may suffer from excessive long delay and, as a consequence of exceeding playback deadline, high packet loss rate. Therefore, server-client model may not be a good option for real-time multimedia streaming service. On the other hand, the new emerging peer-to-peer (P2P) network, originated from BitTorrent distributed file sharing system, is more adequate for real-time streaming services, due to its high scalability as well as its better tolerance of node failure.

Unfortunately, P2P overlay networks are most formed freely without consideration of either balance of peer load or depth of the spanning tree. Furthermore, the popularity of error prone wireless links is increasing rapidly recently such that not only delay time, but also packet loss rate, must be taken into consideration. Whenever the size of the network grows enormously, number of long paths and overloaded peers, accompanying with long transmission delay and high packet loss rate, increases as well. This problem has been well studied and a large amount of researches has been proposed [4,7,9,10,15,17,18]. Spanning trees are widely applied to P2P network since the very nature of itself, such as easy to build and maintain, well constructed for data transport, quick reaction to nodes failures. However, most of these solutions can only take care of one quality parameter such as delay time. In this paper, we propose to build a multicasting tree that can control both packet loss and delay time. Although it is a centralized algorithm, it lays a good foundation for the design of a good distributed implementation.

II. RELATED WORK

There already exists many peer-to-peer solutions [5,8, 10,11,16,18,19] and can be roughly classified into two categories. In mesh-pull based systems, videos are divided into small clips for distribution. A user sends messages to his/her neighbors to request clips of the video. After receiving positive responses, the user retrieves those clips from possessing neighbors. The control messages not only create some traffic overhead, but also induce extra delay. Typical mesh-pull based systems are CoolStreaming [13], PPlive, and Chainsaw [14].

In tree-push based systems, video data distributes in clips as well. Nodes simply receive data from their parents after they demand for it [16]. The overhead caused by large amount of messages is avoided. Typical tree-push based systems are Chunkyspread, Splitstream, ESM, and ZIGZAG [16].

Major characteristics and constraints of both types of systems are summarized as follows.

1. Both types of systems have a much better scalability than client-server systems.
2. Existing solutions do not take into account packet loss rate of links.
3. With respect to a video clip, all participating peers form a spanning tree topology.
4. The topology of the spanning tree is form randomly without any control such that long paths often presented.

As the size of network grows, packets may have to travel through numerous links to reach far-end receivers. The longer the path, the higher the packet loss rate and longer transmission delay. For instance, no one would like to watch a soccer game from the Internet and to see the winning goal few seconds after hearing his/her neighbors' screaming. Therefore, both packet loss rate and delay time must be controlled under respective thresholds.

Most current solutions model a P2P multicasting network into a spanning tree problem. However, traditional spanning tree algorithms have some structural characteristics that may become obstacles on the way to reach our objective. Different spanning tree algorithm constructs trees with different criterions and thus forms unique structure characteristics. The typical two spanning tree algorithm are minimum spanning tree and shortest-path spanning tree.

A minimum spanning tree has a minimum total cost. It often generates a long tail in the tree [18]. A long tail may cause a large hop count and long delay as well as higher packet loss rate. .

As for single-source shortest path algorithms, such as Dijkstra's shortest path algorithm [3], which also generates spanning trees, their objective is to find a shortest path from the source to all other nodes. The node degree in the resulting spanning tree is unbounded. Large node degree will increase the processing time within a node and thus increase total delay.

This paper proposes to model the problem into a Minimum Loss Diameter Spanning Tree (MLDST) problem which can meet stringent delay requirement and minimize the data loss. The rest of this section will briefly review ZIGZAG system, which will serve the benchmark in the evaluation of our solution, and a few constrained spanning tree problems.

2.1 ZIGZAG

ZIGZAG [16] is a single source Tree-push streaming application which had been proved to be height logarithmic and able to bound node degree in a constant. This helps reduce the number of processing hops on the delivery path to each client while avoiding network bottleneck and long end-to-end delay. ZIGZAG organizes members into a hierarchy of bounded-size clusters and builds a multicast tree rooted at a media server.

Peers are organized in a multi-layer hierarchy of clusters and defined recursively according to following rules (where H is the number of layers and $k > 3$ is a constant):

- Layer 0 contains all peers.
- Peers in layer $j < H-1$ are partitioned into clusters of size of $[k, 3k]$. Layer $H-1$ has only one cluster which has a size of $[2, 3k]$.
- A peer in a cluster at layer $j < H$ is selected to be the head of that cluster. This head becomes a member of layer $j + 1$ if $j < H - 1$. The server S is the head of any cluster it belongs to.

The cluster size is upper-bounded by $3k$. The above structure implies $H = \Theta(\log_k N)$ where N is the number of peers. Any peer at a layer $j > 0$ must be the head of the cluster it belongs to at every lower layer.

This administrative organization does not infer a data delivery topology. Cluster members do not receive contents from their heads but from heads of other clusters. A peer, when not at its highest layer, cannot have any link to or from any other peer. It can only link to peers which belong to other cluster at the lower layer, as one may see in the figure. This mapping structure is one of the major contributions of ZIGZAG.

This paper also proved several theorems, including the worst case node degree, the height of the multicast tree and other worst case control overhead. Similar to our consideration that node degree must be bounded, ZIGZAG limits its worst case node degree to be $O(k^2)$ where k is a constant. On the other hand, letting the height of multicast tree to be logarithmic is also a constraint to transmission path length, which proved to be $O(log_k N)$ where N is the number of peers.

2.2 Minimum Diameter Spanning Tree

Minimum Diameter Spanning Tree (MDST) is described in [4] and [7]. Diameter is defined as the longest path from an arbitrary node to any other destination. MDST problem is a variation of MST that it minimizes the diameter of the tree, instead of total edge weight. This problem can be solved in polynomial time $O(n^3)$.

In [1], it proposed a distributed algorithm to find a MDST. The link weights are defined in positive value. The main contribution of this algorithm is that it achieves an efficient time complexity of $O(n)$ and meantime an $O(nm \log n + nm \log W)$ bits communication complexity, where W is the largest link weight in the network, n is the number of vertices, and m is the number of edges.

2.3 Bounded Diameter Spanning Tree

Bound-Diameter Spanning Tree (BDST) and related problems are discussed in [6]. BDST problem is also a variation of MST with total cost and diameter both be bounded. It is NP-complete for any diameter bound larger than 3. BDST is very close what we want except that only one quality parameter can be controlled.

2.4 Minimum Spanning Tree with Bounded Node Degree

Finding a Minimum Bounded Degree spanning tree is also a NP-Complete problem [4]:

Given a graph G (V, E), positive integer $K \leq |V|$, a cost function $c: E \rightarrow R$, find a spanning tree of minimum cost for G in which no vertex has a degree larger than K. This problem remains NP-complete for any fixed $K \geq 2$.

In [15], it proposed a polynomial time algorithm that returns a spanning tree of minimum cost and bounded node degree. Furthermore, it not only set an upper bound on node degree but also a lower bound on node degree. This property harms the data quality when lower bound is large. Although this paper provides a polynomial time algorithm for this problem, it is not adequate for our model because of different objective.

III. PROPOSED APPROACH

3.1 MLDST (Minimum virtual Loss Diameter Spanning Tree)

In order to reduce the damage associated with the long transmission path, the objective of this research is to find a multicast tree for a given P2P IPTV network that demands quality of service. Specifically speaking, we want to build a spanning tree which has minimum data loss rate under several constraints.

"Diameter" of a spanning tree is defined as "the longest path from the root and all other nodes". Delay diameter is the diameter of a spanning tree when link weight is delay. Likewise, loss diameter is the worst loss rate among all peers. However, since loss rate is not addable but multipliable in nature that makes the computation a very complicated task for a graph based algorithm, we furthermore define the following two terms to represent real Loss Diameter in order to simplify algorithmic computation. V*irtual path loss* is simply the summation of the packet loss rate of all links in a path, while *virtual Loss Diameter* is the largest virtual path loss in a tree. Note that virtual Loss Diameter is only an index measuring the goodness of a path. It doesn't represent a real loss such that it is not necessarily bound to 100%. (It is similar to an imaginary part of a complex number.)

By taking one parameter as our objective and other two as the constraints, we propose a new spanning tree construction model.

Although a distributed model is more appreciate to construct a multicasting tree in real network, we model the problem in centralized fashion at current stage. The problem will be extended to distributed version only after we gain a better understanding on the centralized version. The MLDST problem is defined as follows:

Given a graph G (V, E), delay bound D, and degree bound $B > 3$, find a spanning tree T rooted at v_1, such that its virtual Loss Diameter is minimized while its Delay Diameter is no more than D and node degrees are no more than B, where $V=\{v_1, v_2, v_3 \ldots v_n,\}$ is the set of peer nodes and $E=\{e_{i,j} | v_i, v_j \in V\}$ is the set of possible interconnections between pairs of nodes, $d_{i,j}$ is the delay time spent on $e_{i,j}$ and $p_{i,j}$ is the packet loss rate on $e_{i,j}$.

3.2 NP-Completeness of MLDST

(A) MLDST is in NP:

We first show that MLDST $\in$ NP. Assuming that we are given a graph $G(V, E)$, two parameters on each edge, say, delay d and packet loss rate p, and two predefined bound $D>3$ and B. There is a k-nary spanning tree T given, where maximum $d_{1,i} \leq D$, $\forall\ i \in T$ and $k \leq b$. Then we verify this instance by checking if maximum $\Sigma p_{1,i}$, $\forall\ i \in T$ is the minimum amount all possible solutions. The verification algorithm performs in polynomial time.

(B) MLDST is NP-Complete:

As we illustrated in Section 2.3, a BDST (Bounded Diameter Spanning Tree) problem is a NP-Complete problem if the bound of delay diameter is greater than 3. We can reduce BDST to MLDST straightforwardly. Let graph $G'(V',E')$, edge weight $W'=\{ w_{ij} \mid v_i, v_j \in V \}$, a total weight bound C' and a diameter bound B' be a valid instance of BDST, we construct the corresponding instances G of MLDST as follows: $V=V'$, $E=E'$, $D=D'$, a large node degree bound $B' = |V|$, as well as $p_{ij}=d_{ij}=w_{ij}$, for all edges. We can easily proof by contraction that an optimal solution g to G with a minimum virtual Loss Diameter x must be a solution to G'. First, we can easily see that diameter bound B' must be satisfied. Next, g must be smaller or equal to total weight bound C. Otherwise, we can find another solution g' to G' with a total weight $y <= C < x$. In that case, we can use g' to solve G to obtain a solution with a virtual Loss Diameter y, which is a contradiction.

From (A) and (B), we can say that BDST can be reduced to MLDST. As a result, we prove that MLDST is a NP-Complete problem if node degree bound is greater than 3.

3.3 Design Concepts and Objective

According to the problem model describing in previous sections, we designed a heuristic solution for MLDST. Since our objective is to minimize the *virtual Loss Diameter*, we prefer a single-source shortest-path algorithm rather than a minimum spanning tree algorithm, whose objective is to minimize the total cost of the tree, which may create large diameter paths.

Our heuristic algorithm follows Dijkstra's algorithm's footstep. We modify Dijkstra's algorithm by bounding Delay Diameter and node degree and looking for a spanning tree which has a minimum virtual Loss Diameter.

The issues in distributing environment, such as peer churn and membership change, are left behind in proposed solution. Currently, we only focus on the centralized version for the purpose of proof of concept.

3.4 Heuristic MLDST

Our heuristic algorithm is quite simple and easy to understand. Every edge has two network parameters, delay and packet loss rate. While executing a Dijkstra's algorithm, total delay and total packet loss rate of each intermediate path are calculated. We add two constraints to the original Dijkstra's algorithm so that delay diameter and the degree of each node will be constantly examined to meet the constraints. The nodes that exceed Delay Diameter bound will be discarded. If there is any peer that cannot be included in any feasible solution, those peers must seek another tree to join in order to receive rest part of data. (Theoretically speaking, the algorithm must stop for the problem not solvable.)

If a node exceeds the degree limit, the link with high loss rate will first be abandoned. Priority is given to loss rate, instead of delay. That is, we disconnect links with higher loss rate prior to the ones with longer delay. Once a link is disconnected, data must be rerouted to downstream peers.

The resulting MLDST is only responsible for transmitting one piece of data. To receive complete data, many MLDSTs must be constructed for each piece of data. These multicasting trees need not to be fully disjoined. Notice that data partitioning problem is not our concern.

Heuristic MLDST (G, w, s)

```
Initialize-Single-Source(G, s)
    for  $i \leftarrow 1$ to $|V(G)-1|$
        do for each edge $(u, v) \in E[G]$
// use the weight of $(u, v)$ to update current shortest path
            do RELAX$(u, v, w)$
    for each edge $(u, v) \in E[G]$
        do if $d[v] > d[u] + w(u, v)$
            then return FALSE
            do if $d[v] \geq$ Delay Bound
                then return FALSE
                do if Degree$[v] \geq$ Degree Bound
                    then return FALSE
    Return TRUE
```

This algorithm is modified based on Dijkstra's algorithm by adding two constraints into the loop. While algorithm is running, the constraints are verified at the same time to see whether they are both satisfied. The rest part of the algorithm remains the same as the original one.

IV. PERFORMANCE EVALUATION

The proposed Heuristic MLDST algorithm was evaluated using the real network experimental environment, PlanetLab against other tree construction algorithms including minimum spanning tree, shortest-path spanning tree and ZIGZAG tree. The first two trees are used as baseline performance, and ZIGZAG is the competitor of our algorithm.

4.1 Experimental Environment

Due to the fact that some nodes on PlanetLab were aged low performance PCs, high performance peers on PlanetLab were selected to construct our experimental P2P network. One hundred pinging messages were sent to each of carefully selected peers to measure their response time. Peers with shorter delay were given priority to participate in the experiment. Due to the difficulty of managing unstable voluntary Planetlab members, only fifteen peers were selected to participate in the experiment as shown in Table 1. Although the scale is small, the experiment is useful since the performance deviation would be even larger when the network is scaled up.

Table 1 List of Participating Nodes

Node ID	Domain Name
1	Planetlab1.csie.nuk.edu.tw
2	Planetlab1.sfc.wide.ad.jp
3	Pub1-s.ane.cmc.osaka-u.ac.jp
4	Planetlab-1.calpoly-netlab.net
5	Planetlab1.utdallas.edu
6	Planetlab1.postel.org
7	Nodea.howard.edu
8	Pl1.csl.utoronto.ca
9	Planetlab1.cs.cornell.edu
10	Planetlab1.georgetown.edu
11	Planetlab1.utep.edu
12	Vn1.cs.wustl.edu
13	Plgmu2.ite.gmu.edu

14	Planetlab2.eecs.northwestern.edu
15	Planetlab1.cs.stevens-tech.edu

Degree bound was set to 3 while delay bound was set to 1200 ms based on the analysis of measured delay between peers and user tolerance. Given the same set of peers, one spanning tree was built using each of four algorithms.

Since the characteristics of a real network change constantly, only a snapshot of the network was measured. Therefore, this experiment has its own limits. A dynamic version which is beyond the scope of this research will be more appropriate to model the real network. Our experiments were carried out several hours right after the characteristics measurements was performed (i.e. the instantiation of the graph) assuming that the fluctuation of network condition is acceptable. In the production level implementation, the snapshots can be obtained periodically and the computation of the tree can be done automatically right after the snapshot is obtained.

In each experiment, we transmit a MPEG-1 testing video clip, which has a bit-rate of 1150 Kbps, 29.97 fps and sized 352 x 240, from the source peer “planetlab1.csie.nuk.edu.tw” to every other peer in this P2P network for 30 seconds. Quality related metrics, say, PSNR, transmission delay of each packet and total packet loss were measured from the far-end node of the Delay Diameter for the trees generated by all four algorithms.

4.2 Experiment Results

4.2.1 Topologies

Figure 1 shows the result of Heuristic MLDST algorithm on the P2P network we constructed. Node 14, “planetlab2.eecs.northwestern.edu” is the node at the far end of virtual Loss Diameter.

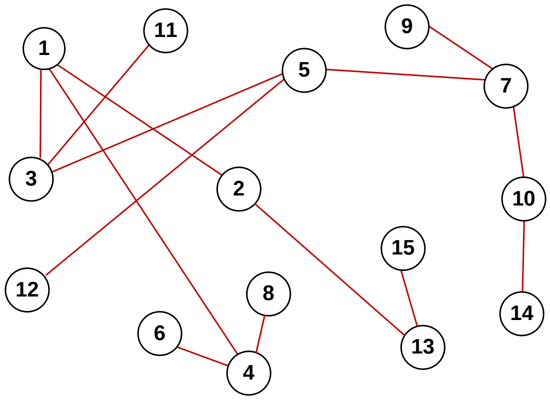

Figure 1 MLDST on PlanetLab

A minimum spanning tree is constructed using Kruskal algorithm as shown in Figure 2. The only parameter that it adapts is delay. At the other end of Delay Diameter, “vn1.cs.wustl.edu” is selected to measure the result.

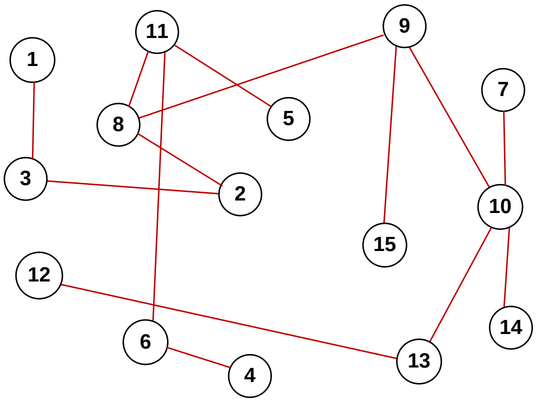

Figure 2 MST on PlanetLab

A shortest path spanning tree was constructed using original Dijkstra’s algorithm as shown in Figure 3. Same as minimum spanning tree topology, shortest path tree only consider delay as link parameter. At the other end of Delay Diameter, “planetlab2.eecs.northwestern.edu” is selected to measure the result.

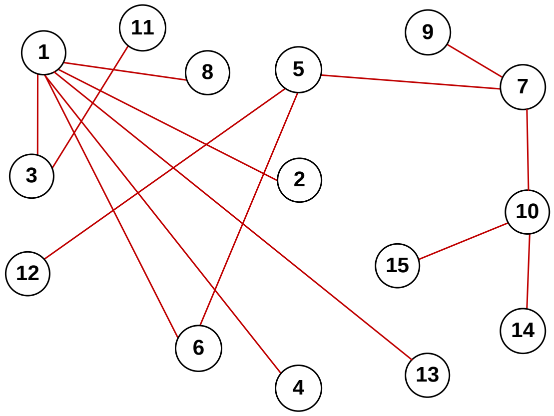

Figure 3 Shortest Path Spanning Tree on PlanetLab

4.2.2 Analysis of Experimental Results

As we can see, MLDST tends to have shorter tail than minimum spanning tree and less number of nodes that have an outstanding node degree than shortest path tree. Notice that longer path leads to larger delay and higher probability of loss. Furthermore, large node degree leads to large processing delay within those nodes. Moreover, increasing delay hinders the data packets to meet the playback deadline at receivers. Figure 4 shows the PSNR of each transmitted frame.

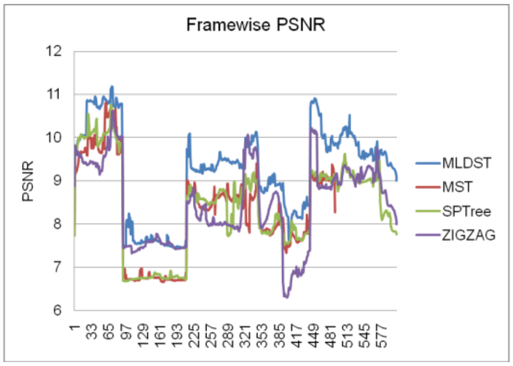

Figure 4 Frame-wise PSNR

Clearly, the measured PSNR values of MLDST are about 10% in average higher than those of other tree construction algorithms. From further examination of the results of MST experiments, we found that the frames of the tail part of the video were all lost during experiment. Notice that the packet loss may be caused by both of network congestion and excessive delay.

Higher PSNR represents better video quality at user end. In summary, MLDST can provide good video quality while total transmission delay is bounded to users' satisfaction.

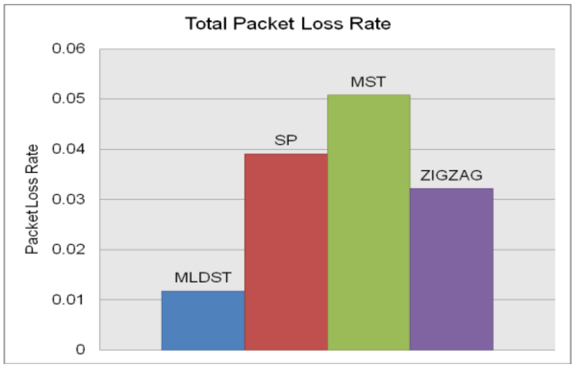

Figure 5 Packet loss rate measured at the end node of virtual Loss Diameter

Figure 5 shows the packet loss rate of each tree. We can see clearly that MLDST shows less packet loss than others and thus can provide higher PSNR as illustrated in Figure 4.

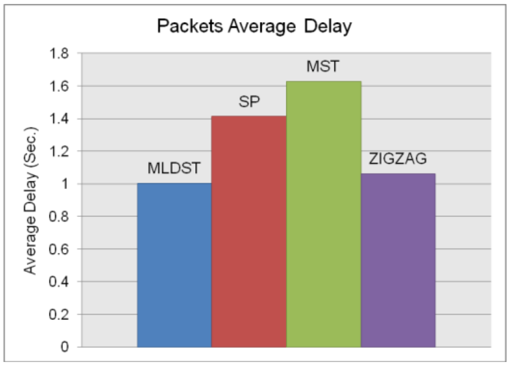

Figure 6 Average packet delay measured at the end node of Delay Diameter

Figure 6 depicts the average delay of each tree at the far end node of the Delay Diameter. Our solution has lowest accumulated delay due to the balanced node degree and the delay constraints.

Since we only take delay as a constraint rather than an objective, a minor superiority in delay is acceptable.

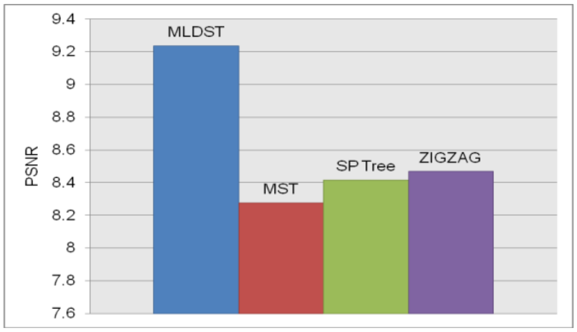

Figure 7 Average frame-wise PSNR

Figure 7 shows average PSNR measured at the far end node of Delay Diameter of all trees. The value is calculated by averaging frame-wise PSNR. MLDST has higher value than others.

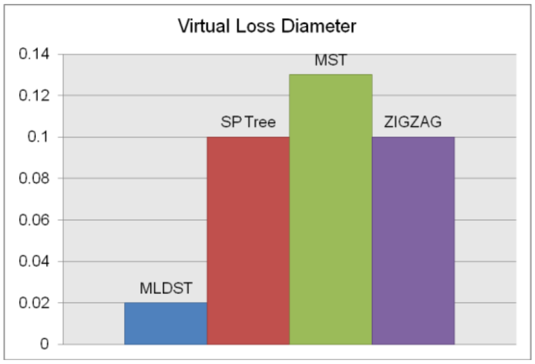

Figure 8 Virtual Loss Diameter

Since we avoid selecting links with high packet loss rate into the multicast tree, the loss diameter of MLDST is explicitly much smaller than others as shown in Fig. 8.

This result further explains why our solution has better data quality.

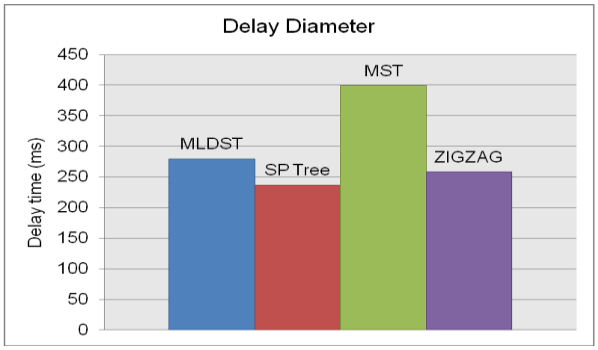

Figure 9 Delay Diameter

Figure 9 shows the Delay Diameter of each participating tree. Note that we take delay as a constraint, higher delay is allowed, as long as the value does not exceed the bound.

V. CONCLUDING REMARKS

Streaming services has been becoming more and more popular in recent years. The reason can be traced back to the explosive growth of Internet. Issues showed up along with this trend and need to be dealt with, such as the user-demanding quality, transmission delay and the fast recovery of lost connection. We present an efficient spanning tree topology to transmit streaming data via peer-to-peer overlay network. After modeling the problem into MLDST, we propose Heuristic MLDST by modifying the single-source shortest-path algorithm, Dijkstra's algorithm, by bounding the Delay Diameter and node degree. Proposed solution depicts a spanning tree with minimum virtual Loss Diameter under delay and node degree constraints. Through several experiment evaluations on PlanetLab, we show that our solution outperforms other tree construction algorithms in video quality. Furthermore, we bound the end-to-end delay to the range of users' satisfaction for the delay requirement.

The issue of peer churn and membership changes is ignored in our research and will be studied in the future.

REFERENCES

[1] Marc Bui, Franck Butelle, and Christian Lavault, "A Distributed Algorithm for Constructing a Minimum Diameter Spanning Tree", *Journal of Parallel and Distributed Computing*, Vol. 64, No. 5, 2004, pp. 571-577.

[2] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Animesh Nandi, Antony Rowstron, Atul Singh, "SplitStream: High-Bandwidth Multicast in Cooperative Environments", *Proc. of the 19th ACM Symp. on Operating Systems Principles*, 2003, pp. 298-313.

[3] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematlk*, vol. l, 1959, pp. 269-271.

[4] M. R. GAREY and D. S. JOHNSON, W. H. Freeman, "Computers and intractability: A guide to the theory of NP-completeness", W. H. Freeman and Company, 1979.

[5] Yu-Hsuang Guo, John K. Zao, Wen-Hsiao Peng, Lin-Shung Huang, Fang-Po Kuo, Che-Min Lin, "Trickle: Resilient Real-Time Video Multicasting for Dynamic Peers with Limited or Asymmetric Network Connectivity", *Proc. of 8th IEEE Int'l Symp. on Multimedia*, Dec., 2006.

[6] Jan-Ming Ho, D.T. Lee, Chia-Hsiang Chang, and C.K. Wong, "Bounded-Diameter Minimum Diameter Spanning Trees and Related Problems", *Proc. of the 5th Annual Symp. on Computational Geometry*, 1989, pp. 276-282.

[7] Jan-Ming Ho, D. T. Lee, Chia-Hsiang Chang and C. K. Wong, "Minimum Diameter Spanning Trees and Related Problems", *SIAM Journal of Computing*, Vol. 20, No. 5, 1991, pp. 987-997.

[8] Xiao-Jun Hei, Yong Liu, Keith W. Ross, "IPTV over P2P streaming networks: The mesh-pull approach", *IEEE Communications*, Vol. 46, No. 2, Feb. 2008, pp. 86-92.

[9] Hung-Chang Hsiao and Chih-Peng He, "A Tree-Based Peer-to-Peer Network with Quality Guarantees", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 19, No. 8, Aug. 2008, pp. 1099-1110.

[10] X. Jia, "A Distributed Algorithm of Delay Bounded Multicast Routing for Multimedia Applications in Wide Area Networks", *Proc. of 6th International Conf. on Computer Communications and Networks*, Sep., 1997, pp. 208-213.

[11] K. Kalapriya, R. Venkatesh Babu, S. K. Nandy, "Streaming Stored Playback Video Over A Peer-to-Peer Network", *Proc. of IEEE Int'l Conf. on Communications*, Vol. 3, Paris, June 2004, pp. 1298-1302.

[12] Bo Li, Hao Yin, "Peer-to-Peer Live Video Streaming on the Internet: Issues, Existing Approaches, and Challenges", *IEEE Communications*, Vol. 45, No. 6, June, 2007, pp. 94-99.

[13] Jiang-Chuan Liu, Sanjay G. Rao, Bo Li and Hui Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast", *Proc. of the IEEE*, Vol. 96, No. 1, January, 2008, pp. 11-24.

[14] Vinay Pai, Kapil Kumar, Tamilmani, Vinay Sambamurthy, Alexander E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast", *Proc. of 4th Int'l Workshop on Peer-to-Peer Systems*, Feb., 2005.

[15] Mohit Singh, Lap Chi Lau, "Approximating Minimum Bounded Degree Spanning Trees to within One of Optimal", *Proc. of the 39th Annual ACM Symposium on Theory of Computing*, San Diego, California, USA, June, 2007, pp.661-670.

[16] Duc A. Tran, Kien A. Hua, Tai Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming", *Proc. of 22nd IEEE INFOCOM*, Vol. 2, 2003, pp. 1283-1292.

[17] DEB Veeravalli, JB Weissman, "A Robust Spanning Tree Topology for Data Collection and Dissemination in Distributed Environments", *IEEE Trans. on Parallel and Distributed Systems,* Vol. 18, No. 5, May, 2007, pp. 608-620.

[18] Vidhyashankar Venkataraman, Kaouru Yoshida, Paul Francis, "Chunkyspread: Heterogeneous Unstructured Tree-Based Peer-to-Peer Multicast", *Proc. of the 14th IEEE Int'l Conf. on Network Protocols*, 2006, pp. 2-11.

[19] Aggelos Vlavianos, Marios Iliofotous and Michalis Faloutsos, "BiTos: Enhancing BitTorrent for Supporting Streaming Applications", *Proc. of 25th IEEE Int'l Conf. on Computer Communications*, 2006, pp. 1-6.