REDUCING END-TO-END TRANSMISSION DELAY IN P2P STREAMING SYSTEMS USING MULTIPLE TREES WITH MODERATE OUTDEGREE

Jeonghun Noh, Aditya Mavlankar, Pierpaolo Baccichet, and Bernd Girod

Department of Electrical Engineering, Stanford University
350 Serra Mall, Stanford, CA 94305, USA
Email: {jhnoh, maditya, bacci, bgirod}@stanford.edu

ABSTRACT

We propose an overlay consisting of multiple trees with moderate outdegree to reduce end-to-end transmission delays in P2P media streaming systems. In real-time media streaming, lower end-to-end delays lead to less waiting time before playback and hence improve interactivity. A theoretical analysis of degree-bounded trees reveals that an optimal number of multiple trees can be chosen by considering the trade-off between the total propagation delay and the queueing delay experienced at intermediate peers. A distributed protocol is presented that allows peers to build multiple degree-bounded trees. From extensive packet-level simulations, we observe that the worst end-to-end transmission delay is minimized when the peer's outdegree, or fan-out, is between 4 and 6 for realistic simulation parameters. This matches well with the predictions from our analysis.

1. INTRODUCTION

For peer-to-peer (P2P) video multicast, an overlay network is often constructed among end users (peers) at the application layer. Tree-based overlays [1, 2, 3] have been a popular choice because a tree structure spans all peers, systematically avoiding the delivery of duplicate packets. In this approach, either one or multiple complementary spanning trees are constructed for data delivery. MutualCast [4] and Dynamic Skip List [5] construct overlays with the largest outdegree and the smallest outdegree, respectively. MutualCast constructs as many trees as the number of peers with non-zero uplink capacity, with a single node forwarding data to all the other nodes in each tree. Dynamic Skip List (DSL) constructs a linked-list type of overlay among peers to provide a quick and scalable video-on-demand service with "trick-modes." The constructed overlay is a single tree with outdegree one. Both MutualCast and Dynamic Skip List can be shown to exhibit linear increase of delay with the number of peers. For delay-sensitive applications, such as live video streaming, the linear increase of the end-to-end transmission delay may aggravate playback latency.

In this paper, we develop the model for the worst end-to-end delay for a system with one or multiple degree-bounded trees. Through the analysis and experiments, we demonstrate:

- An optimal number of multiple trees is obtained by considering the trade-off between the total propagation delay and the queueing delay experienced at intermediate peers.
- Multiple multicast trees with moderate outdegree achieve the lowest end-to-end transmission delay.

An early work that tackles a similar problem of minimizing end-to-end delay is [6]. In this work, the authors employ rateless codes to ease coordination of the multiple streams from parent peers. As peers do not belong to an explicitly constructed tree, peers transmit media streams at arbitrary bitrates to their child peers. The average delay in [6] is modeled as the weighted average of the end-to-end delays of multiple paths. In our study, the system constructs multiple multicast trees in order to deliver the same bitrate of disjoint, complementary streams. In addition, we are interested in minimizing the worst end-to-end delay because the video playout latency is significantly affected by the path with the worst delay.

The remainder of the paper is structured as follows. In Section 2, we introduce degree-bounded trees, followed by an analysis of its end-to-end transmission delay. Section 3 provides the protocols and algorithms for distributed construction of degree-bounded trees. Experimental results follow in Section 4.

2. ANALYSIS OF DEGREE-BOUNDED TREES

2.1. Preliminaries

When two nodes in the tree are directly connected, the node closer to the root of the tree is the *parent* of the other node. Every parent node, apart from the source node, is also called an intermediate node. A degree-bounded tree is a tree whose nodes have a limit on their outdegree. In our system, we used multiple degree-bounded trees, where peers are only allowed to have child peers in one of the trees. A similar restriction in tree construction is found in [7]. The tree in which a peer is allowed to have child nodes is called its *contributory tree*. To the peer, the other trees are *noncontributory trees*. When N_t trees are constructed in the system, one contributory tree and $N_t - 1$ noncontributory trees are assigned to each peer.

For our analysis, we assume that each peer contributes the same amount of uplink bandwidth R to the system. This results in multiple regular degree-bounded trees where all intermediate nodes have the same number of child nodes. The number of trees, N_t, can take on values from 1 to N, where N is the number of peers. In the next section, we show the relationship between N_t (which we choose equal to the maximum outdegree of nodes) and the delay performance.

2.2. Modeling Worst End-to-End Delay

At the source peer, the video is encoded at a constant bitrate R. A fraction of the nodes serve as intermediate nodes in each tree, forwarding data from the source to the rest of the nodes. The number of intermediate nodes in a tree is determined by the outdegree and the number of peers in the system. This in turn determines the tree height, which is defined as the maximum number of hops from the source to a node.

When all the intermediate nodes have N_t child nodes, a tree of maximum depth h can contain up to $\sum_{i=1}^{h} N_t^{i-1}$ nodes. Then, the number of nodes N and tree height h meet the following condition

$$\sum_{i=1}^{h-1} N_t^{i-1} < N \leq \sum_{i=1}^{h} N_t^{i-1}. \tag{1}$$

In case of $N_t = 1$, we have $h = N$ (due to the chain formation). For $N_t > 1$, we manipulate the second inequality, yielding

$$h \geq \log_{N_t}\left(N(N_t - 1) + 1\right). \tag{2}$$

The first inequality in (1) is manipulated similarly. Then, together with the condition that h is a nonnegative integer, h is expressed as

$$h = \left\lceil \log_{N_t}\left(N(N_t - 1) + 1\right) \right\rceil. \tag{3}$$

Eqn. (3) indicates that the tree height decreases when the number of trees N_t increases.

Figure 1 illustrates a tree with outdegree four. S represents the root node, and P_1 and P_2 represent intermediate nodes. Peer P_h is a leaf node with depth h. The end-to-end delay from S to P_h, which is the worst delay in the system, is denoted as T_{DB}. The worst end-to-end delay for a system is defined as the maximum of delays that packets experience while traversing from the source to each destination. The delay at each hop is comprised of three components: propagation delay, queueing delay, and transmission delay.

Propagation delay is the time required for a packet to propagate from one peer to another peer. We assume that the propagation delay between any two peers (including S) is identical, denoted by T_p.

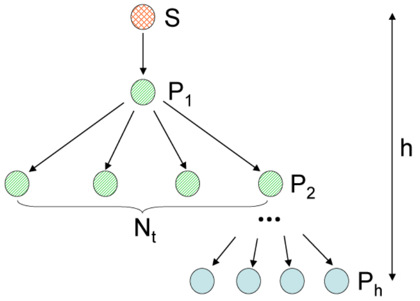

Fig. 1. An example of a degree-bounded tree with outdegree 4 ($N_t = 4$). P_i is a peer at depth i in the tree.

Queueing delay occurs because intermediate nodes duplicate packets to forward to their child nodes. When a packet from S arrives at P_1, it is duplicated to N_t identical packets for P_1's child nodes. The same duplication occurs at each hop along the data path until P_h receives the packet. Assuming that intermediate nodes relay packets to the next hop nodes immediately after they receive them, the queueing delay for the last node served by P_1 is expressed as $(N_t - 1)\frac{L}{R}$, where L is the size of a packet in bits and R is the uplink capacity of the peers. Note that no other traffic is queued up in the intermediate nodes' queues because each node serves as an intermediate node in only one of the trees. For simplicity, we ignore queueing delays that occur in the network. We also assume that there is no other application traffic that involves peers' uplink bandwidth usage.

Transmission delay is the amount of time needed to place a packet onto the network. The transmission delay of a packet with size L is $\frac{L}{R}$.

Putting together these three delay components, the worst end-to-end delay T_{DB} is given as

$$T_{\text{DB}} = hT_p + L\left(\frac{1}{R_S} + \frac{(h-1)N_t}{R}\right), \tag{4}$$

where h is obtained from (3).

2.3. Interpretation of the Delay Models

Figure 2 shows the model prediction of the worst end-to-end delays for systems with 2, 4, 6, 8 and 12 multiple trees. In the figure, the worst delay increases logarithmically with the number of peers in the system. Note that N_t controls the ratio between the propagation delay along the delivery path and the queueing delay at peer queues. On the contrary, for two extreme cases, $N_t = 1$ and $N_t = N$, we observe that the end-to-end delay increases linearly with the number of peers, also depicted in Figure 2.

When $N_t = 1$, nodes are chained together in the overlay because each peer can have only one child peer. Let T_{Chain}

denote the worst end-to-end delay for this chain type of overlay. The length of the chain is equal to the number of peers N. Since each peer has a single child peer, no queueing delay occurs at peer queues. Thus, T_{Chain}, experienced by the peer positioned at the end of the chain, is

$$T_{\text{Chain}} = T_p + \frac{L}{R_S} + (N-1)\left(T_p + \frac{L}{R}\right). \tag{5}$$

When $N_t = N$, each tree is of height 2 with outdegree $N_t - 1$, resulting in the shortest tree with largest outdegree. Trees employed in the MutualCast system [4] are an example of this case. In MutualCast, the media source first sends data packets to its direct child peers for each of the N trees. Then, the peers forward the received packet to the rest of the peers. The worst end-to-end transmission delay T_{MC} of MutualCast is computed as

$$T_{\text{MC}} = 2T_p + \frac{L}{R_S} + (N-1)\left(\frac{L}{R}\right). \tag{6}$$

These two extreme cases of delay demonstrate that when the number of trees is small, queueing delay incurred at each intermediate peer is small. However, packets experience long propagation delay due to large tree depth. When the number of trees is large, the total propagation delay along the path is small. However, packets experience large queueing delay due to high outdegree.

Figure 3 illustrates the relationship between the number of multiple trees (equal to outdegree) and the end-to-end delay. Regardless of the group size, the minimum delay is achieved with around 4 to 6 multiple trees.

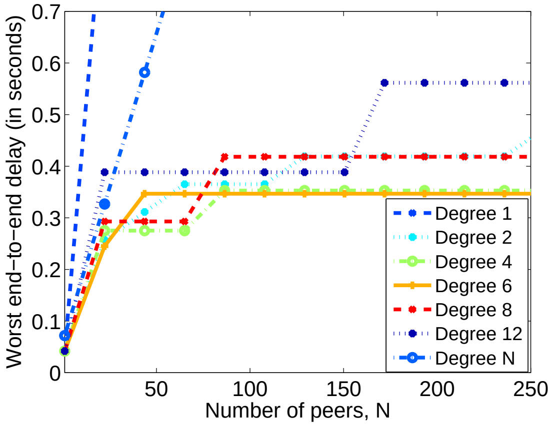

Fig. 2. Worst end-to-end delay under different number of peers. The number of trees is equal to outdegree except for degree N, for which the number of tree is equal to the number of peers. ($L = 1500$ Bytes, R_S = R = 1 Mbps, $T_p = 30$ ms)

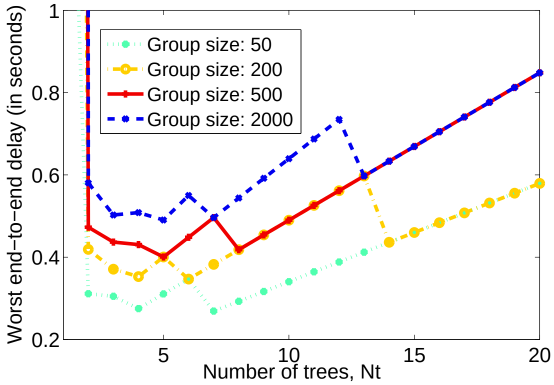

Fig. 3. Worst end-to-end delay under different number of trees. The group size is the number of peers in the system. The number of trees is equal to outdegree. ($L = 1500$ Bytes, R_S = R = 1 Mbps, $T_p = 30$ ms)

3. BUILDING DEGREE-BOUNDED TREES

When a new peer X contacts the source peer S, it is assigned a contributory tree in which it can have child nodes. It also receives a list of peers from which to probe for connections. Based on the probe replies it receives, X selects parent peers for each tree as follows. For its noncontributory trees, it selects parent peers using a common parent selection algorithm. A simple heuristic to choose the parent for the connection is to use metrics, such as available bandwidth or the number of hops to the source peer. For its contributory tree, on the other hand, X attempts to select a position where it can be placed closer to S than any other noncontributory nodes based on the probe replies it receives. When X finds a noncontributory node whose distance to S is equal to or smaller than that of at least one contributory peer, X swaps its position with the noncontributory node. This action, called *peer swapping*, rearranges peers while degree-bounded trees are built incrementally with incoming peers. If all the probed noncontributory nodes are of more distant to S than all the probed contributory nodes, X connects to the contributory node closest to S.

When multiple trees are constructed in a distributed way described above, peers tend to have a limited view of the overlay as they probe only select peers. Nevertheless, peer swapping provides a way of fixing the overlay to guarantee that the outdegree of peers is bounded and all the peers are fully connected to each tree.

4. EXPERIMENTAL RESULTS

We implemented the distributed protocol to construct multiple degree-bounded trees within the NS-2 network simulator [8]. Peers were placed on the randomly chosen edge nodes of the backbone network. The backbone links were sufficiently provisioned with high capacity. The propagation delay of the network links was set to 5 ms. Peers remained in the system until the video session was over. The 10-second long video sequence of *Foreman*, encoded with H.264 at the rate of 280 kbps, was repeated during the 900-second long session. The video stream was divided into N_t disjoint substreams and delivered through as many complementary multicast trees.

We examined the worst end-to-end delay of the degree-bounded trees. Peers computed the average end-to-end delay of packets delivered through each tree, respectively. Peers then chose the worst average end-to-end delay among all the trees. Finally, the mean of the worst end-to-end delay across all the peers was computed over 50 simulations. Figure 4 depicts the average worst end-to-end delay for the system with 4 trees. The delay grew logarithmically as more peers joined, which matches the analysis in the previous section. Around A and B in the figure, however, peers started to experience significantly larger delays. These abrupt increases were caused by an additional depth of peers formed in every tree.

Figure 5 illustrates the worst end-to-end delays depending on the number of trees and the number of peers, where the number of trees is equal to peers' outdegree. The worst end-to-end delay for one tree, where peers form a chain, exhibits the largest delay. Depending on the number of trees (or outdegree), the abrupt change in delay occurs with a different number of peers. When the group size exceeds 100, the systems with 4 to 6 trees demonstrate the smallest delay. Similar patterns were predicted by the analysis presented in Section 2.

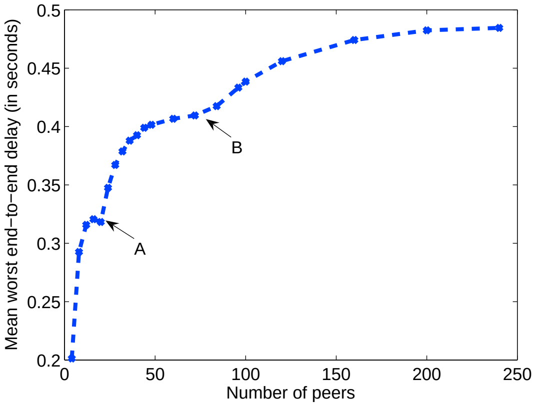

Fig. 4. Mean worst end-to-end delay for the system with 4 degree-bounded trees.

5. CONCLUSION

We proposed an overlay of multiple trees with moderate outdegree to reduce end-to-end delay. The worst end-to-end delay of degree-bounded trees grows logarithmically with the number of peers only when the outdegree is moderate. Our experimental results show that systems with 4 to 6 trees results in the best end-to-end delays for realistic simulation parameters. The experimental results match our delay models which reveal that the outdegree of peers controls the trade-off between the total propagation delay and the queueing delay at peer queues.

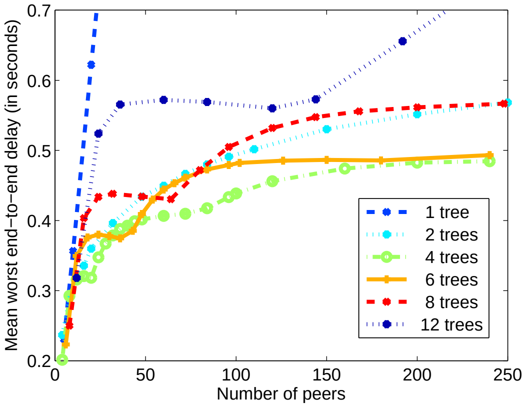

Fig. 5. Mean worst end-to-end delays of degree-bounded trees for different number of trees and peers. The number of trees is equal to outdegree.

6. REFERENCES

[1] Y. Chu, S. Gao, and H. Zhang, "A case for end system multicast," *Proc. ACM Sigmetrics, Santa Clara USA*, June 2000.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," *Proceedings ACM SIGCOMM*, Aug. 2002.

[3] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *Proc. ACM NOSSDAV, Miami Beach, FL*, May 2002.

[4] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An efficient mechanism for one-to-many content distribution," *ACM SIGCOMM ASIA Workshop*, Apr. 2005.

[5] D. Wang and J. Liu, "Peer-to-Peer Asynchronous Video Streaming using Skip List," *2006 IEEE International Conference on Multimedia and Expo*, July 2006.

[6] C. Wu and B. Li, "Optimal peer selection for minimum-delay peer-to-peer streaming with rateless codes," *Proc. ACM Workshop Advances in Peer-to-Peer Multimedia Streaming (P2PMMS 05)*, Nov. 2005.

[7] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient Peer-to-Peer Streaming," *IEEE ICNP 2003, Atlanta, GA, USA*, Nov. 2003.

[8] "The Network Simulator - ns-2," *www.isi.edu/nsnam/ns*.