ISP-friendly Live P2P Streaming

Nazanin Magharei, Reza Rejaie, Ivica Rimac, Volker Hilt and Markus Hofmann

***Abstract*—Swarm-based Peer-to-Peer Streaming (SPS) mechanisms tend to generate a significant amount of costly inter-ISP traffic. Localization of overlay connectivity reduces inter-ISP traffic, however, it can adversely affect the delivered quality.**

In this paper, we systematically examine the performance of SPS for live video over localized overlays. We identify and discuss the fundamental bottlenecks limiting the stream quality, and present OLIVES, an ISP-friendly P2P streaming mechanism for live video. OLIVES maintains a fully-localized overlay to reduce the volume of inter-ISP traffic and incorporates a two-tier inter-ISP and intra-ISP scheduling scheme to maximize the delivered quality to individual peers. One important design choice is to perform basic scheduling at a substream level and to use implicit coordination among peers. This allows OLIVES to efficiently detect missing blocks and pull them into the ISP in a timely manner with a minimum in coordination overhead. Furthermore, OLIVES incorporates a shortcutting technique that limits the buffer requirements for each participating peer and effectively reduced the playout latency. Through analysis and extensive simulations, we demonstrate the ability of OLIVES to deliver high quality streams over localized overlays in various realistic scenarios.

I. Introduction

During the past decade, Peer-to-Peer (P2P) content delivery applications such as BitTorrent and PPLive have been responsible for a substantial fraction of Internet traffic. Most of the above P2P content delivery applications implement both overlay construction and content delivery mechanisms that are largely agnostic to the ISP-level topology [1], [2]. This in turn results in high volumes of inter-ISP traffic, which incurs substantial cost to the ISPs whose customers participate in P2P applications. ISPs have reacted by throttling and limiting the volume of inter-ISP P2P traffic, which on the other hand is affecting the performance of these applications. This problem has motivated researchers to investigate different techniques to make P2P applications more ”ISP-friendly” by controlling their volume of inter-ISP traffic.

The majority of prior studies on ISP-friendly P2P applications have focused on file swarming mechanisms, namely BitTorrent [3], [4]. A common theme among these approaches is to maintain a localized overlay in order to reduce the number of inter-ISP connections and the associated traffic. Reducing the volume of inter-ISP traffic for P2P streaming applications [5], [6] (especially for live video) is more challenging than file swarming for two reasons: *(i)* Streaming applications have stricter timing requirements; *(ii)* The live nature of the content implies limited content availability and diversity. In general, the dependency between the overlay structure and content delivery mechanism for P2P streaming applications is tighter than for file swarming applications. This raises two fundamental questions when tackling the problem of ISP-friendly SPS.

- How do the connectivity of the overlay structure and the details of content delivery affect the performance of a P2P live streaming application?
- How can a P2P streaming application minimize the volume of inter-ISP traffic while ensuring the delivery of high-quality streams to all participating peers?

To the best of our knowledge, only two prior studies have tackled the problem of limiting inter-ISP traffic for P2P streaming applications [7], [8]. However, neither of these two studies does provide answers to the above questions, which we argue are crucial for the design of ISP-friendly SPS mechanisms.

In this paper, we first examine and answer the above fundamental questions; we discuss the effects of overlay structure and block scheduling schemes on the performance of content delivery and explore the design space of SPS mechanisms to achieve ISP-friendliness. To this end, our main contributions are the following: We first quantify the performance limitations of well-known block-scheduling schemes such as most-recent-first and shortest-path-first in the context of localized overlays. We then identify the underlying causes of the fundamental tradeoff between overlay connectivity and delivered quality, and show that they are rooted in misallocations of the overlay connections.

Our findings lay ground for our final contribution, the design of a simple and elegant ISP-friendly SPS mechanism for live video, named OLIVES. OLIVES reduces the inter-ISP traffic to the minimum level by maintaining a fully-localized overlay in each ISP and by implementing a scheduling scheme at two coherent levels: *(i)* inter-ISP scheduling is implemented at edge peers (with external connections) in each ISP to ensure the delivery of the stream in highest quality to the ISP; *(ii)* inter-ISP scheduling is implemented at all peers within an ISP to ensure the delivery of content from each edge peer to all other peers in the same ISP.

The key challenge in the design of inter-ISP scheduling is the required coordination among external connections of an ISP to pull mutually exclusive blocks. Inter-ISP scheduling in OLIVES leverages the logical grouping of blocks into substream to significantly reduce both the overhead and the frequency of coordination. Inter-ISP scheduling uses a distributed and effective approach to ”implicitly” identify and assign unrequested blocks to edge peers to effectively utilize the bandwidth of external connections.

The delivered quality over a fully-localized overlay can generally be maximized only at the cost of larger buffers at

N. Magharei is with the Collage of Computing, Georgia Institute of Technology, Atlanta, GA, USA e-mail: magharei@gatech.edu.

R. Rejaie is with the Department of Computer Science, University of Oregon, Eugene, OR, USA e-mail: reza@cs.uoregon.edu

I. Rimac, V. Hilt and M. Hofmann are with Bell Labs/Alcatel-Lucent Holmdel, NJ, USA e-mail: rimac,volkerh,hofmann@bell-labs.com.

each peer and thus higher playout latency. We derive the buffer requirement mathematically and show that a single ISP with many peers can significantly increase the buffer requirement for all peers. We mitigate this issue by designing a shortcutting technique that makes the buffer requirement at each peer only a function of the number of ISPs in the overlay (rather than the maximum population of peers per ISP).

We extensively evaluate the performance of OLIVES over localized overlays for topologies using detailed session- and packet-level simulations. Our analysis and simulation results show that OLIVES can deliver high-quality streams over a very broad range of scenarios with a maximum level of traffic localization and indicate that OLIVES significantly outperforms known scheduling schemes.

The rest of this paper is organized as follows: Section II presents a background on SPS mechanisms for live video. In Section III, we classify various approaches on the design of an ISP-friendly P2P streaming mechanism. In Section IV, through analysis and simulation we investigate the performance of typical SPS mechanisms over localized overlays and identify their performance bottlenecks. We present details of the design of OLIVES in Sections V and VI. Sections VII and VIII evaluate OLIVES using session- and packet-level simulations, respectively. In Section IX, we present the related work and Section X concludes the paper.

II. Swarm-based Live P2P Streaming: Background

This section presents a background on SPS for live video and provides the required context for the rest of the paper.

A. Overlay Construction

An SPS system for live video usually maintains a randomly-connected overlay (or mesh) [5], [6], [9]. There is a parent-child relationship between each pair of connected peers in the overlay, and content is delivered form a parent peer to its children. All overlay connections implement a TCP-friendly congestion control mechanism such as RAP [10] or TFRC [11].

Without loss of generality and for the clarity of discussion, we assume that the average bandwidth of all connections in the overlay is equal to $BWPC$[1]; we relax this assumption in Section VI. One way to ensure a roughly similar bandwidth for all connections is to set the incoming and outgoing degree of each peer i proportional to its incoming and outgoing bandwidth ($bw_{in}(i)$ and $bw_{out}(i)$), respectively. Thus, given a configuration parameter $BWPC$, peer i tries to maintain $\lceil\frac{MIN(STRBW, bw_{in}(i))}{BWPC}\rceil$ parents and accepts up to $\lceil\frac{bw_{out}(i)}{BWPC}\rceil$ children where $STRBW$ denotes the stream bandwidth.

B. Content Delivery

Each peer can simultaneously pull content from all of its parents while providing content to all of its children. A parent peer reports the availability of new blocks to its children every τ seconds. Knowing the available blocks among its parents, each peer performs block scheduling periodically with the same time interval of τ seconds. The scheduling scheme determines which blocks a child requests from each parent under the objective of maximizing aggregate incoming bandwidth and delivered quality of the peer. Eventually, it also determines the diversity of available blocks across all peers, which in turn affects the overall performance of content delivery [6].

Several studies [6], [12], [13] have shown that *most-recent-first* scheduling maximizes the diversity of blocks among individual peers, achieves optimal streaming quality and minimizes delay for streaming live video in resource-constrained settings. A simple way to implement such a scheme is to carry a hop count in each block that captures the total number of visited peers. Each peer pulls new blocks with the smallest hop count from its parents, which ensures that individual blocks are pulled over the *shortest path*[2] from the source to individual peers.

C. Delivery Trees

The collection of overlay connections that are used for delivery of a block from source to all peers form a source-rooted spanning tree, which is known as the *delivery tree* for that block. Without loss of generality, suppose a video source has D children and its bandwidth is equal to $STRBW$, which is sufficient to send exactly a single copy of each content block to the overlay ($D = \lceil\frac{STRBW}{BWPC}\rceil$). The hop count for block b at peer p indicates the depth of p on the corresponding delivery tree, which also reflects the relative recency of the received block.

Characteristics of the delivery trees (across all blocks) demonstrate the performance of the scheduling scheme on a given overlay as follows: *(i)* the number of delivery trees that contain peer p represents the *delivered quality* to this peer, and *(ii)* the maximum depth (d_{max}) across all delivery trees determines the minimum required buffering at each peer as $buf(sec)$=$d_{max}\times\tau$. In essence, $buf()$ represents the time difference between the playout of the source and a particular peer. Prior studies [5], [6], [13] have shown that d_{max} for the SP scheduling scheme over a randomly connected overlay can be estimated as follows:

$$d_{max} \le \log_D N + 1 + \frac{1}{1-e^{-D}} < \log_D N + 3 \qquad (1)$$

N and D denote the total number of peers and average peer (and source) outgoing degree, respectively. Throughout this paper, we use SP scheduling to represent a well-behaved and well-understood SPS mechanism for live video.

D. Evaluation in Resource-Constrained Settings

Several studies have shown that even a poorly designed block scheduling scheme may exhibit good performance in resource-rich settings [14], in which the capacity of the source

[1] Note that in the case of heterogeneous per connection bandwidth, a connection with bandwidth larger than $BWPC$ can be viewed as a collection of connections with bandwidth of $BWPC$ between the corresponding peers.

[2] Most-recent-first block scheduling is known under several different names in literature [5], [12], [13]. We refer to it as *Shortest Path (SP)* scheduling throughout this paper.

and individual peers is significantly larger than the stream bandwidth. In practice, any SPS mechanism should incorporate some redundancy (in terms of excess source and peer bandwidth, and overlay connectivity) in the system in order to deal with churn among participating peers and bandwidth fluctuations.

However, for the evaluation of scheduling schemes in this paper, we only consider *resource-constrained* settings where source bandwidth is sufficient to send only a single copy of the stream and the aggregate bandwidth contributions of peers equals the overall demand. (*i.e.*, resource index $RI = 1$ is 1). This results in a meaningful evaluation of scheduling schemes since it limits the effect of excess resources on their observed performance and reveals their true capability to manage available resources. We also consider more realistic scenarios by examining the effect of peer and bandwidth dynamics in Section VIII.

III. Achieving ISP-Friendliness

To become ISP-friendly, a SPS mechanism for live video should reduce the volume of external traffic for individual ISPs that host participating peers, without compromising the delivered quality. Intuitively, we can achieve this goal by changing the overlay connectivity, block scheduling or both. However, for any of these three approaches it is important to consider the dependency between the block-scheduling scheme and the connectivity structure of the overlay. In this section, we examine this dependency in the context of these approaches as follows:

1) Revising Block Scheduling: Given a randomly-connected overlay, block scheduling can be revised to primarily utilize connections between internal peers for pulling required blocks and use external connections in a demand-driven fashion only when there is not an adequate number of new blocks among internal parents [7], [8]. Since each new block must be initially pulled into each ISP through external connections, peers with at least one external connection should incorporate an adaptation mechanism to strike the balance between limiting the aggregate cross-IPS traffic and ensuring the delivery of the stream at a high quality to the ISP. The achieved reduction in the external traffic depends on the overall behavior of the adaptation mechanism across peers with external connections; using coordination among peers with external connections could be expensive, and without such coordination the observed performance is likely to depend on several parameters such as the number of ISPs, peers per ISPs and buffer size.

2) Revising Overlay Connectivity: An alternative approach to reduce the volume of external traffic for each ISP is to reduce the number of its external (incoming and outgoing) connections, *i.e.*, making the overlay more localized within each ISP. Prior studies (*e.g.*, [15] and [16]) have reported that in deployed P2P systems (*e.g.*, PPLive and UUSee) the overlay exhibits some level of locality, which implicitly occurs because individual peers receive faster response to their connection request from close-by peers. One explicit scheme to achieve overlay locality is to have each peer establish a fixed fraction of its connections with external peers [17]. In this approach, the amount of external traffic for each ISP linearly increases with the size of its peer population. A more effective approach is to coordinate among peers in each ISP to limit the total number of external connections for the particular ISP. As we show in the next section, localizing the overlay connectivity beyond a certain point, however, affects the performance of block scheduling and decreases the stream quality delivered to individual ISPs. In other words, without changing block scheduling there is a certain limit to the achievable reduction of external traffic through overlay localization.

3) Hybrid Approach: In this approach, both block scheduling and overlay connectivity are revised to achieve ISP-friendliness. Therefore, participating peers in each ISP maintain a fully-localized overlay and they employ a new block-scheduling scheme to ensure high delivered quality to each ISP. In a fully-localized overlay, the number of external connections for each ISP are explicitly controlled such that their aggregate bandwidth does not exceed the stream bandwidth. *In essence, this approach by definition achieves the highest level of ISP-friendliness.*

In this paper, we adopt the hybrid approach to achieve ISP-friendliness. The key challenge in this approach is to design a block-scheduling scheme that can deliver full-quality streams over a fully-localized overlay as we discuss in the following sections.

IV. Effect of Overlay Localization

In the context of overlay localization, a key question is *whether and how overlay localization affects the performance of SPS mechanisms for live video?* To tackle this question, we first define a metric for assessing the level of redundancy in the connectivity between ISPs (or localization). Then, we analytically derive the expected delivered quality to individual ISPs as a function of this level of redundancy and validate our result through simulations. Finally, we present the fundamental performance bottlenecks that affect the delivered quality as the level of redundancy varies.

A. Analysis and Simulation

Assuming all overlay connections have roughly the same bandwidth $BWPC$, maximum overlay localization (*i.e.*, fully-localized overlay) is achieved when the number of external incoming connections for each ISP is set to its minimum value of $D_{\min} = \lceil \frac{STRBW}{BWPC} \rceil$. Note that $D_{\min}$ does not depend on the population of peers in an ISP. We can quantify the level of redundancy in the external connectivity of ISP_i as :

$$R(i) = \frac{D_{\text{in}}(i) - D_{\min}}{D_{\min}}, \tag{2}$$

where $D_{\text{in}}(i)$ is the number of incoming external connections of ISP_i. In a fully-localized overlay $D_{in}(i) = D_{min}$ and thus $R(i) = 0$. Figure 2(a) shows a fully-localized overlay with $D_{\text{in}} = 2$.

For clarity of discussion, we divide peers in each ISP into two groups of: *(i) Internal peers* that are only connected to other peers in the same ISP, and *(ii) Ingress and egress (edge) peers* that have at least one incoming and outgoing external connection, respectively, to a peer in another ISP. For example, peer X and A are ingress and egress peers for ISP_1 in Figure 2(a).

To analytically examine the effect of localization on the delivered quality, we consider an overlay with N peers in a *resource-constrained* setting (as described in Section II). Suppose all overlay connections have the same bandwidth $BWPC$ and there is no churn among peers. Then, all the blocks that the source delivers to its particular child traverse the same delivery tree because of the deterministic nature of SP scheduling. We refer to a group of blocks that are delivered through the same delivery tree as a *substream*. In this simplified case, SP scheduling can be used at the granularity of substreams (instead of blocks) by pulling a substream with the minimum overall hop count (OHC) from each parent. Further, the delivered quality to each peer or ISP can be measured by the number of distinct substreams that it receives.

Given D distinct substreams, our goal is to derive the expected number of distinct substreams that reach each ISP (*i.e.*, delivered quality to each ISP) as a function of its number of incoming external connections K and the total number of peers N with a random connectivity among ISPs. Let $D(i)$ be a random variable representing the number of distinct substreams reaching ISP i and let $\delta_i(t)$ be a random indicator variable where:

$$\delta_i(t) = \begin{cases} 1 & \text{if substream } t \text{ is selected at least once in } K, \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that the external peers are uniformly distributed across the D available substreams, there are $N - \frac{N}{D}$ peers not in the t-th substream. The probability of not connecting to any of these peers in K tries is then given by:

$$\begin{aligned} \Pr\left(\delta_i(t) = 0\right) &= \frac{\binom{N-\frac{N}{D}}{K}}{\binom{N}{K}} \\ &= \frac{\left(N - \frac{N}{D}\right)!\,(N-K)!}{N!\,\left(N - \frac{N}{D} - K\right)!} \\ &= \prod_{j=0}^{\frac{N}{D}-1} \left(1 - \frac{K}{N-j}\right) \qquad (3) \end{aligned}$$

From linearity of the expectation operator and the equality of the expected value of the indicator function with its probability, we can derive the expected number of distinct subtrees as follows:

$$\begin{aligned} E[D(i)] &= E\left[\sum_{t=1}^{D} \delta_i(t)\right] = \sum_{t=1}^{D} E\left[\delta_i(t)\right] \\ &= D\left(1 - \prod_{j=0}^{\frac{N}{D}-1}\left(1 - \frac{K}{N-j}\right)\right) \qquad (4) \end{aligned}$$

To validate the above analysis, we simulate SP and random block scheduling (as a reference) over a resource-constrained overlay with 5000 peers uniformly distributed over 40 ISPs; in- and out-degree of all peers are 12. We have repeated these simulations for different peer and ISP degrees and the results are very similar. Figure 1 depicts the expected value of the number of distinct substreams normalized by the total number of substreams (*i.e.*, $\frac{E[D(i)]}{D}$) as well as the median of the delivered quality to each ISP in our simulations for SP and random schedulings as a function of the redundancy $R(i)$.

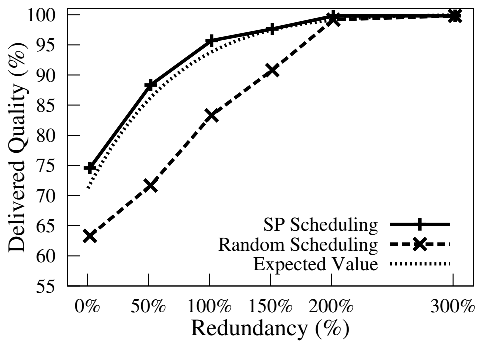


Fig. 1. Delivered quality to ISPs with SP and Random scheduling, along with the expected value of the number of distinct substreams normalized by the total number of substreams (*i.e.*, $E[D(i)]/D$).

Figure 1 not only shows that the analytical and simulation results for SP scheduling are consistent but it also illustrates the following important points:

- The delivered quality to individual ISPs by SP scheduling over a fully-localized overlay drops by more than 25%.
- As redundancy in the external connectivity of ISPs increases (*i.e.*, the overlay localization decreases), the overall performance of content delivery improves. SP scheduling can deliver a high quality stream to ISPs over a localized overlay only when the number of external incoming connections is 3 times that of its minimum value $D_{\min}$ (*i.e.*, $R(i)$ = 200%).[3]
- Compared to SP scheduling, random scheduling results in a lower delivered quality to each ISP over a highly-localized overlay.
- For any large population of peers, the effect of redundancy on the delivered quality of SP scheduling does not vary with the degree or peer population (as Eqn. 4 indicates). Thus, these findings demonstrate the fundamental performance bottleneck caused by overlay localization.

B. Fundamental Performance Bottlenecks

We closely examined our simulation results to gain a deeper insight on the effect of overlay localization on the performance of SP scheduling and identified the following two main problems for content delivery:

1) Misallocation of External Connections: SP scheduling may result in an improper allocation of external connections for pulling certain substreams. This in turn limits the delivered quality to individual ISPs in two ways: *(i)* It affects the availability of some substreams to other ISPs. Consider the fully-localized overlay in Figure 2(a) for the delivery of two substreams S_1 and S_2. Given the overall hop count (OHC) of both substreams at edge peers A and B, both edge peers C in ISP_3, and D in ISP_4, pull S_2 through their incoming external connections from ISP_1. This means that the delivery tree for S_1 is terminated at ISP_1 and this substream cannot reach other

[3] It is worth mentioning that the delivered quality to peers even with 200% redundancy is still low.

ISPs. *(ii)* Since ingress peers within each ISP independently determine the substreams they pull from their external parents, it is likely that not all substreams are collectively pulled into the ISP by its ingress peers.

2) Misallocation of Internal Connections: Even if all substreams are delivered to ingress edge peers of an ISP, SP scheduling does not ensure proper propagation of substreams from their corresponding ingress edge peer to all the internal peers. To demonstrate this point, consider the ISP in Figure 2(b) whose edge peers A and B pull S_1 and S_2 with OHC of 10 and 2, respectively. Furthermore, peer A pulls S_2 from an internal peer B. Since $OHC_2(A) = 3 < OHC_1(A) = 10$, SP scheduling at internal peers C and D pulls S_2 from peer A. This in turn makes S_1 unavailable for other internal peers.

Note that the above performance bottlenecks are not specific to SP scheduling and are even more likely to occur with other scheduling schemes as the performance of random scheduling indicates in Figure 1.

V. OLIVES: An Overview

In this section, we present an overview of our proposed ISP-friendly live P2P streaming, called *OLIVES* [4]. OLIVES adopts a hybrid approach to achieve ISP-friendliness. Participating peers in each ISP maintain a fully-localized overlay and incorporate an overlay-aware scheduling to ensure the delivery of high-quality streams to each ISP and its peers. Since all connections have roughly the same bandwidth, the external traffic for individual ISPs is directly controlled by minimizing the number of ingress and egress connections such that both the aggregate traffic into and out of the ISP are close to the stream bandwidth.

A. Maintaining a Localized Overlay

In OLIVES, participating peers maintain a two-tier overlay where the overlay connection within each ISP form the bottom tier and the external connections between ISPs compose the top tier. Each ISP deploys a local tracker that coordinates internal and external connectivity of its peers to ensure full localization of the overlay even in the presence of churn. The local trackers discover peers external to their respective ISP through a global tracker. This also enables an ISP to enforce any policy by directing external connections towards preferred ISPs (*e.g.,* [3], [18]) at the local tracker.

Upon departure of an edge peer C with an ingress connection from parent peer A, the local tracker prompts another local peer (with the desired properties) to establish the ingress connection with parent peer A. If establishing connection to parent peer A is not successful or peer A departs, local tracker obtains a new external peer through the global tracker. In practice, it is preferred to select more stable peers to act as edge peers for each ISP in order to minimize churn in the top-level overlay. We note that OLIVES can be used in a simpler setting where individual ISPs offer a service to their customers by providing stable and well-provisioned proxy servers that can act as edge peers.

Constructing and maintaining localized P2P overlays is a well-studied and understood topic (e.g., in [7], [17]), and is not one of the contributions of OLIVES. Thus, we omit a detailed discussion of this topic and focus on the content-delivery component in OLIVES.

B. Two-Tier Scheduling

The main contribution of OLIVES is a two-tier block-scheduling scheme that maximizes the delivered quality to peers in a fully-localized overlay. The proposed scheduling is motivated by the performance bottlenecks that we identified in Section IV-B. To properly operate on the two-tier overlay structure, content delivery in OLIVES is also performed at two coherent levels (or tiers) as follows:

- *Inter-ISP Scheduling*: This scheduling scheme runs only at the ingress edge peers of each ISP and manages content pulled through external connections (*i.e.*, top-tier overlay) to ensure the delivery of full-quality stream to individual ISPs.
- *Intra-ISP Scheduling*: This scheduling scheme runs at all peers and only manages content pulled through internal connections (*i.e.*, bottom-tier overlay) to ensure the delivery of full-quality streams from edge peers to all other peers within an ISP.

We present these two scheduling schemes and their interactions in details in Section VI.

C. Reporting and Requesting Blocks

The basic logistics for reporting available blocks to child peers and requesting required block from parents in OLIVES is similar to other SPS mechanisms: Each peer maintains an exponentially weighted moving average (EWMA)[5] of the bandwidth $bw(q)$ from each parent q and periodically reports its newly available blocks to its children. Inter- and intra-ISP scheduling are periodically invoked to identify $n = \frac{bw(q)*\tau}{BlkSize}$ blocks to be requested from parent q where $BlkSize$ denotes the size of each block. Requested blocks from parents are sorted and thus delivered based on their timestamps.

VI. Two-tier Scheduling in OLIVES

The two-tier scheduling in OLIVES incorporates the two following ideas: *(i)* OLIVES leverages SP scheduling (described in Section II) for both inter- and intra-ISP scheduling. OLIVE's inter-ISP scheduling scheme implements SP scheduling at the ISP level by assessing the shortest path as the number of visited ISPs by each block. This approach minimizes the depth of the ISP-level delivery tree (*i.e.*, the number of visited ISPs) for individual blocks. The intra-ISP scheduling scheme adopts the notion of SP scheduling within the scope of individual ISPs by using the number of visited peers counted from the local ingress edge peer as a measure of distance. This approach minimizes the depth of the delivery tree from an edge peer to all internal peers within an ISP. *(ii)* Inter-ISP scheduling scheme groups blocks into substreams

[4] The name is an abbreviation of Overlay-aware LIVE P2P Streaming mechanism.

[5] EWMA offers a good predictor of congestion-controlled bandwidth for each connection in the near future [6].

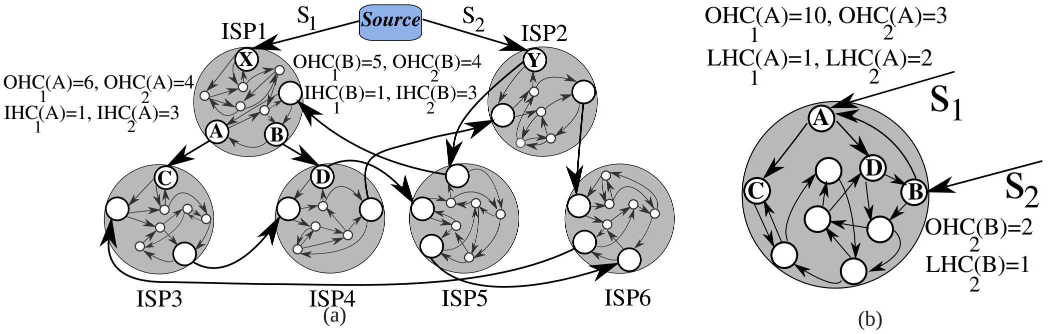


Fig. 2. (a) A fully-localized overlay with 6 ISPs and in-degree of 2. (b) An Intra-ISP view of the overlay; OHC represents the total number of peers that a block has visited; LHC represents the number of internal peers that a block has visited within each ISP; IHC represents the number of ISPs that a block has visited.

and performs SP scheduling at the substream level (see Section IV). This determines the collection of blocks that an edge peer pulls from each external parent. Inter-ISP scheduling at each peer incorporates a complementary mechanism to utilize excess bandwidth of ingress connections for pulling blocks of other substreams. We describe inter- and intra-ISP scheduling in the next two subsections in more details.

To facilitate the above ideas, all blocks of the stream are partitioned into multiple *substreams* based on the outgoing connection that is used by the source to deliver each block. In essence, the stream is logically split into D substreams if the source has D child peers. For example, all the blocks that the source delivers to peer X in Figure 2(a) are associated with substream S_1.[6] Each block i carries its associated substream ID and the three hop-counts that are defined at peer p as follows:

- *ISP Hop Count* $IHC_i(p)$ keeps track of the number of ISPs that a block has visited.
- *Local Hop Count* $LHC_i(p)$ keeps track of the number of internal peers that a block has visited within the current ISP. LHC is reset by the ingress peer that pulls the block into the ISP.
- *Overall Hop Count* $OHC_i(p)$ keeps track of the total number of peers that a block has visited.

A. Inter-ISP Scheduling

The main idea for inter-ISP scheduling is to perform SP scheduling at the ISP level. If we replace each ISP with a single peer, the resulting top-tier overlay has a random connectivity. Thus, SP scheduling can be used for content delivery over the top-tier overlay (by using IHC in each block) if each ISP behaves as a single peer. To achieve this goal, an ISP periodically pulls newly available blocks through its ingress connections to maximize the quality of the stream delivered to itself. This requires periodic block-level coordination among ingress peers if their aggregate ingress bandwidth is limited to the stream bandwidth. For example, each of the ingress edge peers of ISP_1 in Figure 2(a) has one external connection and they closely coordinate to pull mutually exclusive blocks from their external parents. In order to keep the coordination overhead among the edge peers at a minimum level, OLIVES resorts to infrequent coordination at the substream level rather than close coordination at the block level, as we describe next.

1) Substream-Level Coordination Among Edge Peers: If edge peers of participating ISPs remain relatively stable, the top-tier overlay exhibits only infrequent changes during a session.[7] This implies that most blocks of a particular substream follow the same *ISP-level delivery tree* and have the same IHC value at a given external parent.[8]

OLIVES leverages this important observation to perform coordination of requested blocks among edge peers at the substream level. Each edge peer uses the most common value of IHC among available blocks of a substream s at its external parent q to infer $IHC_s(q)$ of that substream. A coordination event is triggered only by the following events infrequent events: *(i)* a peer is replacing a departing edge peer, *(ii)* an existing ingress edge peer changes its external parent, which affects its observed IHC values for some substream, or *(iii)* the common value of $IHC_s(q)$ for a substream changes due to a change in the top-tier overlay between upstream ISPs.

We assume that all edge peers of an ISP know about each other in order to perform coordination in a distributed fashion. When an ingress peer detects any of the above events, it reports an updated list of IHC values (for all substream) to all other ingress peers in the same ISP. Upon receiving this information, each edge peer performs SP scheduling using the current IHC values reported from all ingress peers of the same ISP. Given the deterministic nature of SP scheduling, all edge peers converge onto the same mapping of substreams to external connections so that each can uniquely identify its *designated substream(s)*. [9]

Figure 2(a) demonstrates the behavior of the proposed Inter-ISP scheduling in mapping two substreams to external connections by showing the values of their IHC at the egress edges of ISP_1. The inter-ISP scheduling in ISP_3 and ISP_4 assign substream S_1 to edge peers C and D, which they pull from their respective external parent, namely A and B. We

[6] We emphasize that the substream abstraction offers a logical grouping of blocks that is not related to the stream content or its encoding. Also different substreams may have slightly different bandwidth.

[7] Based on previous studies on stability of peers in P2P systems [19], we assume a peer is stable if it's life time exceeds 40% of the session time.

[8] Similar to node-level delivery trees that we defined in Section II, an ISP-level delivery tree is a source-routed tree that shows the order of delivery of each block to all ISPs in the overlay. Each ISP is represented with a node in the tree.

[9] Alternatively, the coordination can also be performed in a central fashion by sending all the updates to the local tracker that executes the SP scheduling and informs all associated ingress peers about their designated substreams.

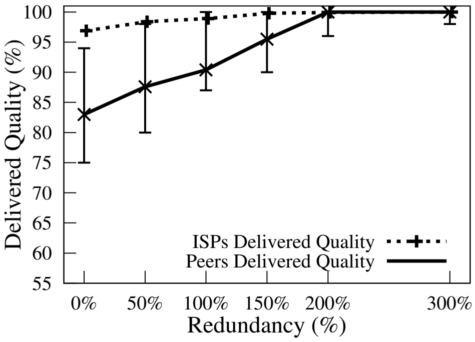


Fig. 3. Effect of Inter-ISP scheduling; median of delivered quality with Inter-ISP scheduling when number of peers and ISPs are 5000 and 40, respectively, and peer degree is 12.

examine the effect of inter-ISP scheduling on the delivered quality to individual ISPs through simulations using the same settings as discussed in Section IV.

Figure 3 presents the median of the delivered quality to ISPs and peers (along with the 10th and 90th percentile of the delivered quality to peers) as a function of the redundancy in external connectivity of individual ISPs. This result reveals that our proposed inter-ISP scheduling increases the delivered quality to ISPs over a fully-localized overlay (*i.e.*, redundancy of 0) from 75% to 97%. However, Figure 3 also shows that the median of the delivered quality to individual peers is limited to 83% due to the lack of a scheduling mechanism that optimizes delivery of blocks from an ingress peer to all other peers within the same ISP. We address this issue and present a solution in the next section on intra-ISP scheduling

2) Coping With Variation of Per-Connection Bandwidth: At each scheduling event, individual edge peers request newly available blocks of their designated substreams from their external parents. We call these designated blocks. However, the variations of average congestion-controlled bandwidth for individual connections across different scheduling periods τ imply that the number of designated blocks n_d may be different from the number of blocks n that can be pulled from an external parent given the available bandwidth of the corresponding connection. This leads to the following two scenarios:

- *Limited or adequate connection bandwidth* ($n \leq n_d$): In this case, the ingress edge peer can only request (a subset of) blocks from the designated substream.
- *Excess connection bandwidth* ($n > n_d$): In this case, the edge peer has excess bandwidth for pulling blocks from non-designated substreams that are not requested by other edge peers.

To fully utilize the excess bandwidth of external connections, one needs to address the following two important questions at each scheduling event: *(i)* How can each edge peer identify unrequested blocks of non-designated substreams? *(ii)* How can these unrequested blocks be properly assigned to edge peers with excess bandwidth? The simple solution of periodically coordinating among all ingress peers at the block-level is expensive and not desirable. We note that since the aggregate bandwidth capacity of ingress connections for each ISP is equal or larger than the stream bandwidth, the aggregate excess bandwidth is theoretically sufficient to pull all unrequested blocks.

OLIVES incorporates an *implicit* approach to identify unrequested blocks, *i.e.*, blocks that have not been delivered to the ISP yet. Each ingress peer leverages the unavailability of a block of a non-designated substreams with an sufficiently early timestamp among all of its internal parent peers as an *implicit but reliable hint* that the block has not been requested by its corresponding edge peer.

When a scheduling event is triggered, an edge peer with excess ingress bandwidth examines the available blocks among its internal parents and identifies the largest timestamp $ts_{max}(s)$ for each non-designated substream s. Then, it subtracts one interval τ from these maximum timestamps, which provides it with a conservative maximum threshold $ts_{th}(s)$ = $ts_{max}(s)$ - τ for the timestamps of blocks of each substream that must have been propagated to these internal parents by now. Blocks of a non-designated substream s with a timestamp lower than $ts_{th}(s)$ that are unavailable at 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 as we have mentioned earlier; and *(ii)* available blocks among internal parents represent the availability of content for a large fraction of peers due to the random connectivity within the ISP.

Once individual ingress peers have independently identified unrequested blocks, they determine responsibility for these blocks as follows: *Given the IDs of individual substreams, each incoming edge peer utilizes its excess bandwidth to pull unrequested blocks of non-designated substreams in a prioritized fashion using a circular order of substreams.* For example, in a scenario with 4 substreams, the designated edge peer for pulling substream 3 uses its excess bandwidth to pull all identified missing blocks for substream 4, then uses any remaining excess bandwidth for substream 1, and finally for substream 2.

This method can effectively manage the utilization of excess bandwidth among edge peers without requesting duplicate blocks for two reasons: First, bandwidth deficit and surplus among external connections are often small, short-lived and move among them. Second, each edge peer detects missing blocks of a non-designated substream s after a certain delay, which is proportional to the peer's distance from the corresponding edge peer of s. This adds a random delay to the reaction of edge peers and reduces the probability of duplication among requested blocks by this mechanism.

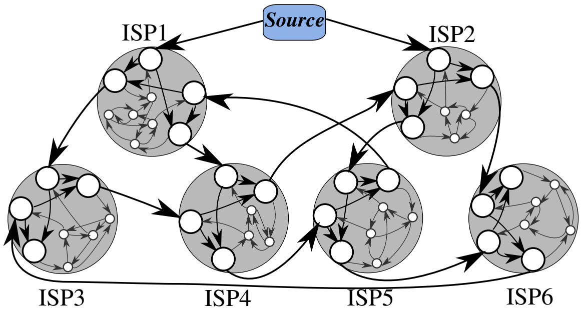


Fig. 4. A fully-localized overlay with shortcuts.

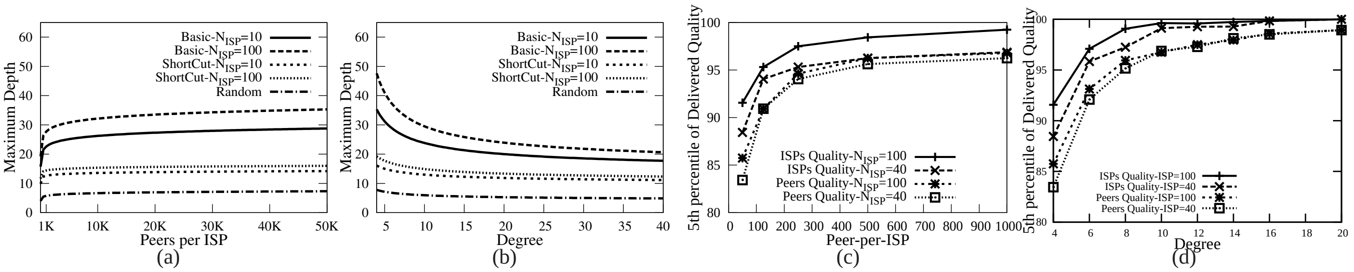


Fig. 5. (a) and (b) show d_{max} for a fully-localized overlay and a random overlay as a function of number of peers per ISP and peer degree, respectively. (c) and (d) show fifth percentile of delivered quality to peers and ISPs as a function of number of peers per ISP and peer degree, respectively.

B. Intra-ISP Scheduling

The goal of intra-ISP scheduling is to deliver each block from the designated edge peer to all internal peers of an ISP. In essence, each ingress peer is treated as the local designated source for the blocks that it pulls into the ISP. Therefore, OLIVES applies the idea of SP scheduling for individual blocks based on their relative local hop count LHC from the corresponding edge peer. Once every τ seconds, each internal peer considers the available blocks among its parents and pulls the blocks with the smallest LHC values from the corresponding parent peers.

Figure 2(b) demonstrates the behavior of the proposed intra-ISP scheduling by showing the average values of LHC for both substreams at peers A and B. In this case, peers C and D use LHC and pull blocks of substream S_1 from the edge peer A, regardless of the total hop count from the source. As our evaluation results from simulations in Section VII show, the delivered quality to peers over a fully-localized overlay reaches 95% with our scheduling scheme.

C. Shortcutting of ISPs

The depth of the delivery trees (and thus the buffer requirement) in OLIVES is affected by the connectivity structure of the overlay and the behavior of inter- and intra-ISP scheduling. We derive the depth of delivery trees in OLIVES and propose a mechanism to reduce the buffer requirement.

The maximum depth of delivery trees d_{max} in OLIVES is the product of two components: *(i)* the maximum depth of the *ISP-level* delivery trees, and *(ii)* the maximum depth of the delivery trees from each ingress peer to all internal peers within an ISP. Since the ISP-level overlay and connectivity within each ISP are both random, d_{max} in terms of hops can be derived by extending Eqn. 1 as follows [20]:

$$d_{max} \leq (\log_D N_{ISP} + 3) \times (\log_D (\frac{N}{N_{ISP}}) + 3) \qquad (5)$$

where N_{ISP} is the number of ISPs. Figures 5(a) and 5(b) plot d_{max} in a localized overlay (using Eqn. 5) and a comparable random overlay as a function of the number of peers per ISP and peer degree, respectively. These figures show that d_{max} in OLIVES, is larger than d_{max} in a comparable random overlay. *In essence, OLIVES maximizes the delivered quality to all ISPs despite the limited inter-ISP connectivity in fully-localized overlays at the cost of forming higher delivery trees and a larger buffer requirement.*

The larger buffer requirement is a disadvantage and more importantly, a large ISP in the overlay can increase d_{max} for all peers. To reduce the buffer requirement, OLIVES adopts the idea of *shortcutting of ISPs*. The basic idea is to minimize the distance between ingress and egress edge peers of an ISP by controlling overlay connectivity internal to the ISP. To this end, each egress peer selects *all* ingress peers as its parents, which is shown in Figure 4. The mesh-like internal connectivity among all edge peers enables each egress peer to provide any block to other ISPs even when the mapping of ingress connections is being changed by the coordination mechanism. d_{max} with shortcuts can then be derived as follows [20]:

$$d_{max} \leq 2\log_D N_{ISP} + \log_D(\frac{N}{N_{ISP}} - D) + 8 \qquad (6)$$

Figures 5(a) and 5(b) demonstrate the ability of shortcutting to effectively reduce d_{max} across the parameter space. Clearly, maintenance of connectivity among edge peers in the presence of churn, which can be performed by the local tracker, contributes to the cost of shortcutting.

VII. Evaluation: Effect of Overlay Connectivity

In this section, we examine how properties of a fully-localized overlay affect the overall performance of the two-tier scheduling in OLIVES, using session-level simulations with substream abstraction. To this end, we run SPS scheduling over overlays generated for a given number of participating ISPs, peers per ISP, peer connectivity and link bandwidth. Peer population and per-link bandwidth are assumed to not exhibit any dynamics, and unless indicated otherwise, the capacity of every connection equals the substream bandwidth. Thus, once a link is utilized for a particular substream it cannot participate in the delivery tree of another substream.

We only focus on resource-constrained settings (as described in Section II) to stress-test OLIVES. Since OLIVES limits the volume of external traffic by controlling the number of ingress and egress connections, the overall external traffic of each ISP is always limited to $STRBW$. Hence, we focus on the delivered stream quality and depth of delivery trees as performance metrics. We will examine the effect of bandwidth dynamics in the next section.

A. Peer Degree, ISP & Peer Population

We start by examining the effect of the following three parameters that primarily determine the overall connectivity of an overlay: peer degree, the number of ISPs and the number of peers.

Figure 5(c) depicts the 5th percentile (a lower bound) of delivered quality to peers and ISPs as a function of peers

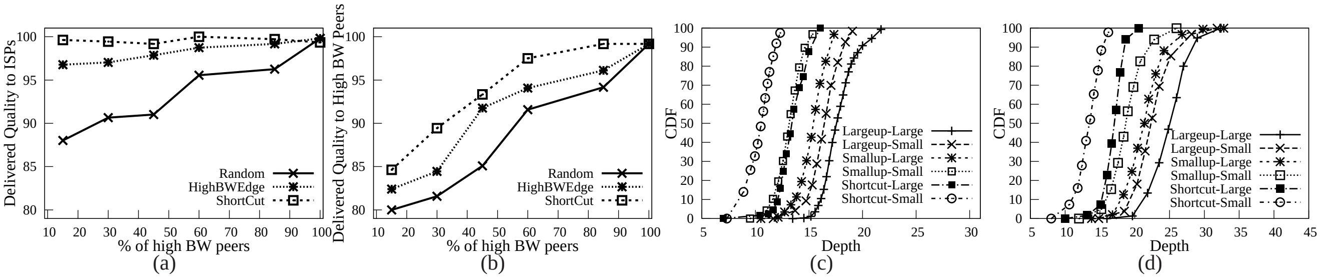


Fig. 6. Fifth percentile of delivered quality (a) to ISPs and (b) to high-bandwidth peers as functions of the percentage of high-bandwidth peers. CDF of (c) the average and (d) the maximum depth of peers in overlays with various criteria for placing ISPs with different sizes.

per ISP. The in- and out-degrees of all peers and ISPs are 4. Each line shows the delivered quality for a certain number of ISPs, namely 40 and 100 ISPs. Figure 5(c) shows that increasing the number of peers per ISP or the number of ISPs improves the delivered quality to ISPs and peers. To explain this, we note that as the population of peers in the overlay increases, it becomes less *clustered* especially when peer degree is small. A lower level of clustering provides more flexibility for formation of delivery trees and, thus, yields higher delivered quality.

Figure 5(d) depicts the 5th percentile of delivered quality to peers and ISPs as a function of the peer degree for a varying number of ISPs in the overlay when the number of peers per ISP is 50. This figure reemphasizes that an increasing peer degree improves delivered quality by improving overlay connectivity at both inter- and intra-ISP levels, which in turn facilitates the formation of delivery trees. *In summary, OLIVES can deliver high quality streams regardless of peer degree, peers per ISP and ISPs in fully-localized overlays. The quality drops only slightly in corner scenarios where all three parameters are small.*

B. Heterogeneous Peer Bandwidth

We now turn our attention to overlays with heterogeneous but symmetric peer bandwidth. In particular, we consider an overlay of 20,000 peers evenly grouped into 40 ISPs with the degree of high and low bandwidth peers set to 12 and 6, respectively. We vary the percentage of high bandwidth peers from 15% to 85%. We assume that the stream is encoded using Multiple Description Coding (MDC), and the delivered quality to each peer is proportional to its incoming bandwidth. We focus on the delivered quality to high-bandwidth peers as they are supposed to receive all substreams in contrast to low-bandwidth peers, which receive half of the substreams.

Figures 6(a) and 6(b) show the 5th percentile of delivered quality to ISPs and peers as a function of the percentage of high-bandwidth peers. In these figures, each line represents one of the following strategies for selection and connectivity of edge peers: *(i) Random*—random peers are selected as edge peers; *(ii) HighBWEdge*—high-bandwidth peers are selected as edge peers; *(iii) Shortcutting*—high-bandwidth peers are selected as edge and shortcutting is used. Figure 6(a) indicates that randomly picking edge peers reduces the delivered quality to some ISPs. The key problem is that *when a high-bandwidth peer has one or more low-bandwidth parents, it becomes more difficult for the scheduling to map the required substreams to parent peers because not every substream is available at every parent.*

With the *Random* strategy, the problem with low-bandwidth parents occurs in both inter- and intra-ISP scheduling, which reduces the delivered quality to ISPs. The *HighBWEdge* strategy 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 affects the intra-ISP scheduling as can be seen by the 5th percentile of the delivered quality to ISPs in Figure 6(a).*Shortcutting* eliminates this latter problem and maximizes the delivered quality to all ISPs, but the delivered quality to peers is lower since high-bandwidth peers may still have internal low-bandwidth parents. Moreover, increasing the percentage of high-bandwidth peers reduces the probability of having a low-bandwidth peer as an edge peer or internal parent, which leads to a higher delivered quality as shown in Figures 6(a) and 6(b).

C. Location of Small and Large ISPs

Next we investigate whether and how the location of large ISPs in the overlay affects the overall depth of delivery trees for all peers. We consider a more realistic scenario by using traces of a P2P streaming application. We use a sample snapshot of PPLive with 50K peers crawled in July 2009, and by mapping the IP addresses to ASes, we determine that the snapshot consists of 970 ASes (or ISPs) with a skewed distribution of peers per ASes (85% of peers are in 10% of ASes). We assume the stream bandwidth to equal 530 Kbps and the incoming bandwidth of peers to be sufficient to receive the full-quality stream. The outgoing bandwidths of 15%, 35% and 50% of peers is set to 128Kbps, 384Kbps and 768Kbps, respectively, inspired by a recent empirical study [21]. We set the in-degree of peers to 12 and the out-degree of peers with 128Kbps, 384Kbps, 768Kbps to 3, 9 and 18, respectively. We examine two scenarios by placing large and small ISPs close to the source, which we call *Largeup* and *Smallup*, respectively.

Figure 6(c) and 6(d) depict the distribution of average and maximum depth of peers across different delivery trees in small and large ISPs for the above scenarios and the shortcut overlay, respectively. For example, the distribution labeled as *Largeup-small* shows the depth of delivery trees for small ISPs when large ISPs are placed close to the source. These figures illustrate that while the average and maximum depth of delivery trees are generally lower for peers in smaller ISPs, in the Largeup scenario the average and maximum depth of both

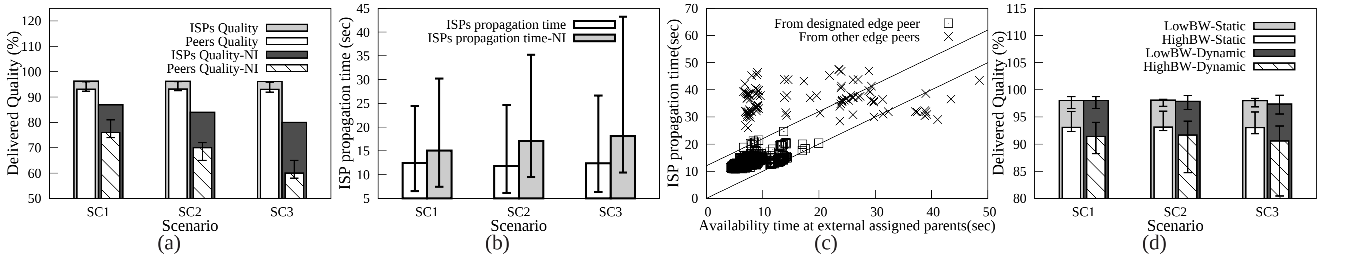


Fig. 7. (a) Delivered quality to ISPs and peers in three scenarios with RTT heterogeneity. (b) Propagation time to the ISP in the three RTT scenarios with and without implicit identification. (c) A scatter plot of block propagation time to the ISP through designated or other edge peers. (d) Delivered quality to high- and low-bandwidth peers in presence of churn.

groups of peers are larger than in the Smallup and shortcut scenarios. In essence, if large ISPs are positioned close to the source, the delivery trees of most peers must cross through large ISPs. Therefore, the larger distance between the ingress and egress peers in large ISPs appear in the upper portion of most of the delivery trees. On the other hand, large ISPs in the Smallup scenario are in random distances (>1) from the source, thus, appear in the upper portion of a smaller fraction of delivery trees. This results in a lower average depth for delivery trees compared to the Largeup scenario.

Figures 6(c) and 6(d) also show the depth of delivery trees for a shortcut overlay, which demonstrates two interesting points: First, the maximum depth of delivery trees and the buffer requirement for both groups of peers have significantly decreased in shortcut overlays compared to Smallup and Largeup scenarios. Second, Figure 6(d) shows a clear gap between the maximum depth of delivery trees for peers in small and large ISPs. *Through shortcutting, peers in smaller ISPs typically have a much smaller maximum depth than peers in large ISPs, thus, require proportionally less buffering.*

VIII. Evaluation: Bandwidth & Peer Dynamics

In this section, our main goal is to evaluate the performance of OLIVES intra- and inter-ISP scheduling in 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 performance bottlenecks. The physical topology is generated using Brite [22] with 15 ASes and 10 routers per AS in top-down mode.

We consider an overlay with 101 ISPs where one target ISP has 100 heterogeneous peers; all other ISPs are simulated as a single peer. This abstraction enables us to examine the effect of packet-level dynamics on the performance of content delivery to a single ISP within the limited scalability of *NS2* simulator.[10] 85% of the peers in the target ISP are connected with low symmetric link-bandwidth of 750Kbps and the rest of peers with high bandwidth of 1.5Mbps. Peers in these two groups maintain an in-degree and out-degree of 10 and 20, respectively.

The video stream has a bandwidth of 1.5 Mbps and is MDC-encoded into 10 descriptions of 150 Kbps each; thus, high- and low-bandwidth peers ideally receive a subset of 10 and 5 descriptions, respectively. The source has a bandwidth of 1.6 Mbps to provide for delivery of the full-quality stream despite potential packet losses and all connections are TCP-friendly congestion controlled using RAP [10] on top of UDP. Each peer starts playing the content as soon as it buffers $d_{max} * \tau$ seconds worth of data for at least one description. Received blocks for each description are buffered separately, organized based on their playout time and drained at a constant rate equal to the description bandwidth. The receiver keeps track of the number of unique descriptions delivered for each played-out block to determine an average of its delivered quality.

We focus on the delivered quality to high-bandwidth peers since low- bandwidth peers receive full-quality streams (proportional to their incoming bandwidth) in all scenarios. The interval for periodic scheduling τ is set to 6 seconds; however, the presented results are not sensitive to the choice of the scheduling interval [6]. Each simulation is run for 2000 simulated seconds and the presented results are averaged over 10 runs with different random seeds.

A. Per-Connection Bandwidth Heterogeneity

We increase the diversity of the average congestion-controlled bandwidth across different connections by controlling the range of RTT values. Towards this end, we consider three reference 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 7(a) shows the delivered quality to the target ISP, and the median (and bars for 5th and 95th percentiles) of the delivered quality to its high-bandwidth peers with implicit identification and without it (labeled as “-NI”) in all three reference scenarios. This figure reveals that the implicit identification mechanism can deliver high-quality streams to all peers despite the increasing level of heterogeneity of average bandwidth among overlay connections. However, in the absence of implicit identification, the delivered quality to the ISP shows a moderate decrease while its gap with the delivered quality to high-bandwidth peers quickly widens with the level of bandwidth heterogeneity. Furthermore, the percentage of duplicate blocks pulled into the ISP is effectively limited to below 1.2% with implicit identification but it varies between 9% and 13% without it.

To explain this, Figure 7(b) depicts the median time (and bars for 5th and 95th percentiles) between the generation of a block at the source and its first arrival at an ingress peer in the target ISP (*i.e.*, propagation time) for the three scenarios. Figure 7(b) indicates that in the absence of identification, the blocks experience a longer propagation time, and this differ-

[10] We have obtained consistent results in simulations with smaller numbers of ISPs and without the abstraction.

ence further grows with the heterogeneity of per-connection bandwidth. *Overall, these results shows that despite large variations in connection bandwidth, the implicit identification in inter-ISP scheduling can effectively utilize excess ingress bandwidth by pulling the missing blocks in a timely manner.*

B. Behavior with Implicit Identification

In scenario *SC1*, an average of 10% of blocks across all substreams are pulled into the ISP by a non-designated ingress peer. 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 identification.

Figure 7(c) is a scatter plot of the propagation time of blocks to an external parent of the target ISP (on the x-axis) versus its propagation time to an ingress peer that first pulls the block (on the y-axis) for all blocks of a sample substream in scenario SC3. Blocks that enter the ISP through their designated edge are marked with a square whereas blocks pulled by other edge peers are marked with an ”X”. This figure illustrates the relative time for availability of a block outside and inside the target ISP. Roughly 90% of blocks 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 7(c) clearly shows that the gap between the propagation time of these blocks outside and inside the ISP is very small (all points are between lines of $y = x$ and $y = x + 2 * \tau$).

Blocks that are pulled into the ISP by non-designated peers can be divided into two groups: *(i)* Blocks that quickly become available at the designated parent but are requested through other external parents after some delay. These blocks were not requested from the designated parent due to the short-term bandwidth deficit of the corresponding external connection. *()ii* Blocks that become available rather late at the designated parent. In essence, the designated edge peer experiences a *content bottleneck* for these blocks and fails to pull them from its external parent.

C. Peer Dynamics

To evaluate the performance of OLIVES in the presence of peer dynamics, we incorporate churn in the three reference scenarios using the churn model reported in [23], [24]. Towards this end, we select peer-session times from a log-normal distribution (μ=4.29 and σ=1.28) and peer inter-arrival times from a Pareto distribution (a=2.52 and b=1.55). In presence of churn, the average aggregate incoming bandwidth to the target ISP could be dropped if an edge peer leaves the session. To cope with this problem in practice, individual ISPs should maintain a level of redundancy in their ingress connectivity proportional to the observed level of churn. Examination of techniques to determine the required level of redundancy as a function of churn is part of our future work. For our simulations, we empirically determined the required level of redundancy for external connectivity of the target ISP to be 10% so that the aggregate incoming bandwidth in the presence of churn is roughly the same as the stream bandwidth.[11]

Figure 7(d) depicts the median (and bars for 5th and 95th percentiles) of the delivered quality to low- and high-bandwidth peers in the target ISP with and without churn labeled as “dynamic” and “static”, respectively. This figure indicates that the performance of low-bandwidth peers is not affected by churn. The median and 95th percentile of the delivered quality to high-bandwidth peers in presence of churn are very similar to the values obtained in the static setting across all scenarios. However, the delivered quality to a small fraction of high-bandwidth peers (5th percentile) slightly decreases in presence of churn and further decreases in scenarios with larger heterogeneity in per-connection bandwidth (*i.e.*, SC3).

In presence of churn, some parent peers might have recently joined the session; thus, block availability among all parents can give an incomplete view of the total internal block availability. Essentially, this can result in an inaccurate identification of non-requested blocks, which leads to more request for duplicate blocks (2%, 5% and 6% for SC1, SC2 and SC3, respectively). *Overall, the resulting change in the delivered quality due to churn is minimal (less than 10%), which shows the ability of the implicit identification mechanism to achieve good performance even in presence of peer dynamics.*

IX. Related Work

P2P traffic localization has received a great deal of attention as a remedy for costly inter-ISP traffic generated by P2P applications. In this section, we discuss related work on P2P traffic localization regardless of the targeted application. Since all P2P applications consist of two main components for overlay construction and content delivery, and existing works typically focus on either one or the other, we structure the following discussion accordingly.

A. Overlay Construction

The majority of existing works on P2P traffic localization focus on the construction of localized overlays by suggesting various neighbor-selection strategies. Generally speaking, their goal is to reduce inter-ISP traffic and improve application performance by selecting candidate peers that are within the same or a close-by ISP. Several studies have suggested using Internet topology maps, AS mappings or IP ranges [17], [25], [26] to derive the location of each peer.

Azureus BitTorent clients use the Vivaldi network-coordinate system [27], which is based on network embedding of latencies, to identify topologically close neighbors. The PPLive streaming system and other works [28]–[32] rely directly on delay metrics between peers for neighbor selection. In PPLive and UUSee, peers periodically exchange neighbor lists with each other and connect to the first peer responding. As measurement studies in [16] and [15] have shown, this strategy tends to provide for some level of implicit localization when latencies between peers in the same ISP are smaller than latencies between peers in different ISPs. Toptb [32] follows a more sophisticated approach for biased neighbor selection based on ping and traceroute measurements for estimating the delay, number of AS hops and available bandwidth between peers, which can reduce the number of AS hops by as much

[11] Note that we do not consider flash-crowd scenario in our simulations.

as 25%. However, the achievable levels of localization in these systems are very sensitive to the adopted neighbor selection mechanisms and the distribution of latencies between peers.

Since ISPs have best knowledge about costs, policies and traffic locality, a collaborative approach between ISPs and P2P applications has been suggested in [4] and [3]. Aggarwal et al. [4] and Bindal et al. [17] propose a network oracle that ranks a list of peers provided by P2P applications according to ISPs preferences such as cost or available bandwidth between peers. The P4P system [3] uses weighted topology maps provided by the ISP to enable a tracker and P2P applications to select local peers. More recently, the IETF has chartered the ALTO working group to define an interface between ISPs and P2P applications [33]. These works are mostly targeting P2P file sharing applications.

Instead of relying on measured topology information or cooperation by ISPs, Choffnes et al. [18] leverage the dynamic DNS redirection mechanisms of content delivery networks (CDNs) for neighbor selection. Exploiting the fact that peers redirected to the same replica server in a CDN are likely to reside in the same or a close-by ISP, the authors were able to locate peers within the same AS in one third of the time and measure a median value of one for the number of AS hops for the suggested neighbors.

Common to all of the above works is their focus on localized overlays. In contrast, we have taken a more comprehensive approach and among others show the fundamental limitations of a localized overlay-only approach.

B. Content Delivery

A few prior studies have proposed new locality-aware techniques for content delivery in live P2P streaming. The works in [34]–[37] focus on the deployment of additional infrastructure across multiple ISPs for separation of the delivery of content in inter-ISP and intra-ISP overlays. These solutions follow hybrid or peer-assisted approaches that rely on abundant streaming resources provided by the ISP or the P2P system such as high-bandwidth servers and amplifiers. In our work, however, we focus on the more general class of P2P streaming systems in which streaming capacities are provided mainly by the participating peers.

A recent study by Y. Chan [38] proposes a tree-mesh solution in which stable peers with higher upload bandwidth are promoted to become group heads. Each of the heads receives the stream through a tree spanning those elected peers only, and further distributes the content in its corresponding group through a mesh. In contrast, we do not rely on any extra resources or special peers in OLIVES but assume all peers have limited upload bandwidth. Furthermore, we follow a mesh-based rather than a tree-based approach; thus, OLIVES avoids the classical problems of tree construction and maintenance, scalability and inability to cope with bandwidth dynamics and fluctuations.

Picconi et al. [7] propose a scheme for ISP-friendly mesh-based live streaming based on an adaptive technique in which each peer maintains two sets of local (i.e., primary) and external (i.e., secondary) neighbors. The rate of delivery from each set is adjusted based on state of the local buffer; external links are dynamically unchoked in response to missing content in the buffer. While this technique reduces inter-ISP traffic, it does not provide any deterministic bound. In fact, the inter-ISP traffic depends on factors such as network conditions, number of ISPs, peer population, bandwidth distribution in each ISP and buffer size. In contrast, OLIVES reduces the inter-ISP traffic in similar settings by more than 70% of the reported values in [7].

Tomozei et al. [8] suggest using network coding for live P2P streaming to minimize the redundant inter-ISP traffic. Through analysis the authors showed that by incorporating network coding their distributed rate-allocation algorithm can achieve near-optimal inter-ISP traffic reduction in static settings. However, the proposed algorithm has a very slow adaptation rate in scenarios that exhibit network dynamics (e.g., bandwidth fluctuations or peer churn).

X. Conclusion

In this paper, we investigated the design and evaluation of an 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 on the above insights, we designed a new Overlay-aware LIVE P2P Streaming mechanism called OLIVES. OLIVES incorporates a two-tier block scheduling scheme over fully-localized overlays to overcome the constraints imposed by localization. Through detailed simulations we evaluated the performance of OLIVES and demonstrated its ability to achieve good performance over a wide range of realistic scenarios while maximizing traffic localization. We believe our work provides valuable insights in the behavior of P2P streaming applications over localized overlays.

References

[1] D. P. T Karagiannis, P. Rodriguez, “Should internet service providers fear peer-assisted content distribution?” in *IMC*, 2005.

[2] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, “A measurement study of a large-scale p2p iptv system,” *TOM*, 2007.

[3] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, “P4p: Provider portal for (p2p) applications,” in *SIGCOMM*, 2008.

[4] V. Aggarwal, S. Bender, A. Feldmann, and A. Wichmann, “Can isps and p2p users cooperate for improved performance?” *SIGCOMM CCR*, 2007.

[5] X. Zhang, J. Liu, B. Li, and T. Yum, “Coolstreaming: A data-driven overlay network for live media streaming,” in *INFOCOM*, 2005.

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

[7] F. Picconi and L. Massouliè, “Isp-friend or foe? making p2p live streaming isp-aware,” in *ICDCS*, 2009.

[8] D. Tomozei and L. Massouliè, “Flow control for cost-efficient peer-to-peer streaming,” in *INFOCOM*, 2010.

[9] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, “Chainsaw: Eliminating Trees from Overlay Multicast,” in *IPTPS*, 2005.

[10] R. Rejaie, M. Handley, and D. Estrin, “RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet,” in *INFOCOM*, 1999.

[11] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicaqt applications,” in *Proceedings of the ACM SIGCOMM*, 2000.

[12] T. Bonald, L. Massouliè, F. Mathieu, D. Perino, and A. Twigg, “Epidemic live streaming: Optimal performance trade-offs,” in *SIGMETRICS*, 2008.

[13] Y. Zhou, D. Chiu, and J. Lui, “A simple model for analyzing p2p streaming protocols,” in *ICNP*, 2007.

[14] M. Zhang, Q. Zhang, and S. Yang, “Understanding the power of pull-based streaming protocol: Can we do better?” *JSAC*, 2007.

[15] Y. Liu, L. Guo, F. Li, and S. Chen, “A Case Study of Traffic Locality in Internet P2P Live Streaming Systems,” in *ICDCS*, 2009.

[16] C. Wu, B. Li, and S. Zhao, “Exploring large-scale peer-to-peer live streaming topologies,” in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2008.

[17] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, “Improving traffic locality in bittorrent via biased neighbor selection,” in *ICDCS*, 2006.

[18] D. R. Choffnes and F. E. Bustamante, “Taming the torrent: A practical approach to reducing cross-isp traffic in peer-to-peer systems,” in *SIGCOMM*, 2008.

[19] F. Wang, “Stable peers: Existence, importance, and application in peer-to-peer live video streaming,” in *INFOCOM*, 2008.

[20] N. Magharei, R. Rejaie, V. Hilt, I. Rimac, and M. Hofmann, “ISP-Friendly Live P2P Streaming,” Tech. Rep. CIS-TR-09-07, 2009. [Online]. Available: http://mirage.cs.uoregon.edu/pub/tr09-07.pdf

[21] C. Huang, J. Li, and K. W. Ross, “Can internet video-on-demand be profitable?” in *SIGCOMM*, 2007.

[22] A. Medina, A. Lakhina, I. Matta, and J. Byers, “BRITE: An Approach to Universal Topology Generation,” in *MASCOTS*, 2001.

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

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

[25] J. Li and K. Sollins, “Exploiting autonomous system information in structured peer-to-peer networks,” in *ICCCN*, 2004.

[26] A. Nakao, L. Peterson, and A. Bavier, “A routing underlay for overlay networks,” in *ACM Sigcomm*, 2003.

[27] F. Dabek, R. Cox, F. Kaahoek, and R. Morris, “Vivaldi: A decentralized network coordinate system,” in *Proceedings of the ACM SIGCOMM*, 2004.

[28] F. Pianese and D. Perino, “Resource and locality awareness in an incentive-based p2p live streaming system,” in *Workshop on Peer-to-peer streaming and IP-TV*, 2007.

[29] X. Jin, “Network aware p2p multimedia streaming: Capacity or locality?” in *IEEE Conference on Peer-to-Peer Computing*, 2011.

[30] M. Alhaisoni, M. Ghanbari, and A. Liotta, “Scalable p2p video streaming,” *International Journal of Business Data Communications and Networking*, 2010.

[31] F. Lehrieder, S. Oechsner, T. Hobfeld, D. Staehle, Z. Despotovic, W. Kellerer, and M. Michel, “Mitigating unfairness in locality-aware peer-to-peer networks,” *Journal of Network Management*, 2011.

[32] S. Ren, E. Tan, T. Luo, L. Guo, S. Chen, and X. Zhang, “Topbt: A topology-aware and infrastructure-independent bittorrent client,” in *INFOCOM*, 2010.

[33] 2011. [Online]. Available: https://datatracker.ietf.org/wg/alto/charter/

[34] J. Zhao and C. Wu, “Characterizing locality-aware p2p streaming,” *Journal of Communications*, 2012.

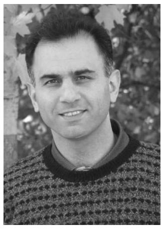
[35] J. Stern, O. Luzzatti, R. Goldberg, E. Weiss, and M. Gonen, “An optimal topology for a static p2p live streaming network with limited resources,” in *IEEE Conference on Parallel and Distributed Systems (ICPADS)*, 2011.

[36] M. Masoud, X. Hei, and W. Cheng, “Constructing a locality-aware isp-friendly peer-to-peer live streaming architecture,” in *Conference on Information Science and Technology (ICIST)*, 2012.

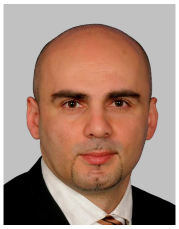
[37] C. Wu, B. Li, and S. Zhao, “On dynamic server provisioning in multichannel p2p live streaming,” *IEEE/ACM Transactions on Networking*, 2011.

[38] Y.-W. Chan, “On the design of a contribution-based, flexible locality-aware p2p streaming network,” *Journal of Networks*, 2011.

Nazanin Magharei is a research scientist at Georgia Institute of Technology. Dr. Magharei joined Georgia Institute of Technology as a postdoctoral researcher in 2011. She obtained her Ph.D. degree from University of Oregon in Computer Science in 2010 and her B.S. degree in Electrical Engineering from Sharif University of Technology, in 2002. Her research interests include Home-networking, Internet measurement, Peer-to-Peer streaming and overlay characterization.

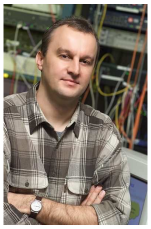


Reza Rejaie is currently an Associate Professor at the University of Oregon. From 1999 to 2002, he was a Senior Technical Staff member at AT&T Labs—Research in Menlo Park, California. He received a NSF CAREER Award for his work on Peer-to-Peer streaming in 2005. Reza has also been the recipient of the European Union Marie Curie Fellowship in 2009. He received his M.S. and Ph.D. degrees from the University of Southern California in 1996 and 1999, and his B.S. degree from the Sharif University of Technology in 1991. Reza has been a senior member of both the ACM and IEEE since 2006.



Ivica Rimac is currently a senior research member of Networked Services Research at Bell Labs, the research organization of Alcatel-Lucent. Ivica joined Bell Labs in 2005 after obtaining a Ph.D. degree in electrical engineering and information technology from Darmstadt University of Technology, Germany. His field of research is computer networking and distributed systems where he has co-authored numerous papers and patents among others in the areas of content networking and peer-to-peer networks.

Volker Hilt is the head of Networked Services Research at Bell Labs/Alcatel-Lucent in Stuttgart, Germany. Dr. Hilt received his master’s degree in Information Systems in 1996 and his Ph.D. in Computer Science in 2001 both from the University of Mannheim in Germany. His field of research is computer networking where he has made contributions in the areas of cloud computing technologies, content distribution networks, peer-to-peer applications, distributed multimedia systems and the Session Initiation Protocol (SIP). Dr. Hilt is a contributor to the Internet Engineering Task Force (IETF) and chairs the SIP Overload Control Working Group. He has published over 50 papers and co-authored more than 15 Internet drafts and RFCs.



Markus Hofmann is the Head of Bell Labs Research, the research organization of Alcatel-Lucent. Dr. Hofmann is known for his pioneering work on reliable multicasting over the Internet and for defining and shaping fundamental principles of content networking. Dr. Hofmann has been active in several professional organizations. He has served as Chair of the Internet Technical Committee (ITC), as Chair of the Open Pluggable Edge Services (OPES) Working Group in the Internet Engineering Task Force (IETF), on the Editorial Board of the Computer Communications Journal and the IEEE/ACM Transactions on Networking. He received his Ph.D. with honors in Computer Engineering from University of Karlsruhe, Germany, in 1998 and joined Bell Labs Research the same year.