# An Adaptive Approach to Optimize Tree-Based Overlay for P2P Streaming

Ming-Hung Hsu, Tai-Hua Hsiao, and Yu-Ben Miao

Abstract—P2P technologies have been popularly adopted for large-scale applications of content distribution, including file sharing, video on demand and live television, for its advantages in system scalability and deployment efficiency. To further improve the distribution effectiveness of application-laver multicast overlay constructed by P2P technologies, in current years several types of overlay optimization approaches were proposed by focusing on different criteria such as end-to-end delay and bandwidth cost of content server. In this study, we integrate two general criteria of overlay optimization: end-to-end delay and uplink bandwidth, into single parametric criterion named adaptive scoring function. This function is designed from user's perspective of P2P streaming application. Based on the adaptive scoring function, we propose a distributed adaptive overlay optimization approach. Compared with optimization approaches based on other criteria including tree-cost, root-delay, and uplink bandwidth, the experiment results show that the proposed approach significantly reduces end-to-end delay without sacrificing streaming quality, more feasible for P2P streaming applications based on tree-based overlay.

*Index Terms*—Peer-to-peer streaming, tree-based overlay, end-to-end delay, overlay optimization.

## I. INTRODUCTION

Nowadays peer-to-peer (P2P) technology has been widely adopted by various applications such as file-sharing, live TV multicast and video-on-demand (VoD) [1]-[6]. For applications with large-scale demands of content distribution, the major advantage of introducing P2P technology is to reduce the transmission load and the hardware deployment cost of content source servers. To distribute contents in a cost-efficient manner, P2P technologies utilize each participant node's owned resources including disk storage, uplink bandwidth and computational capability, and construct an abstract application-layer topology, overlay, above underlying physical network-layer infrastructure. Application-layer multicast (ALM) is performed on P2P overlays. As a result, the efficiency and effectiveness of P2P-based content distribution relies on constructed overlay.

Based on architectural differences, P2P overlays are typically classified into three categories: tree-based, mesh-based and hybrid-based [7], [8]. In contrast, in a tree-based overlay, content data are pushed from an upstream node to a downstream node, with no need to exchange content bitmap. Nodes participated in a mesh-based overlay may acquire content from multiple neighbor owners, while in a tree-based overlay a child node obtains content from single parent node. Therefore in comparison with tree-based overlays, mesh-based overlays are generally with higher resilience in handling peer churn, the phenomenon that nodes frequently leave and join P2P overlay. On the other hand, tree-based overlays are superior in producing less traffic overhead and in higher supportiveness of guaranteeing bounded path delay [9]. Hybrid-based overlays are proposed aiming to combine the advantages of tree-based and mesh-based overlays, while requiring higher costs in construction and maintenance [10].

Real-time and/or interactive streaming applications are sensitive to delay from stream source to end nodes. To support such applications, P2P overlays have to be optimized to reduce unnecessary delay. Tree-based P2P overlays are potentially feasible for delay-guaranteed applications, but two challenges arise when tree-based overlay optimization is going to be realized. The first challenge is that typical P2P systems are dynamic, lacking of entire information of underlying topology, such as node-to-node link latency. Collecting such information by node-to-node measurement is impractical, and thus overlay optimization must be realized in a distributional and local manner. Secondly, while each node's fan-out (decided by the node's uplink bandwidth) is limited, tree-based overlay optimization problem is equivalent to degree-constrained spanning tree optimization (with respect to minimum delay or cost), which is proven to be a NP-hard problem [11], [12].

In this study, we propose an overlay optimization approach by introducing an adaptive scoring function, ASOT. The ASOT parametrically combines two important criteria, i.e. delay and uplink bandwidth, to measure a node's contribution in a P2P system. For tree-based overlay optimization, we use ASOT to measure the current contribution of an upstream node and the estimated contribution of a downstream node by assuming that the upstream node is replaced by the downstream node. The experiment results show that our approach is more effective than other compared approaches in reducing the average path delay from stream source to end nodes.

This study is organized as follows. In section II, we give a brief review of the work related to P2P overlay optimization. Section III explains the proposed adaptive scoring function and its adoption in distributed overlay optimization. Section IV describes experimental scenario settings and compared approaches, including tree-cost, root-delay, and uplink

Manuscript received November 23, 2012; revised December 24, 2012. This work was supported by the Information and Communications Research Laboratories (ICL), Industrial Technology Research Institute (ITRI), Taiwan, Republic of China (ITRI Grant Project Code B352BR2200).

Ming-Hung Hsu, Tai-Hua Hsiao, and Yu-Ben Miao are with the Information and Communication Research Labs, Industrial Technology Research Institute, Taiwan 310 R.O.C. (e-mail: mhhsu@itri.org.tw; taihua@itri.org.tw; miao@itri.org.tw)

bandwidth. After that, Section V presents the experiment results and discussions. Finally Section VI concludes remarks.

## II. RELATED WORK

In the past years, rich investigations in P2P overlay construction or have been proposed [7], [8], [10],[12]-[20]. Among these investigations, a significant part is addressing tree-based overlay construction and/or optimization. Most of these tree-base optimization approaches focus on minimizing single criterion, such as server bandwidth cost and scalability [15], path delay [16], [17], and tree cost [18], [19]. Tree cost is typically defined as the sum of link costs, and link cost is usually represented by the transmission latency between the two end nodes of the link. From user's perspective, optimizing single one of these criteria, such as tree cost, may not guarantee good quality of user experience. In this study we propose an adaptive and parametric criterion from general user's perspective of P2P streaming.

To realize tree-based overlay optimization in a distributed manner, some basic transformation of tree-based overlay are proposed [7], [18]. Fig. 1 illustrates two of the basic transformations, *switching* and *promotion*. Typically A switching/promotion decision is based on the optimization goal.



Fig. 1. Basic transformations of tree-based overlay: (a) switching: r switches from original parent q to p; (b) promotion of r: r switches from original parent q to original grandparent S, taking the place of original parent q.

## III. ADAPTIVE SCORING FOR DISTRIBUTED TREE OPTIMIZATION

Given a specific P2P streaming application, the quality of user experience is influenced by factors including received streaming bit rate and delay. The idea of our adaptive scoring is treating a child node as a user, and its associated parent node as a relay server. A child node evaluates its obtained relaying service by the path delay from source to it and by the streaming rate it received from the parent node. This section will describe in detail our adaptive scoring for optimizing tree-based overlays, ASOT, and how we utilize it to realize distributed overlay transformation.

# A. Adaptive Scoring Function – ASOT

In a tree-based P2P overlay with stream source S, a child node c evaluates its parent node r according to the combination of two criteria, *delay* and *received streaming quality*, as the following (1):

$$= \left(\frac{\theta}{\max(\theta, \text{Delay}_{S}(c, r, \text{Parent}(r)))}\right)^{\alpha} \times \left(\frac{\text{RSR}(c, r)}{B_{S}}\right), \quad (1)$$

where  $Delay_{S}(c, r, Parent(r))$  indicates the delay from source S to node c through the path consisting of the upstream peers including the parent r and the grandparent, Parent(r), of node c. Note that we assume that each node can identify the path delay from stream source to it by some delay-estimation and/or delay-query methods<sup>1</sup>. The  $\theta$  is the goal of delay, which is determined depending on the application characteristics. For instance, interactive applications such as video conferencing require that the delay bound should be low [21] (ex: 150 milliseconds). RSR(c, r) is the received streaming rate that the child node c received from its parent r. RSR(c, r) also implies the partial *uplink bandwidth* of node r consumed by node c.  $B_S$  is the average bit rate of the stream media out delivered from S. Since node r just provides relaying service to node c, RSR(c, r) is normally lower than  $B_{s}$ . Therefore, the ASOT is designed to parametrically combine two criteria: streaming quality and delay. The parameter  $\alpha$  is introduced to weight the importance of the delay criterion with respect to the streaming quality criterion, when computing ASOT score. The value of  $\alpha$  may be zero or positive real numbers. The larger the  $\alpha$  is, the influence of the delay criterion in the ASOT scoring function increases.

#### B. Tree-Based Overlay Optimization with ASOT

To perform distributed overlay transformation, the basic idea is to use ASOT to evaluate the *contribution* of each node in P2P streaming, and then a node with large contribution will be *promoted* to a position of *higher level*, which indicates getting closer to the stream source in a tree-based overlay. For a specific node r with multiple children, its *current* contribution denoted by *cAS*, is the sum of ASOT scores evaluated from the perspectives of its children, as the following (2):

$$cAS_{S}(r) = \sum_{c \in child(r)} ASOT_{S}(c, r) =$$
$$\sum_{c \in child(r)} \left( \frac{\theta}{\max(\theta, \text{Delay}_{S}(c, r, \text{Parent}(r)))} \right)^{\alpha} \times \frac{RSR(c, r)}{B_{S}},$$
<sup>(2)</sup>

where child(r) indicates the set of child nodes of node r.

Since the ASOT scoring function is based on the concepts of path delay and received streaming rate, the result of evaluating a node's contribution dynamically changes with the node's position in the P2P overlay. Therefore, to

<sup>&</sup>lt;sup>1</sup> In our experimental implementation, each node estimated its path delay as the sum of its parent's delay and the link latency between the two nodes. Each child node periodically inquires its parent to update the delay.

determine if a node should be promoted or not, we have to identify that the node will make more contribution than its parent currently do, if the promotion is taken place. We use the following eAS to estimate a node's contribution after it is promoted:

$$eAS_{S}(r) = \sum_{c \in child(r)} \left( \frac{\theta}{\max(\theta, eDelay_{S}(c, r, GP(r)))} \right)^{\alpha} \times \frac{RSR(c, r)}{B_{S}}, \quad (3)$$

where  $eDelay_S(c, r, GP(r))$  indicates that the node *r* has to estimate the resulted delays of its child nodes if node *r* gets promoted to join its original grandparent, GP(r).

Fig. 2 illustrates an example P2P overlay, where link latencies are shown in brackets with parent-child links (solid arrows). By assuming that  $\theta$  is 150ms and  $\alpha$  is 1, we use Fig. 2 to clarify the procedures of using ASOT for distributed overlay optimization as follows:

1) Child nodes (ex:  $c_1$ ,  $c_2$  and  $c_3$ ) periodically and respectively report their RSR and link latency information to their parents;

2) The parent node (ex: node *r*) calculates its cAS score according to the reported information from the children;

3) The parent node (ex: node r) sends a query to its grandparent (i.e. node q) to inquire the grandparent's path delay, and meanwhile estimates the link latency between the two nodes;

4) The parent node (ex: node r) estimates the resulted delays of its children if it gets promoted, based on the information of its grandparent's delay (i.e. node q), the link latency between it and its grandparent, and the link latencies between it and its children;



Fig. 2. Example of introducing ASOT for overlay optimization: solid arrows represent original flow of P2P streaming; dotted arrows represent messages needed for calculating cAS/eAS of node r. This example exhibits that node r will be promoted to replace node p.

5) The parent node (ex: node r) estimates its eAS based on the reported RSR information from its children and the estimated delay of its children;

In Fig. 2, the promotion of node r to replace its parent p will

be performed if two constraints are satisfied: (a)  $eAS_S(r) > cAS_S(r)$ ; (b)  $eAS_S(r) > cAS_S(p)$  (which is assumed to be 1.75 in Fig. 2); (c) node *r* has at least one free branch to accept the join request from the replaced node *p*. Note that Fig. 2 does not show the query message from node *r* to node *p* for inquiring the cAS of *p*. The messages for realizing the promotion of node *r* are also hidden for clarity.

## IV. EXPERIMENTAL SETUP

#### A. Settings of Experimental Scenario

All experiments were conducted with the settings described below. When a node attempted to join/rejoin the P2P overlay, it would consult a rendezvous point to obtain a list of candidate parents, and then randomly selected a parent to join. The selected parent might reject the join request if it had no available fan-out, and the rejected node had to select another parent. A node's maximum fan-out is the ratio of its uplink bandwidth and the average streaming bit rate  $B_s$ . We set  $B_s$  to be 300 Kbps. Each node's uplink bandwidth belongs to one of the four classes: {50Kbps, 350Kbps, 950Kbps, 1550Kbps}. For nodes with 50Kbps, the maximum fan-out is 0, and these nodes are supposed to be *free-riders*. Each node's uplink bandwidth is randomly initialized according to the probability distribution as {30%, 30%, 30%, 10%}, respectively. That means, only 10% of the peers are randomly picked to have the uplink bandwidth of 1550 Kbps and the corresponding maximum fan-out is 5. The maximum fan-out of the stream source is also set to 5.

The condition that peer churns periodically is also simulated in the experiments. As time goes by a cycle of 10 minutes, four percent (4%) of the nodes in the overlay are randomly picked to be churning, i.e., leaving the overlay. The 4% churning nodes would join the overlay again in the next cycle. While the churning of upstream peers would cause downstream peers unable to receive stream, a downstream node would consult the rendezvous point to rejoin the overlay when it observed its current received streaming quality was lower than 0.5.

Each node in the overlay periodically performs optimization evaluation to decide if a transformation of switching or promotion should be performed. The interval is 5 minutes that each node periodically evaluates candidate parents in scope. The scope of a node is up to 2 hops by local region probing. As a result, each node can discover other nodes within three links in the topology. This scope enables a child node to replace the position of its uncle if the child evaluates that the promotion should be performed.

To reasonably simulate the link latency between any two nodes, we suppose that all the nodes are averagely distributed on a 2-dimension plain. The link latency between two nodes is linearly proportional to the Euclidean distance between their 2-dimension coordinates. The minimum, the average, and the maximum node-to-node latencies are 1 ms, 48ms, and 150 milliseconds respectively. Besides, each node estimated its path delay as the sum of its parent's delay and the associated link latency with its parent. Each node also periodically inquires its parent to update its path delay. When there are no exceptional statements, the number of experimented nodes is 900, and the parameter  $\theta$  of ASOT is set to 150 milliseconds. We implemented the peer behavior described in this section in *C* language.

## B. Compared Approaches

Target(r)

Given the above scenario settings, we compared our approach, ASOT, with another four optimization approaches, named *No-opt*, *Cost-S*, *Delay-S*, *BW-P*, respectively.

1) No-opt (No optimization): each node performs random selection to select parent when it joins/rejoins the overlay; No overlay optimization mechanisms are introduced.

2) Cost-S (Tree cost + Switching): the goal of this approach is to minimize the cost (i.e. the sum of link latencies) of P2P overlay; Node r would probe the costs between it and its local candidate parents, Local(r), to choose a target Target(r) to switch to if the latency between r and the target is lower than r and its current parent. Target(r) is chosen based on (4) and note that AvailF(p) is Boolean to indicate if node p has free fan-out to accept more join requests:

$$= \arg_{p \in \text{Local}(r)} \max\left(\frac{1}{\text{Cost}(r, p)} \times \text{AvailF}(p)\right).$$
(4)

3) Delay-S (Path delay + Switching): Node r would inquire the path delays of its local candidate parents, and estimate if switching to a new parent Target(r) reduce its path delay. Target(r) is chosen based on (5).

$$\operatorname{Target}(r) = \arg_{p \in \operatorname{Local}(r)} \max\left(\frac{1}{\operatorname{eDelay}_{S}(r, p, Parent(p))} \times \operatorname{AvailF}(p)\right) (5)$$

4) *BW-P* (Uplink Bandwidth + Promotion): this approach aims to promote nodes with large uplink bandwidths; Node *r* would inquire the uplink bandwidths of its parent and uncles, and a target, Target(*r*), is chosen to be replaced by *r* if *r* has a larger uplink bandwidth. Target(*r*) is chosen based on (6). Note that the consideration of Avail(*r*) in (6) indicates the promotion of *r* will not be performed if *r* has no free fan-out to accept join request from Target(*r*).

$$\operatorname{Target}(r) = \underset{p \in \operatorname{Local}(r)}{\operatorname{arg}} \max\left(\frac{1}{\operatorname{UBandwidth}(p)} \times \operatorname{AvailF}(r)\right)$$
(6)

#### V. EXPERIMENT RESULTS

Fig. 3 illustrates the performances of the compared approaches in path delay average over all 900 peers, given different experiment duration. The best performance is obtained with ASOT ( $\alpha$ =1) after a 60 minutes experiment. We also observe that the performance of *No-opt* decreases as the experiment duration increases. Fig. 3 also exhibits that the overlay optimization approaches implementing peer promotion (ex: ASOT and BW-P) are more effective in reducing average path delay.

Fig. 3 also demonstrates that the influence of peer churning may cause that an overly optimization approach (ex; Delay-S) cannot always step ahead to its target criterion.



Fig. 3. Performance comparison of the compared approaches in average path delay given different experiment duration.

Fig. 4 shows the performance comparison in received streaming quality averaged over all peers. In contrast, Fig. 5 illustrates the results of average level. Putting the two figures into together exhibits an important observation: when an optimization approach reaches a lower average level, it also provides higher streaming quality. This observation is interesting but also intuitively reasonable. When the average level of peers increases, the harmful influence of peer churning on the received streaming quality of downstream peers does increase. Fig. 4 demonstrates that ASOT and BW-P are more resilient with peer churning, while the resilience of No-opt decreases as time goes by.



Fig. 4. Performance comparison in average streaming quality, given different experiment duration



Fig. 5. Performance comparison of the compared approaches in average peer level, given different experiment duration

Fig. 6 displays the performance of each approach in average link cost. Obviously, only the Cost-S approach is effective in reducing link cost. While the ASOT is proposed from user's perspective, Cost-S is potentially more preferred by network service provider.



Fig. 6. Performance comparison of the compared approaches in average link cost, given different experiment duration

Fig. 7 shows the performance comparison in average path delay when different sizes of P2P overlay were considered. The experiment duration was 60 minutes. Although the performance difference between the ASOT and the BW-P is small, the ASOT exhibits more potential in reducing average delay when the size of overlay increases. The average delay of No-opt obviously increases with the size of overlay.



Fig. 7. Performance comparison in average path delay when handling different sizes of P2P overlay

#### VI. CONCLUSION

In this study we proposed an adaptive scoring function, ASOT, for tree-based overlay optimization. The experiment results indicate that the ASOT appropriately and parametrically combines two important criteria: delay and uplink bandwidth, from user's perspective.

For future work, we are investigating in the timing of activating overlay optimization mechanisms, to reduce the overhead generated by periodic optimization evaluation. Moreover, we are also interested in introducing the ideas of the ASOT for optimizing mesh-based overlays.

#### REFERENCES

 Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Networking and Applications*, 2008, pp. 18-28.

- [2] The Bittorrent website. [Online]. Available: http://www.bittorrent.com/
- [3] The Coolstreaming website. [Online]. Available: http://www.coolstreaming.us/
- [4] The PPStream website. [Online]. Available: http://www.pps.tv/
- [5] The SopCast website. [Online]. Available: http://www.sopcast.com/
  [6] The TVU networks website. [Online]. Available: http://www.tvunetworks.com/
- [7] S. Tan, A. G. Waters, and J. S. Crawford, "A performance comparison of self-organising application layer multicast overlay construction techniques," *Computer Communications*, vol. 29, August 2006, pp. 2322-2347, doi:10.1016/j.comcom.2006.02.020.
- [8] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast," *Proc. ICDCS*, 2007.
- [9] W. Zhang, Q. Zheng, H. Li, and F. Tian, "An Overlay Multicast Protocol for Live Streaming and Delay-Guaranteed Interactive Media," *J. Network and Computer Applications*, 2012, pp.20-28.
- [10] O. Abboud, K. Pussep, A. Kovacevic, K. Mohr, S. Kaune, and R. Steinmetz, "Enabling resilient P2P video streaming: survey and analysis," in *Proc. of Multimedia Systems*, 2011, pp.177-197.
- [11] S. Y. Shi, J. S. Turner, M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks," in *Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Port Jefferson, New York, USA, 2001.
- [12] D. Ren, Y.-T. Hillman Li, and S.-H. Gary Chan, "On Reducing Mesh Delay for Peer-to-Peer Live Streaming," *Proc. INFOCOM*, 2008, pp.1058-1066.
- [13] S. Ren, E. Tan, T. Luo, S. Chen, L. Guo, and X. Zhang, "TopBT: A Topology-Aware and Infrastructure-Independent BitTorrent Client," *Proc. INFOCOM*, 2010, pp.1523-1531.
- [14] K. Zhang, K. Li and J. Cao, "Peer-To-Peer Streaming Overlay Construction for Low Diffusion Delay," in *Proc. International Conference on Networking and Distributed Computing (ICNDC)*, Oct. 2010, pp. 402-406.
- [15] T. Small, B. Li, and B. Liang, "Outreach: Peer-To-Peer Topology Construction towards Minimized Server Bandwidth Costs," *IEEE Journal on Selected Areas in Communications*, 2007, pp.35-45.
- [16] S. Banerjee, C. Kommareddy, K. Kar, S. Bhattacharjee, and S. Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications," in *Proc. INFOCOM*, 2003.
- [17] Z. Li and P. Mohapatra, "HostCast: A New Overlay Multicast Protocol," in *Proc. IEEE International Conference on Communications (ICC)*, 2003, pp.702-706.
- [18] D.A. Helder and S. Jamin, "End-Host Multicast Communication Using Switch-Trees Protocols," in *Proc. IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGRID)*, 2002, pp.419-424.
- [19] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast To End Users," in *Proc. INFOCOM*, 2002.
- [20] H. Horiuchi, N. Wakamiya, and M. Murata, "Proposal and Evaluation of a Network Construction Method for a Scalable P2P Video Conferencing System," in *Proc. International Conference on Information Networking (ICOIN)*, 2007, pp.669-678.
- [21] A. Rios, A. J. Gonzalez, J. Alcober, J. Ozon and K. Z. Ghafoor. "Conferencing Services in P2P Networks: Trends and Challenges," *Advances in Next Generation Services and Service Architecture*, River Publishers, 2010, ch. 1, pp. 1-20.



**Ming-Hung Hsu** received the B.S. degree in Computer Science and Information Engineering (CSIE) from National Taiwan University (NTU) in 2002 and the Ph.D. degree in CSIE from NTU in 2011, with the major field of study in information retrieval and natural language processing.

From 2011 to the present day, he was with the

research area of networking communications in the Information and Communications Research Laboratories (ICL) at Industrial Technology Research Institute (ITRI), Taiwan.