ISP-Friendly Live P2P Streaming

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

Volker Hilt, Ivica Rimac, Markus Hofmann
Bell Labs / Alcatel-Lucent
{rimac, volkerh, hofmann}@bell-labs.com

Abstract—Existing Swarm-based Peer-to-Peer Streaming (SPS) applications rely on random connected overlays among peers, which tend to generate a significant amount of costly inter-ISP traffic. To reduce such traffic, localization of overlay connectivity within each ISP has received a great deal of attention by researchers. In this paper, we examine the performance of SPS mechanisms for live video over localized overlays, show that localization significantly degrades their performance, and identify the underlying performance bottlenecks. Leveraging these insights, we propose a novel two-tier overlay-aware block scheduling scheme called OLIVES that differentiates inter and intra-ISP scheduling. OLIVES adopts an implicit coordination between edge peers of each ISP with external parents to ensure proper diffusion of all blocks to each ISP. Moreover, we show that the required buffer size in the localized overlay is significantly larger than the random overlay and propose a shortcutting method to reduce the buffer size. Through analysis and extensive simulations, we demonstrate the ability of OLIVES to deliver high quality stream over localized overlays while minimizing the associated inter-ISP traffic.

I. INTRODUCTION

Peer-to-Peer (P2P) content distribution applications have become increasingly popular over the past decade. Traffic generated by P2P applications makes up a substantial fraction of today’s Internet traffic. In P2P applications, participating peers often form an overlay which is largely agnostic to the underlying physical topology [1], [2]. This in turn increases the cost associated with P2P traffic for individual Internet Service Providers (ISPs) which is a serious concern. This problem has motivated the idea of localizing P2P traffic within individual stub ISPs by localizing the connectivity among their peers [3], [4]. The common assumption in this approach is that localizing the overlay connectivity has minimal impact on the performance of P2P applications. A few studies examined the performance of file swarming mechanisms over localized overlay and reported possible drop in performance in certain scenarios [5] or improvement due to potentially higher available bandwidth within an ISP [3].

Prior studies on this topic have primarily focused on the performance of file swarming mechanisms (i.e., BitTorrent) over localized overlay. However, to our knowledge, the performance of live P2P streaming applications (e.g., [6], [7]) over localized overlay have not been studied. Compared to swarming content delivery, P2P streaming applications (especially for live streams) have more restricted timing requirements for delivery and more limited content availability due to the live nature of content. Given the growing popularity of P2P streaming applications in recent years and the volume of third associated traffic, incorporating the notion of “ISP friendliness” in P2P streaming application becomes increasingly important.

This paper investigates the design and evaluation of ISP-friendly P2P streaming mechanism for live video over the Internet, and makes two important contributions. First, we present the maximum level of localization in overlay connectivity for P2P streaming applications and the feasibility of achieving this goal in different scenarios. We then examine the performance of commonly used P2P streaming applications over an overlay with different levels of localization, and identify fundamental underlying factors that may limit the delivered quality to peers in such settings. Furthermore, we illustrate the impact of redundancy in the external connectivity of individual ISPs on the delivered quality. Second, the main contribution of this paper is a new Overlay-aware Live P2P Streaming mechanism, called OLIVES. In OLIVES, participating peers maintain a fully localized overlay within individual ISPs to effectively limit their external traffic. OLIVES incorporates a two-tier overlay-aware block scheduling scheme to maximize delivered quality to all peers in a fully localized overlay. Content delivery in OLIVES is managed at two tiers: (i) the top tier between ISPs via inter-ISP scheduling, and (ii) the bottom tier among peers within each ISP via intra-ISP scheduling. Furthermore, OLIVES incorporates other mechanisms such as “implicit coordination” for fine-grain management of bandwidth among incoming connections of individual ISPs. We discuss the impact of two tier scheduling on buffer requirement at each peer, and present the idea of “shortcutting of ISP” to limit these side effects. Using detailed simulations, we evaluate the performance of the two-tier scheduling scheme over localized overlay with different graph properties. Furthermore, we examine the performance of OLIVES in more realistic scenarios with peer and bandwidth dynamics. Our results demonstrate the ability of the OLIVES protocol to deliver good performance over a wide range of scenarios and provide valuable insights in the behavior of P2P streaming applications over localized overlay.

We are only aware of one other study on P2P streaming over localized overlay by Picconi et al. [8]. They propose an ISP-friendly swarm-based live streaming mechanism which builds a clustered primary overlay augmented by a large number of dynamically-unchoked, secondary inter-cluster links. In contrast to OLIVES, their approach does not explicitly control external traffic for each ISP. Therefore the level of traffic localization in their approach is significantly smaller than OLIVES.

The rest of this paper is organized as follows: Section II presents a background on Swarm-based P2P Streaming (SPS) mechanisms for live video that provides the required context for the rest of the paper. We derive maximum overlay localization for P2P streaming and its feasibility in section III.
In Section IV, we investigate the performance of commonly used P2P streaming techniques over localized overlay and identify their fundamental performance bottleneck. We present OLIVES protocol in Section V. In Sections VI and VII, we evaluate OLIVES protocol using session level and packet level simulations, respectively.

II. BACKGROUND

Swarm-based Live P2P Streaming: In Swarm-based P2P Streaming (SPS) systems for live video, participating peers maintain a randomly connected overlay (or mesh) over which they incorporate swarming (i.e., pull) content delivery. Without loss of generality, we assume that (i) the overlay is directed. This means there is a parent-child relationship between connected peers where content is always delivered from parent to a child. A bidirectional overlay can be viewed as a special case for directed overlay. (ii) incoming and outgoing degree of any peer \( i \) are proportional to the peer’s incoming and outgoing bandwidth \( (bw_{in}(i)) \) and \( (bw_{out}(i)) \) bandwidth, respectively. Given configurable bandwidth per connection BW P C, peer \( i \) tries to maintain \([\min(STRBW,bw_{in}(i))]/BW P C\) parents and accepts up to \([BW P C/bw_{out}(i)]\) child peers where STRBW denotes the stream bandwidth. This implies that the average bandwidth of all overlay connections is roughly equal to BW PC, which is not a requirement but eases the discussion throughout the paper.

In a live video streaming application, each peer simultaneously pulls content from all of its parents while providing content to all of its children. Parent peers periodically (i.e., once per \( \Delta \) seconds) report the availability of new blocks to all of their children. Knowing the available blocks among its parents, each peer periodically (i.e., once per \( \Delta \) seconds) invokes a block scheduling scheme to determine which blocks to pull from each parent in order to maximize the utilization of its aggregate incoming bandwidth and the delivered quality.

Representing Swarming with Delivery Trees: In SPS, the collection of overlay connections used for the delivery of a block from source to all peers forms a source-rooted spanning tree. This notion of a delivery tree can be generally associated with a subset of blocks including a substream of a video. Without loss of generality, suppose a content source has \( D \) children and its bandwidth is sufficient to only send a single copy of individual blocks to the overlay. Then, all the delivered blocks to each child of source can be viewed as a substream. In the absence of any peer or bandwidth dynamics, if all connections have the same bandwidth, all blocks of a substream traverse the same delivery tree. In this simplified setting, the scheduling scheme at each peer in essence determines which substream to pull from each parent. Characteristics of delivery trees (for individual substreams) in this simplified setting demonstrate the basic performance of the scheduling scheme on a given overlay as follows: (i) the number of delivery trees that contain node \( n \) represents the delivered quality to this node, and (ii) the maximum depth \( d_{max} \) among all delivery trees indicates the required buffering at each peer as \( bw_{in}(sec)=d_{max} \cdot \Delta \). Therefore, the goal of the scheduling scheme is to form \( D \) edge-disjoint spanning trees with minimum depth.

Properties of the delivery trees clearly depend on the interactions between the scheduling scheme and the overlay topology. Several studies [7], [9], [10] have shown that the following scheduling algorithm results in the optimal performance for individual peers: each peer requests the most recent blocks (with largest timestamp) from the corresponding parent. Requested blocks from other parents can be selected using a rarest-first or more-recent-first strategy. To simulate this scheduling scheme, we assume that each block carries a hop count for the number of visited peers. Therefore, the hop count \( \text{OHC}_i(p) \) for (blocks of) substream \( s \) at peer \( p \) represents the depth of \( p \) on the corresponding delivery tree, which also indicates the relative recency of the received block for the substream. Given this information, each peer identifies the parent that has the minimum depth across all delivery trees, and request the corresponding substream from that parent. This procedure is repeated among the remaining parents until the requested substreams from all parents are determined.

Prior studies have shown that the above Shortest-Path (SP) scheduling scheme\(^2\) has two properties: (i) it outperforms other schemes in resource-constrained settings, and (ii) the maximum depth of delivery trees over a randomly connected overlay is limited to:

\[
d_{max} \leq \log N^D + 1 + \frac{1}{1 - e^{-D}} < \log N^D + 3
\]

where \( N \) and \( D \) denote the total number of peers and average peer degree, respectively. This equation indicates that the maximum depth of delivery trees over a randomly connected overlay is equal to the largest shortest-distance of peers from source plus three hops. We use SP scheduling to represent well-behaved swarms for live streaming throughout this paper.

III. UNDERSTANDING OVERLAY LOCALIZATION

The basic idea in external P2P traffic reduction for individual stub ISPs is to localize the connectivity of the overlay within each ISP. More specifically, enabling each peer to connect to other peers within the same ISP reduces the number of external connections, thus, the volume of costly inter-domain traffic.

Maximizing Overlay Localization for Live P2P Streaming: In the context of P2P live streaming applications, the aggregate incoming bandwidth to each ISP should be at least equal to the stream bandwidth to ensure that new packets continuously “stream” to peers in that ISP. Assuming all overlay connections have roughly the same bandwidth (BW PC), maximum overlay localization is achieved when the number of incoming external connections for each ISP is set to its minimum value \( D_{min} = \lceil \frac{\text{STRBW}}{BW PC} \rceil \), where \( D_{in}(i) \) denotes the actual incoming degree of ISP \( i \). Note that this requirement does not depend on the population of peers in an ISP. Given the minimum number of incoming external connections, we can define the

\(^1\)The substream with minimum depth among parents by definition provides the rarest blocks. Thus, our algorithm can be viewed as rarest first with priority for more recent blocks.

\(^2\)We note that different names have been used for this basic scheduling scheme in other studies. Furthermore, the same idea is used for tree construction in tree-based P2P streaming techniques [6], [11].
Feasibility of Overlay Localization: The problem of determining whether it is feasible to achieve maximum localization in connectivity for a given group of M peers in an ISP can be formulated as follows: a group of M nodes with certain in and out degree pairs \( D_{\text{in}}, D_{\text{out}} \) should be connected together such that (i) there is at most one single edge (in each direction) between any pair of nodes, and (ii) the number of un-established incoming and outgoing edges (i.e., minimum external connections for ISP) are \( I(D_{\text{in}}) \) and \( I(D_{\text{out}}) \), respectively. If the number of unestablished edges are zero, the problem is essentially equal to determining whether an integer-pair sequence \( P = \{P_{\text{in}}, P_{\text{out}}\}_{i=1}^{M} \) is digraphic or there is directed graph with degree sequence \( P \). Fulkerson theory [12] can be applied to solve this problem.

**Theorem 1**—Let \( D_{\text{deg}} = \{P_{\text{in}}, P_{\text{out}}\}_{i=1}^{M} \) be a negatively ordered integer-pair sequence. Then \( D_{\text{deg}} \) is digraphic if and only if

\[
\sum_{i=1}^{M} (P_{\text{in}}(i)) = \sum_{i=1}^{M} (P_{\text{out}}(i)) \quad (2)
\]

and for \( s = 1, 2, ..., M \),

\[
\sum_{i=1}^{s} \min(P_{\text{in}}(i), s-1) + \sum_{i=s+1}^{M} \min(P_{\text{in}}(i), s) \geq \sum_{i=1}^{s} (P_{\text{out}}(i)) \quad (3)
\]

To allow \( I(D_{\text{in}}) \) and \( I(D_{\text{out}}) \) unestablished edges assuming \( Min = \min\{I(D_{\text{in}}), I(D_{\text{out}})\} \) and \( Max = \max\{I(D_{\text{in}}), I(D_{\text{out}})\} \) the above theorem can be modified by assuming \( Max \) additional integer-pair sequences in the forms of \[E_{\text{in}}(i), E_{\text{out}}(i)\]_{i=1}^{Max} = \{(1, 1), (0, 1), (1, 0), \ldots\} for \( i = 1, ..., Min \)

\[
\begin{cases}
(1, 1) & \text{if } I(D_{\text{in}}) > I(D_{\text{out}}) \\
(0, 1) & \text{if } I(D_{\text{in}}) \leq I(D_{\text{out}}) \quad (4)
\end{cases}
\]

**Corollary 1**—Let \( E_{\text{deg}} \) be a negatively ordered integer pair sequence of integer pair sequence of \( E_{\text{deg}} = P_{\text{in}}(K) \), while \( P = \{P_{\text{in}}(i), P_{\text{out}}(i)\}_{i=1}^{M} \) and \( K = \{E_{\text{in}}(i), E_{\text{out}}(i)\}_{i=1}^{Max} \). If \( E_{\text{deg}} \) is digraphic then a digraph over degree sequence \( P \) can be built considering \( I(D_{\text{in}}) \) and \( I(D_{\text{out}}) \) unestablished incoming and outgoing edges and given \( L = \sum_{i=1}^{M} (P_{\text{out}}(i)), I(D_{\text{out}}) \leq L \).

**Proof:** (i) If all connections in set of \( K \) nodes are between \( P \) and \( K \), then there should be \( I(D_{\text{out}}) \) connections from \( P \) to \( K \) and \( I(D_{\text{in}}) \) connections from \( K \) to \( P \). Removing the \( Max \) number of peers in \( K \), makes \( P \) a digraph with \( I(D_{\text{in}}) \) and \( I(D_{\text{out}}) \) unestablished incoming and outgoing edges. (ii) If there is at least one connection from node \( k_1 \) to \( k_2 \) in \( K \) then there is at (a) least one connection form \( p_1 \) to \( p_2 \) in \( P \) that can be cut to establish connections from \( p_1 \) to \( k_2 \) and \( k_1 \) to \( p_2 \) while (b) there is not any existing connections between such nodes previously.

To prove (a), if there is no internal connection in \( P \), then all connections from set of nodes in \( P \) are to the set of \( K \). Assuming \( M \) is the number of those connections:

- **case** \( M = I(D_{\text{in}}) \): then there should be least \( M + 1 \) nodes in \( K \) in form of \((1, *)\) (to have one internal connection in \( K \)). The number of nodes in \( K \) in the form of \((1, *)\) at most \( I(D_{\text{out}}) \) this means that all nodes in \( K \) in the form of \((1, *)\) have an incoming connection from set of nodes in \( P \) and there cannot be any internal connection in \( K \), thus, \( P \) should have at least one internal connection.

- **case** \( M < I(D_{\text{out}}) \): then there should be least \( M + 1 \) nodes in \( K \) in form of \((1, *)\) (to have one internal connection in \( K \)). The number of nodes in \( K \) in the form of \((1, *)\) at most \( I(D_{\text{out}}) \) thus, \( M \) cannot be bigger than \( I(D_{\text{out}}) \) and case dismissed.

- **case** \( M = I(D_{\text{out}}) \): then as \( M = L \), \( L \cup I(D_{\text{out}}) \), which is against the problem assumption. Thus this case is also not valid.

(b) is also valid, as there cannot be any existing connection between \( p_1 \) to \( k_2 \) or \( k_1 \) to \( p_2 \) as incoming and outgoing of nodes in \( K \) are at most 1, so they cannot be connected to any other node previously and also connect to each other.

In a setting where all peers and the ISP have the same in and out degree (i.e., \( P_{\text{in}}(i) = P_{\text{out}}(i) = ID_{\text{in}} = ID_{\text{out}} = D) \), the basic requirement for the feasibility of localization is that peer population needs to be larger or equal to \( D \). In the remainder of this paper, we primarily focus on scenarios where maximum overlay localization is feasible by assuming an adequately large population of peers in individual ISPs and \( R(I(j)) = 1 \) for all \( j \). Participating peers, however, may have heterogeneous and asymmetric bandwidth connectivity.

**IV. SWARMING LIVE VIDEO OVER LOCALIZED OVERLAY**

One of the key question in swarming live video over localized overlays is whether and how localization of the overlay connections affects the performance of content delivery for live streams? We simulate SP scheduling over an overlay with 5000 peers that are evenly distributed over 40 ISPs, whereby the incoming and outgoing degree of all peers are 12. We focus on a resource-constrained setting where the source sends...
a single copy of each block of video. While this simulation scenario captures only the basic behavior, it is suitable to reveal the major performance bottlenecks.

Figure 2(a) depicts the median value (along with the 5th and 95th percentile) of the delivered quality to participating peers as a function of the level of redundancy in external connectivity of individual ISPs. This figure illustrates the following important points: First, SP scheduling over a fully-localized overlay reduces the delivered quality by 35%. Second, as redundancy in external connectivity of ISPs increases (i.e., the overlay localization decreases), the overall performance gradually improves. To deliver a good quality to a large fraction of peers, at least 20% redundancy (3 times stream bandwidth) is required. Interestingly, repeating these simulations with various peer and ISP degrees leads to similar results as shown in Figures 3.

A. Fundamental Performance Bottlenecks

Our basic simulation study reveals two fundamental performance bottlenecks of SPS mechanisms over localized overlays. For the discussion, we introduce the following differentiation amongst peers in an ISP: edge peers have at least one external incoming connection, and internal peers do not establish any external incoming connections.

1) Misallocation of External Connections: SP scheduling may result in improper mapping of external connections to substreams which in turn leads to the first two bottlenecks: (i) It may limit the delivery of substreams to only a subset of ISPs. Consider the overlay in Figure 1(a) for the delivery of two substreams which in turn leads to the first two bottlenecks: (i) If tree 1 is terminated at ISP 1 so that this substream cannot reach peers in other ISPs. (ii) Since edge peers within each ISP independently determine the requested substream from their external parents, it is likely that all the substreams are not collectively pulled by all edge peers of an ISP.

In Figure 2(a), the line labeled by "ISPs Delivered Quality" shows the median delivered quality to individual ISPs in our simulations. The low delivered quality to individual ISPs is due to a combination of the above two problems.

2) Misallocation of Internal Connections: Even if all substreams are delivered to an ISP, SP scheduling may not result in a proper propagation of a substream from an edge peer to all internal peers within an ISP. To demonstrate this problem, consider an ISP in Figure 1(b) whose edge peers A and B pull substream 1 and 2, respectively. Furthermore, peer A pulls substream 2 from an internal peer (peer B in this case) such that \( \text{OHC}_2(A) < \text{OHC}_1(A) \). As a result, internal peers C and D pull substream 2 from peer A due to a smaller hop count, which in turn makes substream 1 unavailable for other peers in this ISP. Note that the above performance bottlenecks are not specific to SP scheduling and may be even further aggravated in other scheduling schemes.

B. The Role of Redundant External Connections

A simple strategy to reduce the effect of misallocation of external connections is to increase the redundancy in external connectivity. However, this brings up the question "What is the minimum required redundancy in the external connectivity of an ISP to ensure delivery of all substreams with a high probability?" Suppose \( K, P, \) and \( D \) denote the number of external connections of an ISP, total population of peers in the system, and the number of substreams, respectively. Given an ISP, the goal is to determine the minimum \( K \) that results in connecting to \( D \) distinct delivery trees with shortest distance from source. This problem is equal to determining the minimum number of samples \( K \) from a basket of \( P \) balls that are divided into \( D \) distinct colors such that at least one ball from each color is selected. The probability of not selecting a subtree of type \( t \) within \( K \) samples is:

\[
P(n(t)) = P(n(p_0) \cap n(p_1) \cap n(p_2) \ldots \cap n(p_{N_t}))
\]

\[
P(n(p_0)) = 1 - \frac{K}{P}
\]

\[
P(n(p_0) \cap n(p_1)) = (1 - \frac{K}{P}) \cdot (1 - \frac{K}{P - 1})
\]

Thus we have:

\[
P(n(t)) = (1 - \frac{K}{P}) \cdot (1 - \frac{K}{P - 1}) \cdot (1 - \frac{K}{P - 2}) \cdot \ldots \cdot (1 - \frac{K}{P - (N_t - 1)})
\]

\[
= \prod_{j=0}^{N_t-1} (1 - \frac{K}{P - j})
\]

The expected value of \( K \) for having \( D \) distinct samples can be computed by first introducing an auxiliary function as [13]:

\[
\delta(t) = \begin{cases} 
0 & \text{If tree } t \text{ is not selected} \\
1 & \text{If tree } t \text{ is selected}
\end{cases}
\]

The expected number of distinct trees is:

\[
E[\text{Distinct}] = E \left[ \sum_{t=0}^{D-1} \delta(t) \right] = \sum_{t=0}^{D-1} E[\delta(t)] = \sum_{t=0}^{D-1} (1 \cdot P(\delta(t) = 1) + 0 \cdot P(\delta(t) = 0)) = \sum_{t=0}^{D-1} \sum_{t=0}^{D-1} (1 - P(n(t))) = \sum_{t=0}^{D-1} \sum_{j=0}^{N_t-1} (1 - \frac{K}{P - j})
\]

Figure 2(c) depicts the normalized expected value of the number of distinct substreams (normalized by the number of substreams) as a function of the redundancy in external connections per ISP. Note that the probabilistic improvement causes by redundancy does not depend on degree of individual
peers (D). This confirms our earlier observation that our simulations results in Figure 2(a) is largely independent of peer degree as shown in Figures 3. Interestingly, the effect of redundancy on delivered quality in Figure 2(c) is very similar to our simulation results in Figure 2(a). We note that the effect of redundancy on delivered quality to individual ISPs does not depend on the degree of each ISP or peer population, and thus exhibits the fundamental effect of redundancy in ISP connectivity.

V. ISP-FRIENDLY SWARM-BASED P2P STREAMING

In this section, we present OLIVES, a swarm-based P2P streaming protocol for live video. The basic idea in OLIVES is to effectively control the external traffic of individual ISPs by limiting the number of external connections (i.e., maintaining a fully localized overlay). The main contribution of the OLIVES protocol is a two-tier overlay-aware scheduling scheme that maximizes the delivered quality to individual peers in a localized overlay. In this paper, we primarily focus on content delivery and only briefly describe the construction of the overlay, since this is a straightforward and well-understood topic.

In OLIVES, participating peers in each ISP maintain a fully localized overlay by maintaining the minimum number of external connections through high-bandwidth and more stable peers. A local tracker within each ISP controls the total number of incoming and outgoing external connections and ensures that the aggregate incoming bandwidth is more than the stream bandwidth. Therefore, the tracker monitors all edge peers. Once an edge peer departs, the tracker elects another peer as an edge peer preferring those that are long-lived and have high bandwidth. External parents in other ISPs are discovered through a session level tracker and are randomly selected.

Motivated by the performance bottlenecks that we have identified in Section IV, content delivery in OLIVES is managed at two coherent or tiers as follows: Inter-ISP Scheduling: At this level, OLIVES focuses on the delivery of full-quality streams to individual ISPs. Inter-ISP scheduling is only concerned with external connections. Intra-ISP Scheduling: At this level, OLIVES ensures the delivery of each substream from edge peers to all internal peers. Intra-ISP scheduling is only responsible for managing internal connections. Since the main idea is to adopt SP scheduling at the two levels, in the OLIVES protocol each content block (or substream) carries the following three counters: (i) ISP Hop Count (IHC) keeps track of the number of ISPs that a block (or substream) has crossed. (ii) Peer Hop Count (PHC) keeps track of the number of internal peers that a block has visited within a single ISP. Therefore, this counter is reset by the corresponding edge peer where a block enters an ISP. (iii) Overall Hop Count (OHC) keeps track of the total number of peers (regardless of their ISP) that a block has visited. We use the following notations for the value of these counters for substream i at node p: IHC_i(p), PHC_i(p) and OHC_i(p).

A. Inter-ISP Scheduling

Suppose that we represent each ISP in the overlay as a single peer. We call this an ISP-level or top-tier overlay. The goal of the Inter-ISP scheduling is to ensure that each ISP receives all substreams of the video. This problem is very similar to the problem of content delivery over a randomly connected overlay since connectivity among ISPs is random. Therefore, the SP scheduling can be adopted to achieve a good performance over the ISP-level overlay as follows: each edge peer examines the available substreams at its external parent p and pulls the substream i with the smallest IHC_i(p).
We demonstrate the behavior of the proposed Inter-ISP scheduling in mapping external connections using our earlier example in Figure 1(a), which shows the values of IHIC for both substreams at the outgoing edge peers of ISP1. Thus, inter-ISP scheduling results in both peers C and D pulling substream 1 from peers A and B, respectively.

The above inter-ISP scheduling ensures the availability of each substream around individual ISPs. However, in the absence of any coordination an ISP’s set of edge peers may not pull all substreams due misallocation of external connections, as discussed in subsection IV-A. OLIVES addresses this problem by incorporating a central coordination mechanism among edge peers within an individual ISP, which is implemented at the local tracker. After the departure of an edge peer, the local tracker selects another peer to serve as edge peer by establishing an external connection, and invokes the coordination mechanism. 3. First the tracker obtains the available substream(s) and their corresponding IHIC() values from the external parents of its edge peers. Based on this information, the tracker emulates SP scheduling across all external connections, determines which substream each edge peer should pull, and notifies all edge peers about their designated substreams. The coordination mechanism ensures that external connections are efficiently utilized.

Figures 2(b) shows the median delivered quality to individual ISPs as well as individual peers with the proposed inter-ISP scheduling as a function of redundancy in the connectivity of individual ISP using the same settings as discussed in Section IV. This figure reveals that inter-ISP scheduling increases the minimum delivered quality to individual ISPs over a fully-localized overlay (i.e., redundancy is 0) to 97%. This result demonstrates that inter-ISP scheduling is able to maximize delivered quality to individual ISPs. However, the resulting improvement in the minimum delivered quality to individual peers is only 17% (i.e., minimum delivered quality to peers is limited to 83%). This gap in performance must be related to the content delivery within individual ISPs, which we discuss next.

B. Intra-ISP Scheduling

The goal of intra-ISP scheduling is to deliver each substream from the designated edge peer to all internal peers of an ISP. Therefore, OLIVES applies the idea of SP scheduling for different substreams based on their relative hop count from the corresponding edge peer captured in PHC(), i.e., each edge peer is treated as the source for its designated substream. Towards this end, each internal peer considers the available substreams among its parents along with their PHC(s) and pulls the substream with the minimum value from each parent. Figure 1(c) demonstrates the intra-ISP scheduling scheme by revisiting the example in Figure 1(b). In this case, internal peers C and D use PHC() and pull substream 1 from edge peer A which leads to the desired behavior.

3When an edge peer loses its external parent, it establishes a connection to another external parent and then the coordination mechanism is invoked

C. Dealing with Bandwidth Dynamics

So far, we considered a rather simplified setting where overlay is static and all connections have the same bandwidth. In such a setting, the notion of delivery tree for each substream reveals fundamental performance limits of scheduling schemes over localized overlay. However, in practice both intra- and inter-scheduling schemes should be able to effectively cope with two important issues: (i) the dynamics of peer participation (or churn) , (ii) the short and long-term variation of congestion-controlled bandwidth across different internal and external connections.

The above description of intra- and inter-ISP scheduling schemes only discusses coarse grain mapping of substreams to connections. In practice, some connections may have higher or lower than the bandwidth of one substream. To effectively utilize bandwidth of individual connections, both scheduling schemes must be able to extend the basic notion of substream mapping by requesting content at a finer granularity, i.e., periodically requesting a list of blocks from each parent. Each peer maintains a moving average of bandwidth bw(i) from each parent i. This average is used as a rough estimator for bandwidth from a parent during the next interval.

Block-based Intra-ISP Scheduling: At each scheduling event, an internal peer p maps each substream to a particular parent, as we have discussed earlier. Suppose that substream s is mapped to parent peer ps. Given the expected value of BWPC and block size b, we determine the target number of blocks of substream s that should be pulled from peer ps in one interval Δ as nref = \( \frac{BWPC \cdot Δ}{b} \). The actual number of blocks that can be pulled from this parent based on its average bandwidth bw(ps) is nnew = \( \frac{bw(ps) \cdot Δ}{b} \). Suppose the actual number of new blocks of substream s at peer ps is nref. Given this information, intra-ISP scheduling occurs in the following two phases.

Phase I: This phase implements the basic mapping of substreams to different parents. Since the number of requested blocks of substream s from parent ps is limited to \( \text{MIN}(n_{\text{ave}},n_{\text{new}}) \), two following scenarios could occur for each connection: (i) Resource Deficit: In this case, the parent is unable to provide the sufficient number of blocks. Therefore, peer p only requests the most recent blocks with highest timestamps in an ordered list (based on increasing timestamp). Furthermore, peer p keeps track of the number of blocks that have not been requested from the designated parents. (ii) Resource Surplus: In this case, peer p requests nref most recent blocks of substream s from parent ps. It also keeps track of the number of extra blocks that can be pulled from this parent using any surplus bandwidth.

Phase II: Since the aggregate incoming bandwidth from all parents is equal or larger than stream bandwidth, the aggregate block surplus must be sufficient to pull all the deficit blocks from non-designated parents of peer p. The scheduling maps unrequested blocks for different substreams in phase I among external parents with surplus bandwidth based on their available blocks such that aggregated allocated bandwidth to all substreams is roughly even. Once mapped blocks to each parent are determined, blocks identified in phase I and appended by identified ones in phase II are requested from
corresponding parents ordered by their timestamp. Ordering blocks in each phase based on their timestamps is useful in the inter-ISP scheduling.

**Block-based Inter-ISP Scheduling:** Similar to intra-ISP scheduling, inter-ISP scheduling should also cope with bandwidth variations among different external connections for each ISP. We recall that the coarse-grained mapping of substreams to edge peers are determined by the central coordination mechanism. The basic idea of fine-grained inter-ISP scheduling is very similar to intra-ISP scheduling during phase I. If the external incoming connection has a resource deficit, the edge peer allocates the available bandwidth for pulling the most recent blocks of the designated substream. If the external connection has resource surplus, it allocates $\text{BW}_{PC}$ bandwidth for pulling $n_{ref}$ blocks of designated substream.

In Phase II, the key challenge is to effectively manage the excess bandwidth of some external incoming connections to pull deficit blocks from non-designated substreams. A deterministic solution to this problem would require a tight and fine-grained coordination among edge peers which is expensive and undesirable. In OLIVES, each edge peer leverages the “unavailability of a block from non-designated substream in its neighborhood” as a hint to identify blocks that have not been requested by their corresponding edge peer. The main intuition is that a block (of a non-designated substream) with sufficiently early timestamp that is missing from all internal parents of an edge peer, with a high probability has not been pulled into the ISP by the corresponding edge peer. To implement this idea, in each scheduling event an edge peer examines the available blocks among its internal parents and identifies the largest available timestamp for each non-designated substream $s$, $t_{s_{max}}(s)$. Then, it subtracts one interval $\Delta$ from these maximum timestamps to identify a conservative maximum threshold for timestamp of blocks that must have been propagated to these internal parents by now. Blocks from a non-designated substream $s$ with lower timestamp that are missing from all internal parents are unlikely to have been requested by its designated edge peer for two reasons: (i) all requested (and thus delivered) blocks from parents are ordered based on their timestamps, and (ii) available blocks among internal parents indicate the flow (and availability) of content for a large fraction of peers in a randomly connected overlay within the ISP.

The above strategy enables each edge peer to identify blocks that most likely have not been pulled into the ISP. There still needs to be a coordination among edge peers with excess bandwidth to determine which missing blocks are pulled by each edge peer. OLIVES uses the idea of “implicit coordination”. Towards this end, given the IDs of individual substreams, each edge peer utilizes its excess bandwidth to pull only missing blocks of non-designated substreams (i.e., acts as a backup entry point for those substream) in a prioritized fashion using a circular order among substreams. Further the edge peer selects a random subset of these blocks and appends an ordered list of this blocks to its request. This implicit coordination strategy can effectively manage the utilization of excess bandwidth among edge peers (i.e., pulls a small fraction of duplicate blocks into the ISP) for the following reasons: First, a single edge peer is often able to pull a random subset of missing blocks from another substream, and randomness significantly reduces the probability of duplication. Second, edge peers detect missing blocks of a non-designated substream at different times based on their distance from the corresponding edge peer. This adds a random delay to their reaction and reduces the probability of duplication. Third, bandwidth deficit and surplus on external connections are often short-lived and thus move among external connections.

**D. Buffer Requirement**

The maximum depth of a delivery tree in OLIVES two-tier scheduling is the product of two components: (i) the maximum depth of the ISP-level delivery tree on the top-tier overlay, and (ii) the maximum depth of the delivery tree from each edge peer to all internal peers within individual ISPs. Since the overlay connectivity within individual ISPs as well as top-tier inter-ISP are both random, the maximum depth of the delivery tree (in terms of hops) can be derived by extending Eq. 1 in Section II as follows:

$$\left(\log_{D}^{\text{ISP}(D-1)} + 3\right) \ast \left(\log_{D}^{\text{ISP}(D-1)} + 3\right)$$

Figure 4(a) plots the maximum depth of delivery trees in a localized overlay and comparable random overlay using Eq. 8 as a function of the number of peers per ISP. As the figure shows the depth of delivery tree in OLIVES is always larger than the delivery tree over a comparable random overlay. OLIVES manages to deliver each substream to all ISPs despite limited inter-ISP connectivity in localized overlay at the cost of forming taller delivery trees. Taller delivery trees means more buffering at each peer.

**Shortcuting of ISPs:** The larger buffer requirement in OLIVES could be considered a drawback. One practical implication of this requirement is that addition of an ISP with a large number of peers to the overlay can significantly increase the buffer requirements for peers in other ISPs with small population. OLIVES adopts the idea of “shortcutting” to reduce this buffer requirement. The basic idea is to minimize the distance between the incoming and outgoing external connections for each ISP by controlling the internal connectivity of each ISP. In OLIVES, each outgoing edge peer selects all incoming edge peers as parent as shown in Figure 5(a). This mesh-like internal connectivity among all edge peers enables each
outgoing edge peer to provide any substream to other ISPs even when mapping of incoming external connections changes by the coordination mechanism. Shortcutting has two opposite effects on the depth of delivery trees: First, it significantly reduces the depth ($OHC()$) of incoming edge peers of all ISPs on any delivery subtree. Second, it may slightly increase the distance between incoming edge peers and other internal peers of the corresponding ISPs on the delivery tree. This increase is at most one hop due to the logarithmic relation between depth within an ISP and its population. In summary, the overall effect of shortcutting leads to a significant decrease in the depth of delivery trees and thus buffer requirement. The maximum depth of the delivery trees with shortcuts can be derived as follows:

$$((\log_{D}(\frac{N}{D^{2}} - 1)) + 2) + ((\log_{D}(\frac{N}{D^{2}} - D)) + 3) + 1$$

Figure 4(a) and 4(b) shows the maximum depth of delivery trees in a localized overlay with shortcutting. This figure clearly demonstrates the ability of shortcutting to reduce depth of the delivery trees across the parameter space. Clearly, the cost of shortcutting is the overhead of maintaining the connectivity among edge peers in the presence of churn which can be performed by the local tracker.

VI. EVALUATION: EFFECT OF OVERLAY CONNECTIVITY

In this section, we examine how various overlay properties affect the overall performance of two-tier scheduling in OLIVES using substream abstraction for content delivery. Towards this end, we only focus on fully localized overlay where incoming and outgoing peer and ISP degree are equal to the number of substreams. The effect of bandwidth dynamics are examined in the next section.

Effect of Peer Degree, ISP Degree & Peer Population: We start by examining the effect of the following three parameters that primarily determine the overall connectivity of an overlay: peer degree, number of ISPs and the number of peers. Figure 6(a) depicts the 5th percentile of delivered quality to individual peers and ISPs as a function of peer population per ISP. The incoming and outgoing degree of all peers and ISPs is 4. Each line shows the results for a certain number of ISPs, namely 40 and 100 ISPs. Showing the 5th percentile of delivered quality indicates that 95% of peers or ISPs in each scenario received a higher quality than the shown value. This figure shows that increasing the number of peer per ISP as well as ISPs in the overlay initially improves the performance. To explain this, we note that as the population of nodes in a graph increases, the graph becomes less "clustered". This effect is more pronounced when node degree is small. The lower level of clustering provides more flexibility for delivery trees to reach all nodes and thus results in higher delivered quality. Figure 6(a) demonstrates this phenomenon both in the ISP-level overlay and within each ISP.

Figure 6(b) depicts the 5th percentile of delivered quality to peers and ISPs as a function of peer degree when the number of peers per ISP is 50. Each line shows the result for a different number of ISPs in the overlay. This figure clearly shows that increasing peer degree improves delivered quality. Increasing peer degree improves the overlay connectivity at both levels which facilitates the formation of intra- and inter-ISP delivery trees by the scheduling. In summary, the two-tier scheduling in OLIVES exhibits a good performance in most combinations of peer degree, peer per ISP and ISP per overlay. The performance is moderately dropped only in corner scenarios where all three parameters are small.

Heterogeneous Peer Degree: We now turn our attention to an overlay with heterogeneous (but symmetric) peer degree. In particular, we consider an overlay with 20,000 peers that are evenly grouped into 40 ISPs. 15% of peers have in and out degree of 12 (high bandwidth) and other 85% have in and out degree of 6 (low bandwidth). We assume the number of delivered substreams and thus delivered quality to each peer is proportional to its incoming bandwidth (i.e., degree), e.g., by using MDC encoded video. We focus on delivered quality to high bandwidth peers as they should receive all substreams whereas low bandwidth ones should only receive half of the substreams. Figure 6(c) shows the distribution of delivered quality to individual peers and ISPs for three different overlay construction strategies: (i) edge peers are randomly selected (labeled as “Random”), (ii) only high degree peers are selected as incoming/outgoing edges (labeled as “HighBWEdge”), (iii) only high bandwidth peers are selected as incoming/outgoing edges and shortcutting is used (labeled as “Shortcuts”). Figure 6(c) indicates that allowing low bandwidth peers as edge peers reduces the delivered quality to some ISPs. When a high bandwidth peer has one or more low bandwidth parents, it becomes more difficult for the scheduling scheme to map the required substreams among the parents because every substream is not available at all parents [7]. In a random overlay, the problem with low-bandwidth parents occurs in both inter-ISP and intra-ISP schedulings which leads to lower a delivered quality to ISPs. Placing only the high bandwidth peers as edge eliminates the problem between edge peers and significantly improves delivered quality to ISPs. However, since the relay of each substream through individual ISPs is determined by the intra-ISP scheduling, the problem with low bandwidth parents within each ISP still has some effect on the delivered quality at the ISP level. Shortcutting will eliminate this later problem and maximize the delivered quality to all ISPs. The gap between the delivered quality to ISPs and peers in each scenario, is due to the problem with low bandwidth parents for high bandwidth peers. Clearly, changing the relative percentage of high and low bandwidth peers as well as their degree of bandwidth heterogeneity affect the observed phenomenon as shown in Figure 7(a) with 15% low bandwidth and 85% high bandwidth peers.

Heterogeneous Population per ISP: To examine the effect
of heterogeneous peer population across different ISPs, we consider a scenario with 100K homogeneous peers with degree 12 across 20 ISPs. 90K peers are located in one large ISP and remaining 10K peers are evenly divided among small ISPs. The 5th percentile of delivered quality to both ISPs and peers is roughly 99 and 96%, respectively. Figure 8(a) shows the distribution of average and maximum depth of delivery trees among peers in all small ISPs (labeled "**Depth-Small") and peers in the large ISP (labeled "°Depth-Large") without shortcutting. This figure illustrates that while the average depth of delivery trees is generally lower for smaller ISPs, the presence of a large ISP significantly increases the maximum depth of delivery trees for all peers in the overlay. However, we note that the effect on delivery trees for peers in small ISPs depends on two factors: (i) whether any of their delivery trees crosses through the large ISP, and (ii) the relative distance between the incoming and outgoing edge peers in the large ISP compared to smaller ISP. This subtle effect results in different impact on peers in each group. Figure 8(b) shows the same characteristics of delivery trees in a similar scenario with shortcutting that clearly demonstrates two interesting points: First, the maximum depth of delivery trees (and thus buffer requirement at each peer) for both group of peers has decreased. Second, there is a clear gap between both maximum and average depth of delivery trees in both group of peers. Peers in smaller ISPs typically have a much smaller maximum depth than peers in large ISPs, and thus require proportionally less buffering. The small fraction of peers with the lowest maximum depth in this scenario are edge peers that have the shortest distance from source.

VII. EVALUATION: BANDWIDTH & PEER DYNAMICS

In this section, our main goal is to evaluate the performance of OLIVES block-based intra-ISP and in particular inter-ISP scheduling schemes in the presence of bandwidth and peer dynamics. Towards this end, we use ns2 to conduct packet level simulations. This allows us to construct various scenarios and reliably identify underlying causes in each simulation. We consider an overlay with 101 ISPs where one ISP, called target ISP, has 100 heterogeneous peers but other ISPs are simulated as a single peer. This enables us to examine the effect of packet level dynamics and ISP-level overlay on the content delivery to a single ISP within the size of feasible scenarios in packet level simulations. 85% of peers in the target ISP have low (750Kbps) and the rest have high (1.5Mbps) symmetric access link bandwidth. Peers in these two groups maintain (incoming and outgoing) degree of 10 and 20, respectively. The video stream has 1.5 Mbps bandwidth and is MDC encoded with 10 descriptions of 150 Kbps. Source bandwidth is set to 1.6 Mbps to ensure the delivery of full quality stream (with much redundancy) to its 12 child peers collectively. All connections are congestion controlled. The physical topology is generated by Brite [14] with 15 AS and 10 routers per AS in top-down mode and RED queue management in all routers. We only focus on the delivered quality to high bandwidth peers since low bandwidth peers receive full quality stream in all scenarios. The interval for periodic scheduling ($\Delta$) is 6 seconds. Each simulation is run for 2000 simulating seconds.

Effect of per-Connection Bandwidth Heterogeneity: We increase the range of average congestion controlled bandwidth across different connections by controlling the range of RTT values. Towards this end, we consider three scenarios $SC1$, $SC2$ and $SC3$ by randomly selecting the delay on each access link from the following ranges [5ms, 25ms], [5ms, 100ms], and [5ms, 150ms], respectively. Figure 9(a) shows the normal-
ized delivered quality to the target ISP, and median delivered quality to high bandwidth peers in the target ISP (with a bar for 5th to 95th percentile) with and without the implicit coordination mechanism \(^3\) in all the above three scenarios. This figure reveals that the implicit coordination mechanism can maintain high delivered quality to all peers despite the increasing heterogeneity of average bandwidth among overlay connections. However, in the absence of implicit coordination among edge peers, the delivered quality to the ISP shows a moderate decrease while its gap with the delivered quality to high bandwidth peers is quickly widens with larger bandwidth heterogeneity. To explain this, Figure 9(b) depicts the median (with bar for 5th and 95th percentile) time between the generation of a packet at source and its first arrival at an edge peer in the target ISP (i.e., propagation time) for the same scenarios in Figure 9(a). Figure 9(b) indicates that in the absence of coordination, packets experience a longer propagation time, and this difference further grows with the heterogeneity of per connection bandwidth. The reduction in the propagation time in the presence of implicit coordination, is mainly due to the ability of inter-ISP scheduling to effectively utilize excess bandwidth to pull the missing packets. Furthermore, the percentage of duplicate packets that are pulled into the ISP can be effectively limited below 1.2% but it varies between 9% to 13% without implicit coordination.

**Behavior of Coordination Mechanism:** Due to the implicit coordination mechanism, on average 10% of packets across all substreams are pulled into the ISP by a non-designated edge peer in scenario SC1. This number increases to 16% and 18% in scenarios SC2 and SC3, respectively. We take a closer look at the micro-level dynamics of the implicit coordination mechanism. Figure 9(c) depicts the propagation time of each packet to a designated external parent (as an x value) and its propagation time to an edge peer that first receives the packet (as y axis) for all packets of a substream. Packets that enter the ISP through their designated edge are marked differently. This figure illustrates the relative time for availability of a packet outside and inside the target ISP. Roughly 90% of packets in this substream are pulled into the ISP by the designated edge peer as soon as they become available at the corresponding external peer. Figure 9(c) clearly shows that the gap between available time of these packets outside and inside the ISP is very close. Packets that are pulled into the ISP by non-designated peers can be divide into two groups: First, those packets that quickly become available at the designated parent but they are requested from others after some delay. Despite their early availability, these packets were not requested from the designated parent due to the short-term bandwidth deficit of the corresponding external connection. Second, those packets that became available rather late at the designated parent. In essence, the designated edge peer experienced “content bottleneck” and could not pull these packets from its external parent.

**Effect of Churn:** To evaluate the performance of OLIVES in the presence of peer dynamics, we incorporate churn in scenario SC1 using the churn model reported in empirical studies on deployed P2P streaming systems [15], [16]. Towards this end, we select peer session times from a log-normal distribution ($\mu=4.29$ and $\sigma=1.28$), and peer inter-arrival times from a Pareto distribution ($a=2.52$ and $b=1.55$). In the presence of churn, the aggregate average incoming bandwidth to the target ISP is affected by the behavior of the peer discovery mechanism to identify external peers. To examine OLIVES without relying on a particular discovery mechanism, we increase the external degree of the target ISP by 10% so that the aggregate incoming bandwidth in the presence of churn is roughly the same as stream bandwidth. Figure 9(d) depicts the distribution of delivered quality to low and high bandwidth peers, and the same distributions after incorporating churn into our simulations. This figure indicates that the performance of low bandwidth peers does not affect the churn. The delivered quality to a small fraction of high bandwidth peers slightly changes in the presence of churn. The change in the delivered quality depends on (i) the aggregate time for replacing any departing peer, and (ii) whether a new peer has higher or lower bandwidth. Overall, the resulting change in the delivered quality due to churn is minimal (less than 4%) which shows the ability of the implicit coordination to achieve good performance even in presence of peer dynamics.

**VIII. Conclusion & Future Works**

In this paper, we investigated the design and evaluation of ISP-friendly P2P streaming mechanism for live content over the Internet. We examined the performance of commonly used P2P streaming applications over localized overlays and identified fundamental underlying reasons that adversely affect the performance of such applications. Based the above insights, we designed a new Overlay-aware LIVE P2P Streaming mechanism called OLIVES. OLIVES incorporates a two-tier block scheduling scheme over maximum localized overlays to overcome the constraints imposed by localization. Through
detailed simulations we evaluated the performance of OLIVES over various realistic scenarios. We demonstrated the ability of OLIVES to achieve good performance over a wide range of scenarios. We plan to pursue this work along the following directions. First, we would compare various methods of accommodating localization in the context of live P2P streaming. Second, we plan to run experiments over PlanetLab to examine OLIVES in a realistic environment. Third, we would optimize the overlay maintenance mechanism in presence of churn to minimize the effect of edge peer dynamics.

REFERENCES