Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches

Nazanin Magharei, Reza Rejaie[‡]
University of Oregon
{*nazanin,reza*}*@cs.uoregon.edu*

Yang Guo
Thomson Lab
Yang.Guo@thomson.net

***Abstract*—Existing approaches to P2P streaming can be divided into two general classes: *(i) tree-based approaches* use push-based content delivery over multiple tree-shaped overlays, and *(ii) mesh-based approaches* use swarming content delivery over a randomly connected mesh. Previous studies have often focused on a particular P2P streaming mechanism and no comparison between these two classes has been conducted. In this paper, we compare and contrast the performance of representative protocols from each class using simulations. We identify the similarities and differences between these two approaches. Furthermore, we separately examine the behavior of content delivery and overlay construction mechanisms for both approaches in static and dynamic scenarios. Our results indicate that the mesh-based approach consistently exhibits a superior performance over the tree-based approach. We also show that the main factors attributing in the inferior performance of the tree-based approach are *(i)* the static mapping of content to a particular tree, and *(ii)* the placement of each peer as an internal node in one tree and as a leaf in all other trees.**

I. INTRODUCTION

Using Peer-to-Peer overlay has become an increasingly popular approach for streaming live media over the Internet due to its potential scalability and ease of deployment. This approach is generally referred to as *P2P streaming*. In P2P streaming, participating end-systems (or peers) actively contribute their resources (mainly outgoing bandwidth) by forwarding their available content to their connected peers. Since the aggregate available resources in this approach organically grow with the user population, this approach can potentially scale with the number of participating peers in a session.

Existing approaches for live P2P streaming can be generally divided into two classes: *tree-based* approaches and *mesh-based* approaches. The tree-based P2P streaming approach expands on the idea of end-system multicast [1] by organizing participating peers into multiple diverse trees. Then, each description of a Multiple Description Coded (MDC) content is pushed through a separate tree [2], [3]. The mesh-based P2P streaming approach is inspired by file swarming mechanisms (such as BitTorrent) where participating peers form a randomly connected mesh and employ a swarming content delivery mechanism over a recent window of content [4]. However, the limited availability of new content in live streaming coupled with the notion of "quality" for the delivered stream (*i.e.*, number of descriptions) introduce new dimensions in the design and evaluations of mesh-based P2P streaming. Most of the previous studies on P2P streaming have focused on a particular mechanism and evaluated certain aspects of its performance. However, to our knowledge, the performance of these two classes of P2P streaming approaches have not been directly compared.

In this paper, we compare and contrast the performance of tree-based and mesh-based P2P streaming approaches. We provide an overview of a representative protocol in each class and expose their similarities and differences. We then compare the performance of tree- and mesh-based approaches using the representative protocols in two steps as follows: First, we examine the performance of content delivery in these approaches over a properly connected and static overlay. We present the notion of "delivery tree" for individual packets in the mesh-based approach which enables us to clearly compare the behavior of content delivery in tree- and mesh-based approaches. Our results show that the tree-based approach is sensitive to the ratio of peer bandwidth to description bandwidth. This implies that the tree-based approach has a sweet spot for peer bandwidth where it can effectively utilize available resources and provide the desired quality. We also examine the effect of peer degree (*i.e.*, number of trees), bandwidth heterogeneity, and peer population. Our evaluations reveal that swarming content delivery in mesh-based approach exhibits a superior performance across a wide range of scenarios. This is primarily due to the ability of the swarming mechanism to minimize the impact of a low bandwidth connection on the connected child peer by providing the required content through other parents. In contrast, the tree-based approach requires each description of the content to be delivered through a particular tree which extends the adverse effect of a low bandwidth connection to all its downstream peers on that tree.

Second, we investigate the ability of both approaches to cope with churn from two angles: *(i)* the performance of content delivery on a distorted overlay, and *(ii)* the cohesion of the overlay structure under persistent churn. We model a distorted overlay by removing a random subset of participating peers from a properly connected overlay without repairing it. We show that the swarming delivery in the mesh-based approach can effectively utilize available resources over distorted overlays whereas the tree-based approach exhibits poor performance in such circumstances. We also quantify the cohesion of the overlay under churn using three metrics: ancestor changing rate, the average degree of connectivity, and the frequency of deadlock events (only in the tree-based approach). Our results indicate that peers always experience a higher degree of stability in the mesh-based approach. More interestingly, in the mesh-based approach, the longer a peer remains in the system, the higher the degree of stability

---

[‡] This material is based upon work supported by the NSF CAREER Award CNS-0448639.

it experiences, and thus the higher the delivered quality it receives.

In summary, this paper makes two important contributions: *(i)* leveraging the notion of delivery tree for individual packets, we identify the key differences between mesh-based and tree-based approaches to P2P streaming. This in turn sheds an insightful light on the inherent limitations and potentials of these two approaches; *(ii)* we also identify the underlying causes for the observed differences between tree- and mesh-based approaches.

The rest of this paper is organized as follows: In Section II, we present an overview of representative protocols for both approaches and their key design components. We also elaborate on their similarities and differences, and present the notion of delivery tree in the mesh-based approach. Section III compares the performance of content delivery in both approaches over properly connected and static overlays. Ability of these approaches to cope with churn is examined in Section IV. Finally, Section V concludes the paper.

II. TREE- VS MESH-BASED P2P STREAMING

In this section, we present an overview of tree- and mesh-based P2P streaming approaches and identify their similarities as well as differences. This provides the required background for this work and motivates our evaluation methodology. We assume all pairwise connections for data delivery between peers are congestion controlled in both tree- and mesh-based approaches. This ensures that these approaches behave in a network-friendly fashion and achieve proper bandwidth sharing among incoming (and outgoing) connections to (from) individual peers. We also assume that both approaches leverage *Multiple Description Coding (MDC)* to accommodate the bandwidth heterogeneity among participating peers. In MDC, a stream is encoded into multiple sub-streams called *description*. Each description can be independently decoded. Furthermore, receiving multiple unique descriptions results in a higher quality. This enables individual peers to receive the proper number of descriptions proportional to their aggregate incoming bandwidth in order to maximize their received quality.

A. Tree-based P2P Streaming

In the tree-based approach, an overlay construction mechanism organizes participating peers into multiple trees. Each peer determines a proper number of trees to join based on its access link bandwidth. To minimize the effect of churn and effectively utilize available resources in the system, participating peers are organized into multiple *diverse* trees. Toward this end, each peer is placed as an *internal* node in only one tree, and as an *external* (or leaf) node in other trees. Then, each description of an MDC encoded content is delivered through a specific tree. The content delivery is a simple push mechanism where internal nodes in each tree simply forward any received packets for the corresponding description to all of their child nodes. Therefore, the main component of the tree-based P2P streaming approach is the tree construction algorithm.

Tree Construction Algorithm: The goal of the tree construction is to maintain multiple balanced, stable and short trees. In this paper, we use the following central tree construction algorithm that to our knowledge, represents the best practice among existing solutions [2], [3]:

Each peer is placed as an internal node in only one tree and leaf node in other participating trees. When a peer joins the system, it contacts the bootstrapping node to identify a parent in the desired number of trees. To keep the population of internal nodes balanced among different trees, a new node is added as an internal node to the tree that has the minimum number of internal nodes. To maintain short trees, a new internal node is placed as a child for the node with the lowest depth (the first node as we traverse the tree in a breadth-first fashion) that can accommodate a new child or has a child that is a leaf. In the latter case, the new node replaces the leaf node and the partitioned leaf should rejoin the tree similar to a new leaf. When an internal node of a tree departs, each one of its child nodes as well as the subtree rooted at them are partitioned from the original tree, and thus should rejoin the tree. Peers in such a partitioned subtree initially wait for the root of the subtree to rejoin the tree as an internal node. If the root is unable to join the subtree after a certain period of time, individual peers in a partitioned subtree independently rejoin the tree with the same position (as leaf or internal node).

A tree can always accept a new internal node. However, in the presence of churn, a tree could become *saturated* and thus unable to accept any new leaf node. We denote this as a *deadlock event*. A deadlock event occurs when a tree loses a fraction of its internal nodes within a short period of time which reduces the number of leaf nodes that it can accommodate. In such a scenario, the number of internal nodes at different trees becomes imbalanced, where spare slots for leaf nodes are available on other trees but they can not be used to resolve the deadlock of the saturated tree. When a leaf node experiences deadlock, it periodically tries to rejoin the tree until it succeeds.

B. Mesh-based P2P Streaming

In the mesh-based approach, participating peers form a randomly connected overlay, or a *mesh*. Each peer tries to maintain a certain number of parents (*i.e.*, incoming degree) and also serves a specific number of child peers (*i.e.*, outgoing degree). Upon arrival, a peer contacts a bootstrapping node to receive a set of peers that can potentially serve as parents. The bootstrapping node maintains the outgoing degree of all participating peers. Then, it selects a random subset of peers that can accommodate new child peers in response to an incoming request for parents. Note that the pairwise connections in the mesh-based approach can be used for content delivery in both bidirectional or unidirectional fashion. In this study, we only consider uni-directional connections in the mesh-based approach (*i.e.*, connected peers have a parent-child relationship) for two reasons: *(i)* this results in a directed overlay which is very similar to multiple trees and thus facilitates our comparison, and *(ii)* the mesh-based

P2P streaming exhibits better performance over unidirectional overlays [4]. We use PRIME [4] as a representative P2P streaming mechanism for the mesh-based approach in this study.

The mesh-based approach employs the swarming content delivery similar to BitTorrent. The main advantage of the swarming content delivery is its ability to effectively utilize the outgoing bandwidth of participating peers as the group size grows. Swarming content delivery couples push content reporting with pull content requesting. Individual peers periodically report their newly available packets to their child peers and request specific packets from individual parent peers. A parent peer periodically receives an ordered list of requested packets from each child peer, and delivers the packets in the requested order. The requested packets from individual parents are determined by a *packet scheduling* algorithm at each child peer. The packet scheduling algorithm is a key component of a mesh-based P2P streaming mechanism that should achieve the following three design goals: *(i)* effectively utilizing the available bandwidth from all parents peers, *(ii)* pulling a proper number of descriptions (*i.e.*, desired quality) from all parent peers, and *(iii)* ensuring in-time delivery of requested packets. The pattern of content delivery for individual packet through the mesh (*i.e.*, the path that a packet traverses to reach all peers) depends on the collective behavior of the packet scheduling algorithm at individual peers as well as the connectivity of the overlay topology.

Packet Scheduling: In this study, we use PRIME that incorporates the following packet scheduling algorithm: Each peer maintains two pieces of information for individual parents: *(i)* the available packets, and *(ii)* the weighted average bandwidth (*i.e.*, bandwidth budget). Furthermore, individual peers monitor the aggregate incoming bandwidth from all parents and slowly adapt the number of requested descriptions (or their target quality) with the aggregate bandwidth. Given the per-parent information along with the target quality n, the scheduling algorithm is periodically (*i.e.*, once per Δ seconds) invoked to determine a set of packets that should be requested from each parent as follows: First, the scheduler identifies the packets with the highest timestamp that have become available among parents since the last request (during last Δ seconds). The scheduling algorithm requests all of these new packets (up to n descriptions per timestamp) from the corresponding parent(s). Second, the missing packets for each timestamp (up to n descriptions per timestamp) are identified and a random subset of these packets is requested from all parents to fully utilize their bandwidth. The total number of requested packets from each parent is determined by its bandwidth budget. To balance the load among parents, when a packet is available at more than one parent, it is requested from the parent that has the lowest fraction of its bandwidth budget utilized. Clearly, when a parent does not have sufficient number of useful packets for a child peer, the bandwidth of its congestion controlled connection to that child peer can not be fully utilized. We refer to such an event as *content bottleneck*.

C. Similarities & Differences

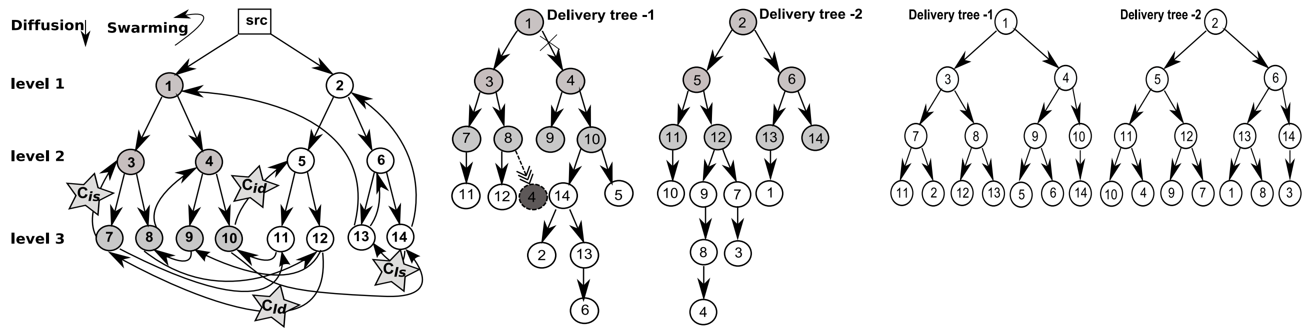
In this subsection, we describe the similarities and differences between two approaches which helps us identify the underlying causes for the observed behavior by each approach in our evaluations.

Similarities: The tree-based and mesh-based approaches have a great deal of similarities as follows: First, while these approaches use different overlay construction algorithms, the overall shape of their resulting overlays is very similar. More specifically, the superimposed view of multiple diverse trees is in fact the same as a directed random mesh. Second, the content delivery in both approaches enable individual peers to receive different pieces of the content.At the peer level, each peer receives content from multiple parents and sends content to multiple child peers in both approaches. At the system level, the collection of edges used for the delivery of a single packet form source to all participating peers form a source-rooted tree in both approaches that we call the *delivery tree*. Third, both approaches require participating peers to maintain a loosely synchronized playout time that is sufficiently (τ seconds) behind source's playout time. This requires τ seconds worth of buffering at each peer which accommodates the diversity of different paths from source in the tree-based approach, and out-of-order packet arrival in the swarming content delivery of the mesh-based approach. The value of τ depends on the maximum hop count from source to different participating peers through the overlay which is a function of peer population and peer degree. For a fair comparison, we assume that both approaches use the same value of τ in comparable scenarios.

Differences: The key difference between the mesh-based and the tree-based approaches is how the delivery tree of individual packet is formed. In the tree-based approach, the delivery tree for all packets of a particular description is the corresponding overlay tree for that description. In essence, the delivery tree of each packet is indeed pinned down by the tree construction mechanism because of the *static* mapping of descriptions to trees. This has an important implication: when the bandwidth of a connection is less than the description bandwidth, the packets for that description can not be "streamed" at a proper rate to all the descendant peers. In contrast, in the mesh-based approach, the delivery tree for individual packets is dynamically shaped as the packet traverses through the overlay. The dynamic formation of the delivery tree enables the mesh-based approach to effectively utilize the available resources. In particular, when a connection has low bandwidth, its descendant peers can still receive their required packets through alternative paths from other parents. The dynamic formation of the delivery tree in the mesh-based approach is essential in understanding its behavior, and it is explained in further details in the following subsection.

D. Delivery Tree in Mesh-based Approach

To derive the delivery tree in the mesh-based approach, we need to present the proper pattern of content delivery over a mesh that maximizes the utilization of outgoing bandwidth



(a) Organized view of a random mesh (b) Delivery trees in Mesh-based (c) Delivery trees in Tree-based

Fig. 1. Organized view of an overlay, with two delivery trees for mesh- and tree-based approaches.

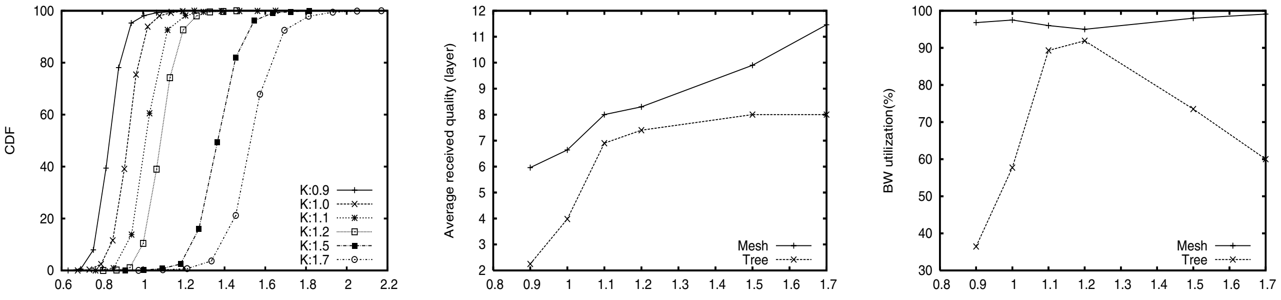
among participating peers. Toward this end, we introduce the organized view of a randomly connected mesh by grouping peers into *levels* based on their shortest distance (in hops) from source through the overlay as shown in Figure 1(a). Peers that are one-hop away from source (source's children) are in level 1, peers that are two hops away from source are in level 2, and so on. The number of levels is equal to the *depth* of the overlay or the maximum distance of a peer from source. To efficiently utilize source's bandwidth, we assume that source should deliver each packet only once. The pattern of delivery for a single packet over an organized mesh should consist of the following two distinct phases in order to maximize the utilization of outgoing bandwidth among participating peers [5]: *1) Diffusion Phase:* Once a new packet becomes available at the source, a single peer p in level 1 pulls the packet during the next interval Δ. Then, all the p's child peers in level 2 pull a copy of the packet in the following interval and so on. All connections from peers in level i to peers in level $i{+}1$ ($i{<}depth$) are used for diffusing new packets through the overlay and thus called diffusion connections. The diffusion connections are shown with straight arrows in Figure 1(a). Since each packet is only delivered once from the source, the subset of peers that receive a packet during its diffusion phase, form a subtree, called *diffusion subtree*. The diffusion subtree consists of a peer in level 1 (as its root) and all of its descendant peers in lower levels. For example, the shaded nodes in Figure 1(a) form a single diffusion subtree. Note that the number of distinct diffusion subtrees in the overlay is equal to the number of peers in level 1 (*e.g.*, two diffusion subtrees in Figure 1(a)). At the end of the diffusion phase of a packet, only a subset of peers that are located on a diffusion subtree have received that packet.

2) Swarming Phase: During the swarming phase, peers on different diffusion subtrees exchange their new packets (or swarm) to contribute their outgoing bandwidth. All the connections from a peer in level i to a peer in level j ($j{\leq}i$) are used for swarming and thus called swarming connections. These connections are shown with curly arrows in Figure 1(a).

The swarming connections can be divided into the following four groups based on the locations of two peers that they connect: *(i)* connecting peers at the bottom of two different diffusion subtrees (C_{ld}), *(ii)* connecting peers at the bottom of the same diffusion subtree (C_{ls}), *(iii)* connecting a peer at the bottom of one diffusion subtree to an internal peer on a different diffusion subtree (C_{id}), *(iv)* connecting a peer at the bottom of one diffusion subtree to an internal peer on the same diffusion subtree (C_{is}). A sample connection from each group is marked with star and proper label in Figure 1(a). The delivery tree of a packet in the mesh-based approach consists of two parts: *(i) the top portion* of the delivery tree that must be the same as one of the diffusion subtrees, *(ii) the bottom portion* of the delivery tree consists of a collection of swarming connections that are extending (or hanging from) the diffusion subtree. Our packet scheduling algorithm implies that different groups of swarming connections can only be attached to the delivery tree at certain locations based on the following rules:

- C_{is} and C_{ls} can be attached at any part of the bottom portion of the delivery tree.
- C_{ls} and C_{ld} can only be attached to C_{ls} or C_{is} type connections. Otherwise, they form an ending branch for the delivery tree.
- C_{id} and C_{ld} can only be attached to the diffusion subtree.
- C_{is} and C_{id} can only be attached as an ending branch of the delivery tree.

Figures 1(b) and 1(c) illustrate two delivery trees in the mesh-based and tree-based approaches for the overlay in Figure 1(a), respectively. We summarize our main points in this section as follows: The delivery tree of individual packets in the tree-based approach is determined by the overlay construction mechanism. As a result, a low bandwidth connection in an overlay tree can limit the rate of data delivery to all of the downstream peers. In contrast, the delivery tree in the mesh-based approach is dynamically determined by the collective behavior of packet scheduling mechanisms among participating peers (*i.e.*, swarming content delivery). This enables individual peers to gracefully cope with a low bandwidth connection by receiving their desired packets from other parents through other paths. For example, if the connection from peer 1 to peer 4 in Figure 1(b) has a low bandwidth, peer 4 (as well as its descendant peers in the diffusion subtree, such as peers 9 and 10) can still receive a subset of packets from other swarming parents (*e.g.*, peer 8). In essence, the dynamic formation of a delivery tree implies that a peer can appear at different parts of the delivery tree for different packets. One side effect of the dynamic formation of the delivery tree in



(a) Distribution of normalized per-connection BW (b) Average received quality (c) BW utilization(%)

Fig. 2. Effect of per-connection bandwidth

the mesh-based approach is their longer depth compared to the mesh-based approach as shown in Figures 1(b) and 1(c).

III. CONTENT DELIVERY IN STATIC GROUP

In this section, we examine the performance of content delivery mechanism in both approaches over a static overlay using *ns* simulations. In our simulations, the physical topology is generated using Brite [6] with 15 AS, 10 routers per AS in top-down mode, and *RED* queue management at all routers. The delay on the access links is randomly selected between [5ms, 25ms]. All pairwise connections between peers employ *RAP* congestion control mechanism. Core links have high bandwidth (4Gbps to 10Gbps) and thus individual connections only experience bottlenecks at the edge. To quantify the utilization of available bandwidth for each connection, we use the following methodology to decouple the available bandwidth from the available content as follows: when a parent experiences content bottleneck and does not have any useful packet to deliver to a particular child, it sends a especially marked packet with the same size to that child. We define the *bandwidth utilization* as the ratio of the number of data packets to the total number of delivered packets. We also define *the average delivered quality* for each peer as the average number of descriptions it receives during a session.

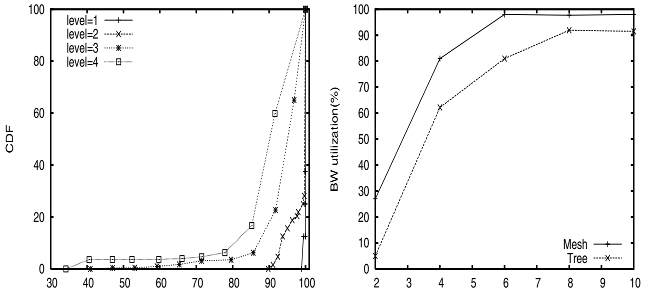
In both approaches, all peers maintain synchronized playout time that is τ seconds behind the source's playout time. The value of τ is selected to be the minimum value that can accommodate in-time delivery of packets for a given population and peer degree. Based on this strategy, we conservatively set τ to 24 seconds in our simulations. We use the following default values for other parameters: each stream has 20 descriptions and all descriptions have the same constant bit rate of 80Kbps (bw_d). Δ is set to 4 seconds. Each scenario consists of 200 homogeneous peers with symmetric bandwidth, and access link bandwidth of all peers is set to $deg \cdot bw_d$ where deg denotes the degree of each peer. Thus, each peer should be able to receive deg descriptions which we refer to as *target quality*. In both approaches, the source degree is equal to the peer degree (deg). Furthermore, source bandwidth is set to the minimum value that is required for the delivery of the desired aggregate quality to the overlay (*i.e.*, the delivered quality to all peers in level 1, collectively). The aggregate delivered quality in each simulation is equal to the quality that peers with the highest bandwidth can obtain.

A. Effect of Per-Connection Bandwidth

We first examine the effect of per-connection bandwidth on the system performance. Since all peers have the same incoming and outgoing degree of deg, by setting the access link bandwidth to $deg \cdot K \cdot bw_d$, we can control the average per-connection bandwidth to be $K \cdot bw_d$. We can vary the access link bandwidth by changing K in order to investigate the effect of per-connection bandwidth on system performance. Figure 2(a) depicts the distribution of per-connection average bandwidth (normalized by bw_d) for different values of K where peer degree is 8. This figure clearly demonstrates that different connections obtain different average bandwidth due to the dynamics of congestion control. As the peer bandwidth increases, the median value of the distribution proportionally increases and it becomes slightly more skewed. The key question is *"whether the distribution of per-connection bandwidth affects the performance of tree- or mesh-based P2P streaming approach?"*

Figure 2(b) presents the average delivered quality as a function of K in both approaches. Figure 2(b) reveals that the average delivered quality in the mesh-based approach is proportionally improved with the peer bandwidth and *can* even exceed the target quality. In contrast, the average delivered quality in the tree-based approach is poor when connection bandwidth is less than or equal to the description bandwidth ($K \leq 1$). As the per-connection bandwidth increases, the average delivered quality reaches the targeted quality of deg descriptions but cannot go beyond this limit regardless of the per-connection bandwidth.

Figure 2(c) shows the average bandwidth utilization across all connections as a function of K. In the mesh-based approach, participating peers achieve high bandwidth utilization (>%95) and can properly adjust the delivered quality for any value of per-connection bandwidth. In contrast, the aggregate bandwidth utilization in the tree-based approach has a sweet spot (at K=1.2) where it reaches %90. However, for all other values of K, it exhibits a significantly lower bandwidth utilization. The poor bandwidth utilization for small values of K is due to extended effect of a single low bandwidth connection on all of its downstream connections. But when the per-connection bandwidth is large, the bandwidth of individual connections significantly exceeds the description bandwidth. This results in the *content bottleneck* since parent peers do



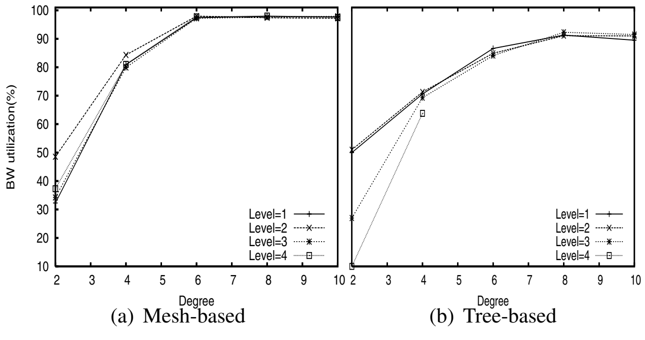
(a) Distribution of per-connection BW utilization based on peers' level (b) Aggregate BW utilization(%)

Fig. 3. Effect of peer degree

not have sufficient useful content to utilize the available bandwidth. *In summary, the tree-based approach has a sweet spot for the ratio of per-connection bandwidth to description bandwidth where high resource utilization and thus high delivered quality is achieved. In contrast, the mesh-based approach can effectively utilize any value of peer bandwidth and deliver a proportionally higher stream quality.* For the remaining evaluations in this section, we set the value of K to 1.2 for the tree-based approach to achieve its best performance. In a nutshell, this implies that our results in this section represent an upper bound for the performance of the tree-based approach.

B. Effect of Peer Degree (Number of Trees)

In this subsection, we investigate the effect of peer degree on system performance. Figure 3(a) shows the distribution of per-connection bandwidth utilization across peers that are n hops away from source and their child peers (labeled as level n) for different values of n in the tree-based approach. This figure demonstrates that connections that are further away from the source have a lower average utilization due to the higher probability of experiencing low bandwidth among their upstream connections. The mesh-based approach exhibits a high bandwidth utilization ($>$%95) across all connections in a similar setting since it can cope with content bottlenecks (the result is not shown here). Figure 3(b) depicts the average bandwidth utilization as a function of peer degree for both approaches. This figure reveals that by increasing peer degree the bandwidth utilization rapidly improves for both approaches. In the tree-based approach, increasing peer degree reduces the depth of all trees which decreases the number of upstream connections and thus the probability of a content bottleneck. In the mesh-based approach, the improved utilization for higher degree is due to the larger number of parents peers which provides more flexibility for packet scheduling and reduces the probability of content bottleneck. More importantly, the mesh-based approach exhibits a higher utilization across all degrees primarily due to its flexibility to dynamically map its required content to parents and effectively use their available bandwidth.

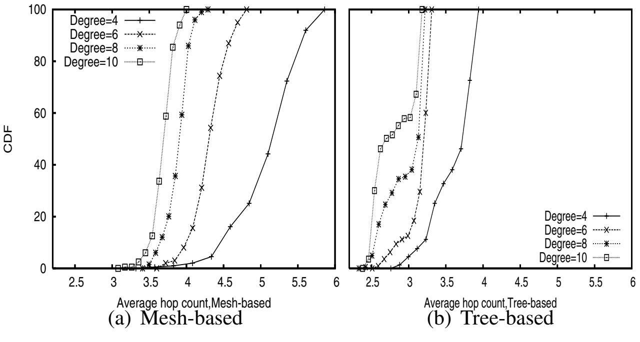


(a) Mesh-based (b) Tree-based

Fig. 4. Avg. aggregate BW utilization based on peers' shortest distance

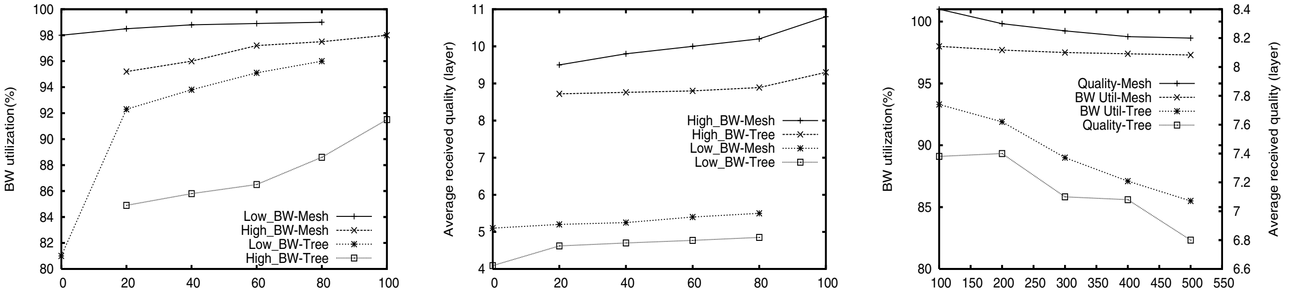
Figure 4(a) and 4(b) show the average bandwidth utilization among peers at different distance from source. These figures reveal that the aggregate bandwidth utilization does not depend on peers location in the overlay for both approaches. The aggregate average bandwidth utilization depends on the average distance of each peer across the delivery tree of different packets. In the mesh-based approach, because of the random connectivity among peers, average distance of all peers is very similar. In the tree-based approach, for large peer degrees, the average distance of all peers is similar. To explain this, we note that the average distance of each peer is primarily determined by the depth of individual trees since each peer is placed as a leaf in all but one tree. On the other hand, the observed disparity for small peer degrees (degree < 4) in the tree-based approach is due to the pronounced effect of peer distance on the tree where it serves as an internal node.

Average hop count (*i.e.*, the number of peers that a packet visits before reaching each peer) among delivered packets to each peer represents its average distance across all delivery trees. Figure 5(a) and 5(b) depict the distribution of average hop count among delivered packets to each peer. These figures present two interesting points: *(i)* the average path length is generally longer in the mesh-based approach. This is mainly due to the flexibility of swarming delivery that allows a peer to receive missing packets through a longer path from its swarming parents, *(ii)* as the peer degree increases, the average path length in both approaches decreases but for different reasons. For the tree-based approach, increasing degree reduces the depth of all trees and results in a lower average hop count for individual peers. In the mesh-based approach, increasing degree reduces the depth of delivery trees by providing more shortcuts in the mesh. This in turn enables each peer to receive



(a) Mesh-based (b) Tree-based

Fig. 5. Distribution of average hop count (or path length) among peers



(a) Aggregate BW utilization (b) Average Delivered Quality (c) Access Link Utilization & Average Quality

Fig. 6. Effect of Bandwidth heterogeneity & Group Size

more packets through a shorter path which leads to a shorter and more homogeneous average path length among peers.

C. Effect of Bandwidth Heterogeneity

To explore the effect of bandwidth heterogeneity among participating peers, we consider two groups of peers with symmetric bandwidth of 480 Kbps and 960 Kbps, and peer degree of 5 and 10, respectively. Since the bandwidth to degree ratio is the same for both groups, the average per-connection bandwidth should be roughly the same across all connections. Figure 6(a) depicts the average bandwidth utilization for high and low bandwidth peers in both approaches as a function of the percentage of high bandwidth peers in the group. Figure 6(b) presents the average delivered quality in the same scenarios. These figures indicate that both groups of peers consistently achieve a higher utilization and receive a significantly better quality in the mesh-based approach. The bandwidth utilization and thus the delivered quality to individual peers in the mesh-based approach depends on the aggregate quality of available content among their parents [4]. Therefore, as the percentage of high bandwidth peers increases, the performance of the mesh-based approach gradually improves. In the context of tree-based approach, the main determining factor for both utilization and quality is the average depth across different trees. Increasing the percentage of high bandwidth peers rapidly drops depth of all trees which in turn improves both utilization and the delivered quality.

D. Effect of Group Size

To investigate the effect of group size, we examine a group of peers with homogeneous bandwidth and peer degree 8. Figure 6(c) presents the average bandwidth utilization and average delivered quality among all peers for both approaches as a function of group size. Figure 6(c) reveals that as the group size increases, both the utilization and the delivered quality in the tree-based approach gradually drops whereas the mesh-based approach consistently exhibits high performance. To explain the behavior of the tree-based approach, we note that for a given peer degree, the depth of individual trees increases with the group size. This in turn decreases the per-connection bandwidth utilization due to the higher chance for content bottleneck among different connections. In contrast, the flexibility of swarming content delivery enables mesh-based approach to effectively scale with group size.

IV. ABILITY TO COPE WITH CHURN

The dynamics of peer participation (or churn) could disrupt content delivery and adversely affect the delivered quality to participating peers. Such a disruption occurs when a peer loses its direct parent, or any upstream node along the path from source. In this section, we examine the effect of churn on the tree- and mesh-based approaches.

To cope with churn, an affected peer should rejoin the proper tree in the tree-based approach, or connect to a new parent in the mesh-based approach. While the effect of churn is often transient, its impact on the delivered quality depends on many factors including details of the recovery mechanism, amount of buffering at each peer, and the characteristics of churn. Therefore, instead of quantifying the delivered quality in dynamic scenarios, we examine the performance of these approaches at the following two levels: *(i)* the performance of content delivery on distorted overlays, and *(ii)* the cohesion of the overlay structure under persistent churn. This methodology not only allows us to separately examine the effect of churn on content delivery and overlay construction mechanisms but also simplifies the comparison between two candidate approaches.

A. Content Delivery on Distorted Overlays

To model a distorted overlay, we use a properly connected overlay in both tree- and mesh-based approaches and then assume that $x\%$ of randomly selected peers simultaneously depart without repairing the overlay. The resulting distorted overlay represents the worst case scenario for the overlay as it evolves due to churn. By changing x, we can control the level of distortion in the overlay. Figure 7 depicts the median utilization of aggregate bandwidth among peers in a distorted overlay (as well as its 5th and 95th percentile as a bar) for both approaches as a function of x. This figure clearly illustrates that bandwidth utilization among peers in the mesh-based approach is significantly higher than the tree-based approach. This is primarily due to the ability of the swarming delivery to cope with unbalanced incoming/outgoing degree among participating peers in a distorted overlay. In contrast, the diverse nature of tree structures implies that the departure of any peer in the tree-based approach reduces the delivered quality to all of its descendant peers on the tree where it serves as an internal node. Therefore, the departure of a larger fraction of peers leads to a proportionally larger drop

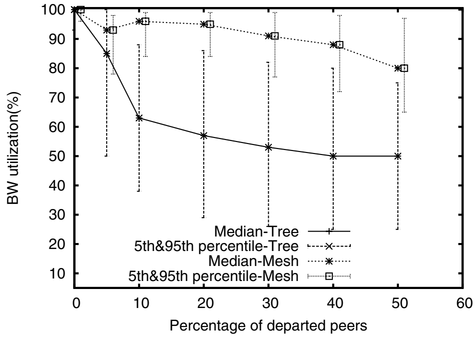


Fig. 7. Median, 5th and 95th percentile of bandwidth utilization among peers after x% of randomly selected peers have departed

in bandwidth utilization and widens its distribution among peers.

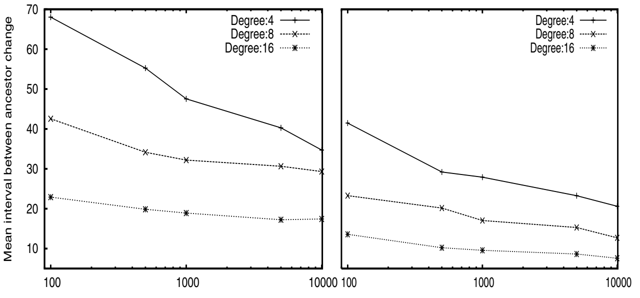
B. Cohesion of the Overlay Under Churn

We now turn our attention to the ability of each approach to maintain a cohesive overlay in the presence of churn. For this analysis, we use our session level P2P simulator, called *psim*. *psim* abstracts out packet level dynamics and allows us to examine significantly larger group sizes. Furthermore, *psim* enables us to accurately model churn and simulates the pairwise latency between peers using the King dataset [7]. *psim* also uses a central bootstrap mechanism with a random selection algorithm for peer discovery and peer selection. To incorporate a realistic model for churn in our simulations, we select peer session times from a log-normal distribution (with μ=4.29 and σ=1.28) and peer inter-arrival times from a Pareto distribution (with a=2.52 and b=1.55) as reported by recent empirical studies [8], [9]. The length of each simulation is 6000 seconds to model a roughly 2-hour event. Presented results are measured at the steady state and averaged over multiple simulations with different random seeds.

Ancestor changing rate: Figures 8(a) and 8(b) depict the mean interval between ancestor changes as a function of peer population in the steady state for three different peer degrees in both mesh- and tree-based approaches, respectively. In the tree-based approach, the ancestor nodes consist of both direct parents as well as any upstream nodes on the path from source. In the mesh-based approach, the ancestor nodes include direct parents as well as any upstream node on the diffusion subtree. These figures demonstrate that the path from source to individual peers is more stable in the mesh-based approach (20%-70%) than in the tree-based approach (5%-40%). The ancestor changing rate increases with the peer degree since the larger number of parents increases the likelihood that one of them leaves the system. Furthermore, for a specific peer degree, the ancestor changing rate increases with peer population. This is mainly due to the fact that the average distance of individual peers increases with peer population in both approaches. Figures 8(a) and 8(b) also show that the slope of change in stability is higher for smaller peer degrees due to the stronger effect of population on overlay depth in these scenarios.

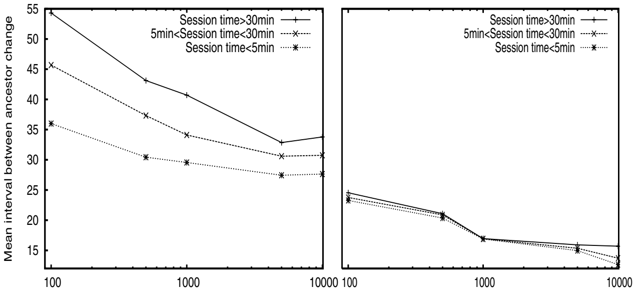
An interesting question is *"whether the observed ancestor changing rate for individual peers is correlated with their session times?"*. To investigate this issue, we divide all peers into three groups based on their session times (st) as follows: *(i)* 30min$<st$, *(ii)* 30min$\leq st \leq$5min, and *(iii)* $st<$5min. Figures 9(a) and 9(b) depict mean interval between ancestor change within each one of these three groups for both approaches with peer degree 8. In the mesh-based approach, peers with higher session times on average experience a higher degree of stability among their ancestor. This is primarily due to the fact that once a connection is established between two long-lived peers, it remains in place for a long period of time. This enables long-lived peers to gradually move to higher levels of the overlay and improves the stability of higher levels. However, in the tree-based approach, there is no visible correlation between the ancestor changing rate and peer session time since all three groups exhibit roughly the same ancestor changing rate across different degrees. This is the direct result of maintaining diverse trees. By forcing each peer to be an internal node in one tree and leaf node in all other trees, the departure of each peer causes instability for all the downstream nodes on the tree where it serves as an internal node.

Frequency of Deadlock Event: As we explained in Section II, a deadlock event occurs in the tree-based approach when a tree becomes saturated and can not accept a newly arriving (or partitioned) leaf peer. Figure 10 shows the average percentage of leaf peers that experienced deadlock as a function of peer population for three different number peer degrees. This figure indicates that the percentage of deadlock events drops as the peer degree decreases or the peer population increases.



(a) Mesh-based (b) Tree-based

Fig. 8. Mean interval between ancestor change



(a) Mesh-based (b) Tree-based

Fig. 9. Mean interval between ancestor change (degree=8)

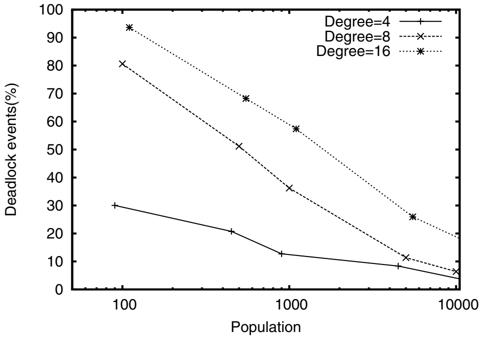


Fig. 10. Percentage of deadlock events in the tree-based approach

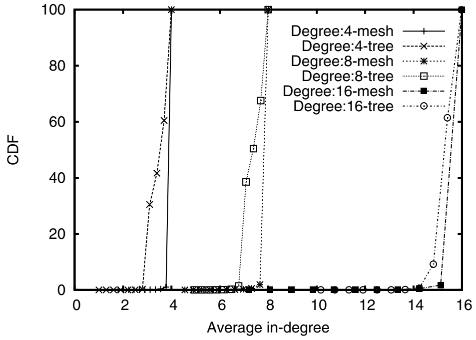


Fig. 11. CDF of weighted avg. incoming degree (Pop=10000)

Increasing peer population increases the number of leaf peers that a tree can accommodate and thus reduces the percentage of deadlock events. Increasing peer degree has an opposite effect since it increases the average number of partitioned leaf peers when an internal node departs. This higher rate of partitioning events among leaf nodes leads to a larger percentage of deadlock events for higher degrees. Figure 10 also shows that in a group of 1000 peers with peer degree 8, on average 40% of join (or rejoin) attempts results in a deadlock. This implies that a peer may remain partitioned from one or more trees for an extended period of time. We further quantify the partitioned intervals in the next subsection.

Average Peer Connectivity: None of the above metrics properly capture the duration of partitioning intervals for those peers that may not be able to quickly connect to the desired number of parents due to deadlock events in the tree-based approach or inefficient peer discovery in the mesh-based approach. To properly quantify the degree of connectivity for individual peers, we keep track of the weighted average incoming degree of individual peers over time. Each spike in Figure 11 presents the distribution of weighted average incoming degree for both approaches across a group of 10,000 peers with a particular target peer degree (4, 8, 16). This figure reveals that the mesh-based approach enables individual peers to reach much closer to their target degree despite churn. As the peer degree increases, the gap between the average degree and the target peer degree grows in both approaches but due to different reasons. In the tree-based approach, the percentage of deadlock events increases with the peer degree (as we showed in Figure 10) which results in extended partitioning intervals for a significant fraction of peers and thus limits their average degree. In the context of mesh-based approach, as the target peer degree increases, it becomes increasingly more difficult for participating peers to maintain their incoming degree at the target level. At any point of time, a significant fraction of peers are in the state of flux, searching for more parents to reach their target degree. The absolute gap between two approaches narrows by increasing peer degree simply because the average degree in the mesh-based approach experiences a larger drop. Interestingly, while the *absolute* gap between the average degree and the target degree widens with peer degree in both approaches, the ratio of the average degree to the target degree which is a better indication for delivered quality, is indeed increasing with the target peer degree. In a nutshell, the delivered quality in both approaches should increase with peer degree.

V. CONCLUSION

In this paper, we compared the performance of tree-based and mesh-based P2P streaming approaches through simulations. We illustrated the similarities and differences between these approaches. We then evaluated the performance of their content delivery mechanisms over a properly connected overlay. We also investigated their ability to cope with churn, in particular the performance of their content delivery over a distorted overlay and the cohesion of their overlay under persistent churn. Our results indicate that that the mesh-based approach consistently exhibits a superior performance over the tree-based approach. We find that the static mapping of content to a particular overlay tree and diverse placement of peers in different overlay trees are two key factors attributing to the inferior performance of the tree-based approach.

REFERENCES

[1] Y. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *ACM SIGMETRICS*, 2000.

[2] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *ICNP*, 2003.

[3] M. Castro, P. Druschel, A.-M. Kermarrec, A. R. A. Nandi, and A. Singh, "SplitStream: High-bandwidth content distribution in a cooperative environment," in *ACM SOSP*, 2003.

[4] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming," in *INFOCOM*, 2007.

[5] N. Magharei and R. Rejaie, "Understanding Mesh-based Peer-to-Peer Streaming," in *NOSSDAV*, 2006.

[6] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," in *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, 2001.

[7] T. M. Gil, F. Kaashoek, J. Li, R. Morris, and J. Stribling, "King dataset," 2004. [Online]. Available: http://pdos.csail.mit.edu/p2psim/kingdata/

[8] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," in *IMW*, 2002.

[9] K. Sripanidkulchai, A. Ganjam, and B. Maggs, "The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points," in *ACM SIGCOMM*, 2004.