Noname manuscript No.
(will be inserted by the editor)

On Performance Evaluation of Swarm-based Live Peer-to-Peer Streaming Applications

Reza Rejaie · Nazanin Magharei

Received: Oct 27, 2012 / Accepted: ?

Abstract During recent years, swarm-based Peer-to-Peer streaming (SPS) mechanisms have become increasingly popular for scalable delivery of live streams over the Internet. The performance of SPS mechanisms depends on the overall effect of several factors including the connectivity of the overlay, the details of packet scheduling scheme and environment settings (*e.g.*, peer and source bandwidth). Prior studies often presented overall performance of their proposed techniques in terms of delivered quality to all peers at a particular setting without demonstrating their inherent performance bottlenecks. Therefore, it it difficult to determine whether and how the reported performance of a SPS mechanism might change as a function of available resources or the connectivity of the overlay.

In this paper, we present a simple yet effective methodology for performance evaluation of SPS mechanisms. Our methodology leverages an organized view of an overlay coupled with a two-phase notion of content delivery in SPS mechanisms to derive a set of metrics that collectively capture the behavior of each phase of content delivery. Therefore, the collection of our metrics can be viewed as the ”signature of content delivery” of a given SPS mechanism. We also present the signature of a well-performing SPS mechanism that can be used as a reference for assessment of other mechanisms. To demonstrate the ability of our proposed evaluation methodology in identifying performance bottlenecks of SPS mechanisms and their underlying causes, we conduct two case studies: *(i)* assessing the performance of a set of candidate packet scheduling schemes, and *(ii)* examining the effect of overlay localization on the performance of SPS mechanisms. In addition to illustrating the use of our methodology through examples, our case studies shed an insightful light on the performance bottlenecks in our target scenarios.

Keywords P2P Streaming · Performance Evaluation · Swarming

University of Oregon
Reza Rejaie
E-mail: reza@cs.uoregon.edu

Nazanin Magharei
E-mail: nazanin@cs.uoregon.edu

1 Introduction

Swarm-based Peer-to-Peer streaming (SPS) mechanisms offer a promising approach for scalable streaming of live content over the Internet. In SPS mechanisms, participating peers typically form a randomly connected overlay over which they incorporate swarming (*i.e.*, pull-based) content delivery. The key component of swarming content delivery is a *packet scheduling* scheme at individual peers that determines which subset of packets should be pulled from each one of neighbors. A well-designed packet scheduling scheme enables participating peers to effectively contribute their resources (*i.e.*, outgoing bandwidth) which in turn improves the utilization of available resources among peers and leads to a better scaling property compared to the traditional tree-based approaches to Peer-to-Peer streaming of live content [1, 4, 7, 8, 24].

The performance of any SPS mechanism is often assessed by the quality of delivered stream to individual peers that depends on the combined effect of the following three factors: *(i)* details of the packet scheduling scheme, *(ii)* connectivity structure of the overlay, and *(iii)* environment settings such as available resources (source and peer bandwidth), heterogeneity of bandwidth among peers, and the dynamics of peer participation (*i.e.*, churn). More importantly, one of these factors may compensate or cancel the effect of others on the overall performance of a SPS mechanism. For example, a poorly designed packet scheduling scheme may exhibit good performance if plenty of excess resources are available in the system [23]. Given the subtle effect of these factors on the overall performance, dissecting the behavior of swarming content delivery in order to identify the underlying performance bottleneck(s) is rather challenging and requires a deep understanding of SPS mechanisms.

A majority of prior studies on SPS mechanisms during the past decade have primarily focused on delivered quality to participating peers to evaluate the overall performance (*e.g.*, [13, 20, 22, 25]). However, to our knowledge, these studies have not examined the underlying factors that may limit the observed performance of their proposed mechanisms. On the other hand, theoretical analysis of content delivery in SPS mechanism (*e.g.*, [1, 23]) present performance limits for some SPS mechanism but do not usually incorporate the dynamics of peer participation or bandwidth heterogeneity and their potential impacts on performance in practice. In the absence of a clear evaluation methodology, it is difficult to properly identify the inherent performance bottlenecks of a given SPS mechanisms in order to conduct a meaningful comparison among different mechanisms or improve their performance.

This paper presents a new approach for detailed performance evaluation of SPS mechanisms. Toward this end, we makes the following two contributions: First, we present a new evaluation methodology to clearly assess the performance of a given SPS mechanism and identify its main performance bottleneck(s). Our methodology builds on the following important concepts that we presented in our earlier study on the design of SPS mechanisms for live video [7]: *(i)* the notion of organized view of a randomly connected overlay that facilitates the analysis of content delivery, and *(ii)* the two-phase pattern of content delivery that consists of a diffusion and swarming phase over the organized view of an overlay. These two concepts motivate a set of metrics for our methodology to effectively capture the behavior content delivery during the diffusion and swarming phases. Furthermore, the collection of these metrics for a given SPS mechanism can be viewed as its ”signature”. We also infer the signature of a well-performing SPS mechanism that can be used as a reference for assessing the behavior of a new mechanism.

Second, we conduct two case studies to demonstrate the ability of our proposed methodology in identifying performance bottlenecks of SPS mechanismsas follows: *(i)* we dissect the performance of a collection of candidate scheduling schemes that represent its main design spectrum in both resource-constrained and resource-rich settings. We show the inherent performance bottlenecks of the scheduling schemes that do not explicitly prioritize newly generated packets. We further demonstrate that by adding the sufficient amount of excess peer and source bandwidth, any poorly-designed scheduling scheme may exhibit good performance. *(ii)* we evaluate the effect of overlay localization on the overall performance of a well-designed packet scheduling scheme. Our findings illustrate that localizing the overlay connectivity beyond a certain level significantly degrades the performance of a SPS mechanism.

These case studies not only demonstrate the ability of our methodology to dissect the behavior of SPS mechanism in action but they also shed an insightful light on the dynamics of content delivery and its fundamental limitations in our target scenarios. We believe that our proposed methodology offers a simple yet powerful approach to meaningfully evaluate and compare different SPS mechanisms.

The rest of this paper is organized as follows: In Section 2, we provide the required background for the rest of the paper by presenting the main components of SPS mechanisms, the organized view of an overlay and an overview of the two-phase content delivery in SPS mechanisms. Section 3 presents our evaluation methodology which includes our key metrics, the signature of a well-behaved content delivery mechanisms and our simulation settings. We present our first case study in Section 4 by examining a group of candidate scheduling schemes and leveraging our methodology to evaluate their performance over a randomly connected and dynamic overlay in both resource-constrained and resource-rich settings. Section 5 describes our second case study where we evaluate the performance of a SPS mechanism over localized overlays for ISP-friendly content delivery using our methodology. Finally, Section 6 concludes the paper.

2 Swarm-based P2P Streaming: An Overview

In this section, we present the main components of SPS mechanisms, their main design parameters and basic dynamics of swarming content delivery for live video that collectively serve as the required background for our study. Any SPS mechanism consists of the following two basic components: *Overlay Construction* and *Content Delivery*. We will briefly describe each component and elaborate their design choices.

2.1 Overlay Construction

In SPS mechanisms, participating peers maintain a randomly connected overlay (*i.e.*, random overlay). The connection between each pair of peers could be directed or undirected (*i.e.*, content can be delivered in one or both directions). To consider a general case, we assume that peers form a directed overlay where there exists a parent-child relationship between connected peers. Peers learn about a subset of participating peers through a central or distributed peer discovery mechanism. The target number of parents and children that each peer tries to maintain is proportional to its incoming and outgoing access link bandwidth, respectively. All connections are congestion controlled using RAP [17] or TFRC [3]. Therefore, maintaining a similar bandwidth-degree ratio ensures that all connections have roughly the same average bandwidth and thus minimizes the possibility of bandwidth bottleneck on the access links [7, 9, 12].

2.2 Content Delivery

Participating peers perform swarming content delivery over the constructed overlay. Toward this end, each peer progressively reports its available packets to its children. Knowing all the available packets among its parents, each peer implements a *packet scheduling* scheme that periodically (once per Δ seconds) determines a subset of packets that should be requested (*i.e.*, pulled) from each parent. The collective behavior of packet scheduling scheme across all participating peers determines the global pattern of delivery for individual packets from source to all participating peers through the overlay [7].

We assume that the content is coded using Multiple Description Coding (MDC) and all descriptions have a similar constant rate. While MDC encoding is not a requirement for SPS mechanism, it accommodates the heterogeneity of access link bandwidth among peers as the number of delivered descriptions to individual peers is proportional to their incoming access link bandwidth [4, 7]. In the context of live P2P streaming applications, source progressively generates new $segment$ of content once every Δ seconds. A segment consists of a collection of packets with consecutive timestamps ($t_{src} - \Delta$, t_{src}) across all descriptions where t_{src} denotes source's playout time. All peers maintain a loosely synchronized playout time (t_p) that is ω seconds behind the playout time of live source ($t_{src} - \omega = t_p$). This implies that each peer requires to maintain at least ω seconds worth of buffering to accommodate out-of-order delivery of packets that are required for swarming. Maintaining a close playout time maximizes the overlap across buffered packets among all peers which in turn accommodates swarming content delivery. Figure 1 depicts a snapshot of ω second buffer at individual peers for a stream with five descriptions. We divide the buffer into smaller sub-window of Δ seconds. Each Δ seconds, the playout time (t_p) progresses by Δ seconds (*i.e.*, peers play Δ seconds worth of packets) and individual peers slide their buffer forward by Δ seconds. Therefore, the first sub-window (t_p, t_p+Δ) consists of packets that are being played in the current interval. The second sub-window (t_p+Δ, t_p+2*Δ) is called *playing region* and consists of packets that are played during the next interval. Finally, the last sub-window (t_{last},t_{new}) consists of packets that have become available among parents since their last report but have not been requested yet. We refer to this sub-window as *new region*. The area in the buffer between the playing and new regions consists of packets that can be used for swarm-

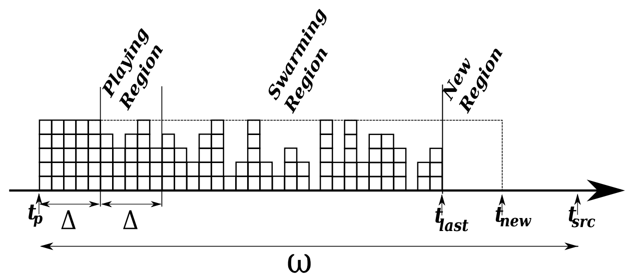


Fig. 1 A snapshot of peer buffer with different regions, assuming a MDC stream with five descriptions

Parameter	Description
t_p	Peer's playout time.
t_{src}	Source's playout time.
t_{last}	Largest available timestamp in the peer.
t_{new}	Largest reported timestamp by parents.
ω	Total duration of buffer at each peer.
Δ	Period of scheduling events.

Table 1 Description of parameters in Figure 1

.

ing. We refer to this area as *swarming region*. Table 1 summarizes all the parameters in Figure 1. For further details on design issues, trade-off and parameters related to SPS mechanisms, we refer interested readers to our earlier work [7].

2.3 Global Pattern of Delivery

The performance of a SPS mechanism (*i.e.*, delivered quality to individual peers) clearly depends on three different parameters: the connectivity structure of the overlay, the scheduling scheme, and resource availability, namely source and peer bandwidth. It is essential to determine whether and how each one of these factors may affect the overall performance of a SPS mechanism. However, this is a challenging goal as these parameters may cancel each other's impact. For example, some scheduling schemes clearly exhibit poor performance in a resource-constrained setting but increasing resource could dramatically improve their performance [16, 23]. To identify the separate impact of these key parameters on the performance of SPS mechanisms, a clear understanding of the global pattern of delivery for individual packets and its relationship with overlay connectivity and scheduling schemes is essential. To achieve this goal, we leverage two concepts that we developed in our earlier study [7]: *(i)* an *organized view* of a directed overlay, and *(ii)* the *two-phase pattern* of content delivery that maximizes the delivered quality to individual peers. This will motivate our evaluation metrics and our methodology to dissect the behavior of swarming content delivery that we present next.

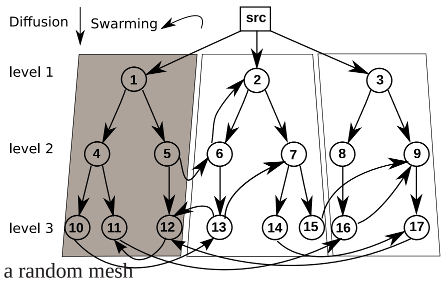


Fig. 2 Organized view of a random mesh

Organized View of an Overlay: In a directed and randomly connected overlay, peers can be organized into *levels* based on their shortest distance (in hops) from source through the overlay. Figure 2 shows an organized view of an overlay with 17 peers in three levels. For example, in Figure 2 peers 1-3 are in level one, peers 4-9 are in level two and the rest of the peers are in the third level. Typically, a peer with degree d in level i has one parent in level $(i-1)$ that we call *diffusion parent*, and $d-1$ parents in the same or lower levels that we call *swarming parents*. In Figure 2, any connection from a diffusion parent (called diffusion connection) is shown with a straight arrow whereas any connection from a swarming parent (called swarming connection) is shown with a curvy arrow. For example, peer 12 in Figure 2 has peer 5 as its diffusion parent and peers 13 and 17 as its swarming parents.

Two-Phase Content Delivery: Given the live nature of source in SPS mechanisms, there is only a limited window (ω seconds) worth of packets for swarming among peers. For effective swarming content delivery, it is essential to utilize outgoing bandwidth of all participating peers which in turn requires each peer to have sufficient amount of packets that are needed by its child peers. This observation motivates the following two-phase pattern of delivery for each packet that maximizes the performance of swarming content delivery.

- *(i) Diffusion Phase*: In this phase, each newly generated packet should be aggressively duplicated as it is pulled away from the source until it reaches the peers at the the bottom level. The main feature of this phase is that the timestamp of pulled packets from the diffusion parent in each period is larger than all the previously pulled packets. Without loss of generality, we assume that source only has sufficient bandwidth to deliver a single copy of each packet to a peer in level 1. For example, when a new packet is pulled by peer 1 (in Figure 2), it is pulled by peers 4 and 5 in the next interval, and then pulled by their child peers, namely peer 10, 11 and 12 in the following interval. Such a collection of peers that are rooted at a peer in level one and receive the same collection of packets during the diffusion phase is called a *diffusion subtree*. Figure 2 marks three diffusion subtrees associated with the three peers in level one. The number of intervals for the diffusion of each packet is equal to the depth of the overlay. A collection of packets that are delivered through a diffusion subtree can be viewed as a *sub-stream*.
- *(ii) Swarming Phase*: At the end of a diffusion phase of a packet, that packet is available among all peers at the bottom of the corresponding diffusion subtree. During the swarming phase, each peer pulls the substreams associated with other

diffusion subtrees from its swarming parents. A peer with degree d usually has $d-1$ swarming parents. If swarming parents of a peer are located on distinct diffusion subtrees (*e.g.*, peer 12 in Figure 2), then it can receive other substreams in one interval, *i.e.*, swarming phase takes only one intervals. However, on a randomly connected overlay, some peers are likely to have more than one swarming parent on each diffusion subtree. Using both modeling [24] and simulation [7], previous studies demonstrated that over a randomly connected overlay, the maximum number of swarming intervals is three. Since the buffer size should accommodate worst case scenario, the smallest buffer size for each peer in a SPS mechanism with live content is $(depth + 3) * \Delta$ where $depth$ denotes the number of levels in the overlay. Note that the overlay depth and thus buffer size is a logarithmic function of peer population (*i.e.*, $depth$ = $log_{d-1}(pop)$ where d and pop denotes peer in-degree and peer population [7]. SPS mechanisms can operate with a larger buffer size but it leads to a larger gap between source's and peer's playout time.

The organized view of the overlay simplifies the distinction between connections that are used for diffusion vs swarming phases. Any connection between a peer in level i and level $i+1$ is used for diffusion (straight arrows in Figure 2) whereas the rest of connections are used for swarming. The two-phase pattern for delivery leads to the most efficient content delivery for SPS mechanisms. As we showed in our previous study [7], the *shortest path scheduling* scheme leads to the above global pattern of content delivery. Shortest path scheduling is essentially the same as most-recent-first packet scheduling that is known under different names in literature [1, 24, 25]. One way to implement the shortest path scheduling scheme is to include a hop count in each packet that captures the number of peers (hops) that the packet has visited. Then, each peer pulls packets with the smallest hop count from each one of its parents.

Visible Horizon: The notion of two-phase content delivery has a clear implication on the range of timestamps for available packets among parents of peers in each level (or visible horizon). The main limitation in the visible horizon for peers at each level is the largest timestamp (*i.e.*, most recent packets) that they observe among their diffusion parent(s). Peers in level 1 are directly connected to the source and thus learn about the availability of new packets at most Δ seconds after their generation. Therefore, peers in level 1 can observe all packets in all $depth$+3 sub-windows across the buffer state. Peers in level 2 learn about the availability of the most recent packets roughly one interval after these packets are pulled by peers in level 1. Therefore, peers in level 2 only observe packets in the left $depth$+2 sub-windows of their parent. Applying the same reasoning, we can infer that peers in level l ($l \leq depth$) of the overlay only observe packets in the left $depth$+3-(l-1) sub-windows of their parents' buffer. Therefore, if peers have the minimum buffer size, peers at the bottom level of an overlay observe available packets only in the left four sub-windows of their parents' buffers. We leverage this key insight to infer the packet availability for peers at each level of the overlay in the next section.

3 Evaluation Methodology

The global pattern of content delivery reveals the main performance bottlenecks of a given SPS mechanism. An effective evaluation methodology should adequately capture and examine the pattern of content delivery. Toward this end, we leverage the notion of two-phase content delivery over the organized view of the overlay and devise a set of metrics to capture the behavior of diffusion and swarming phases of content delivery as follows:

– *Diffusion Rate* of level i presents the rate of arrival of *new* packets to all peers in level i. We consider the average diffusion rate over a sufficiently long time (several windows) to eliminate short-term variations due to congestion controlled connections.
– *Diffusion Time* of a packet to level i is the time that elapses from its generation at source until the first copy of this packet arrives at a peer in level i. Diffusion time of packets to a level shows the variations in the delivery time of individual packets that are not captured by the diffusion rate. The diffusion time of level i is presented as a distribution across all delivered packets to this level. We measure the diffusion time in terms of the number of intervals (Δ) for easy comparison with periodic pulling of packets between peers.
– *Packet Availability* illustrates how effectively swarming can be performed during the swarming phase. Packet availability for peer p represents all the available unique packets among p's (diffusion and swarming) parents. We determine the percentage of available packets in each sub-window of the buffer (as shown in Figure 1) among all parents of individual peers. Then, we average the packet availability in each sub-window across parents of all peers in level i to determine the packet availability for a peer in level i.

3.1 Signature of Content Delivery

The collection of diffusion rate, diffusion times and packet availability for different levels of an overlay can be viewed as a ”signature” of a content delivery mechanism. In this subsection, we derive the signature of a well-performing content delivery mechanism by deriving the value of all of the metrics as follows:

- $DiffusionRate(i)$ = *Stream Rate:* The diffusion rate to *all* levels of the overlay must be equal (or very close) to stream bandwidth. This condition ensures that peers in all levels continuously receive new packets which in turn accommodates effective swarming.
- $DiffusionTime(i) \approx i*\Delta$*:* Given the mis-alignment of packet generation time and pulling intervals, a new packet can be pulled to level 1 within one interval (Δ) after its generation time. Delivery of each packet to each lower level requires at least an additional interval. Therefore, the diffusion time of a packet to level i should take around i intervals. This implies that the minimum required intervals for diffusion of a packet to all levels is $depth$ intervals.
- *Diagonal Packet Availability Across Sub-windows:* We can derive the packet availability in each sub-window of the buffer for peers in level 1 using the notion of

visible horizon that we presented in Section 2. To clearly illustrate our point, we consider an example overlay with 4 levels ($depth$=4) where individual peers and source have out-degree (and in-degree) of 4 (d=4). Typically, a peer has one diffusion parent and $d-1$ swarming parents that are often located at the bottom level. Furthermore, the amount of unique packets that are available through each diffusion sub-tree (*i.e.*, a substream) is equal to 25% of all the packets in each sub-window. We can derive the amount of unique available packets in each sub-window of the buffer among parents of individual peers in level 4 (the bottom level) based on the following observations: First, the visible horizon for each peer in level 4 is the four left sub-windows Each peer can observe 25% of all the packets in these four sub-windows from its diffusion parent (*i.e.*, the sub-stream associated with that diffusion sub-tree). Second, the available packets among the swarming parents depend on their levels in the overlay and the diffusion sub-trees where they are located. We assume that a typical swarming parent is located at the bottom level and consider two scenarios: *(i)* If all swarming parents are located on distinct diffusion sub-trees, they can provide the other three sub-streams with one interval delay. Therefore, in this best case, the peer can observe all the remaining packets in the first three sub-windows as shown with the dotted line in the top plot of Figure 3.1. *(ii)* If swarming parents are not on different diffusion subtrees, the packet availability across the first three sub-windows linearly decreases since packets in earlier sub-windows have more time to be pulled. This worst case scenario is shown with the solid line in the top plot of Figure 3.1. Finally, since different peers slide their windows in a synchronized fashion, some of the remaining packets in the fourth window may also be available among swarming parents. In practice, the average packet availability is some where between the above two cases. To extend the above discussion to derive packet availability among peers in higher levels, we note that the availability among swarming parents for all levels is similar, and the main difference is the visible horizon that increases for peers in higher level. For example, the visible horizon for peers in level 3 and level 2 are five and six left sub-windows, respectively. Therefore, as shown in the corresponding plots in Figure 3.1, the main difference between packet availability for level four compared to level 3 and 2 is the longer visible horizon for these levels which leads to the availability of 25% of packets in the 5th and 6th sub-windows, respectively. Note that the packet availability for level 1 peers is 100% across all sub-windows since they are directly connected to the source which has all the packets.

3.2 Evaluation Plan

We use packet level simulations to demonstrate how our proposed methodology and metrics reveal the performance bottlenecks of a SPS mechanism. Toward this end, we consider two case studies: *(i)* evaluating the performance of a set of candidate scheduling schemes (as we describe in Section 4.1), and *(ii)* examining the effect of localizing overlap connectivity on the performance of SPS mechanisms. Clearly, these exercises shed an insightful light on the behavior of these mechanisms as well. Next, we present our basic simulation setup and our candidate scheduling schemes.

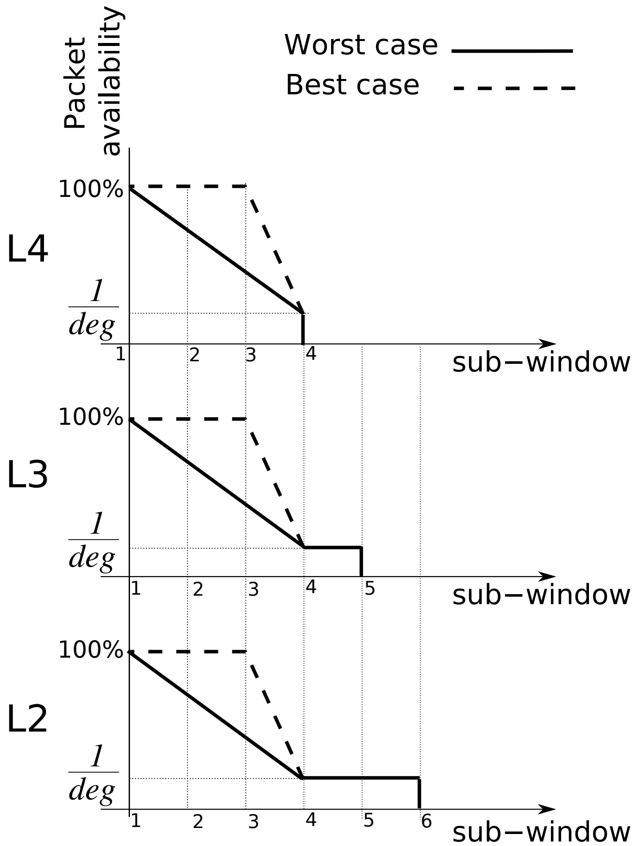


Fig. 3 The inferred best and worst case packet availability with a high-performing scheduling scheme for peers at the bottom 3 levels of the overlay with four levels assuming peers with in- and out-degree of four.

Simulation Setup: Before we present our simulation-based evaluation in the next two sections, we briefly describe our basic setting across our simulations. We use $ns2$ packet-level simulator as it captures packet level dynamics, delay and loss that are essential for the meaningful evaluation of swarming content delivery. In our simulations, physical topology is generated by Brite topology generator [11] with 15 ASes and 10 routers per AS using the top-down mode. Delay on each access link is randomly selected between $[5ms, 25ms]$. Core links have high bandwidth in the range of 4 to 10 Gbps to ensure that congestion occurs at the edge. This in turn avoids any subtle effect of major congestion in the core.

The stream is MDC encoded and has 6 descriptions with the same constant bit rate of 160 Kbps per description. Overlay consists of 500 peers with symmetric access link bandwidth. In our simulations peers may have access link bandwidth of 480 Kbps (3 descriptions) or 960 Kbps (6 descriptions) and maintain 6 or 12 parents (and the same number of child peers), respectively. Source bandwidth is set to 1 Mbps which is the minimum value for delivery of a full-quality stream to peers in level 1. All overlay connections are congestion controlled using RAP [17]. To incorporate a realistic model for churn among participating peers, 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) as reported by empirical studies [18, 19]. The duration of each interval or sub-window (Δ) is set to 6 seconds and the buffer size at each peer is set to its minimum value (ω=($depth$+3)*Δ) where $depth$ denotes depth of an overlay in each setting.

4 Case Study 1: Evaluating Candidate Packet Scheduling Schemes

In this section, we present a set of candidate scheduling schemes and then examine their performance over a dynamic overlay in resource-constrained and resource-rich scenarios as our first case study.

4.1 Candidate Scheduling Schemes

Many scheduling schemes have been proposed by prior studies on SPS mechanisms that have some similar components. Rather than evaluating a few previously-proposed schemes, we describe a set of candidate scheduling schemes that represent the entire design spectrum. This in turn allows us to compare and contrast these schemes in a more structured manner which is more insightful and better illustrates the value of our proposed methodology. The packet scheduling scheme at each peer determines an ordered list of specific missing packets that are requested (pulled) from each parent peer. The key design tradeoff for such a scheme is the relative priority between packets with smaller timestamps to ensure their in-time delivery before their playout time or packets with the largest timestamps to increase the diversity of new packets for effective swarming [7]. As we showed in Figure 1, packets across a peer's buffer can be divided into three regions based on their timestamp. Therefore, in practice, the above design tradeoff can be translated into two basic design choices for a scheduling scheme: *(i)* the relative priority of missing packets in different regions, and *(ii)* the strategy (either *random* or *rarest-first*) to select a subset of missing packets from a region. These two basic dimensions of the design space motivate the following *candidate* scheduling schemes:

- *Rare* or *Rand*: These schemes consider all the missing packets across the entire buffer (t_p+Δ,t_{new}) and use the rarest-first (*e.g.*, [24], [14]) or random (*e.g.*, [2], [13]) strategy for selecting requested packets, respectively. By considering the entire window, these schemes implicitly diversify requested packets. They also implicitly address in-time delivery of packets since packets with smaller timestamps get more opportunity to be selected.
- *PRare* or *PRand*: These schemes explicitly address the timing requirement (*e.g.*, [22]) by first requesting all the missing packets in the playing region, and then using the remaining packet budget to select rare/random packets from the rest of

the window [t_p+2 $*\,\Delta$, t_{new}]. We collectively refer to these two technique using *P** notation.

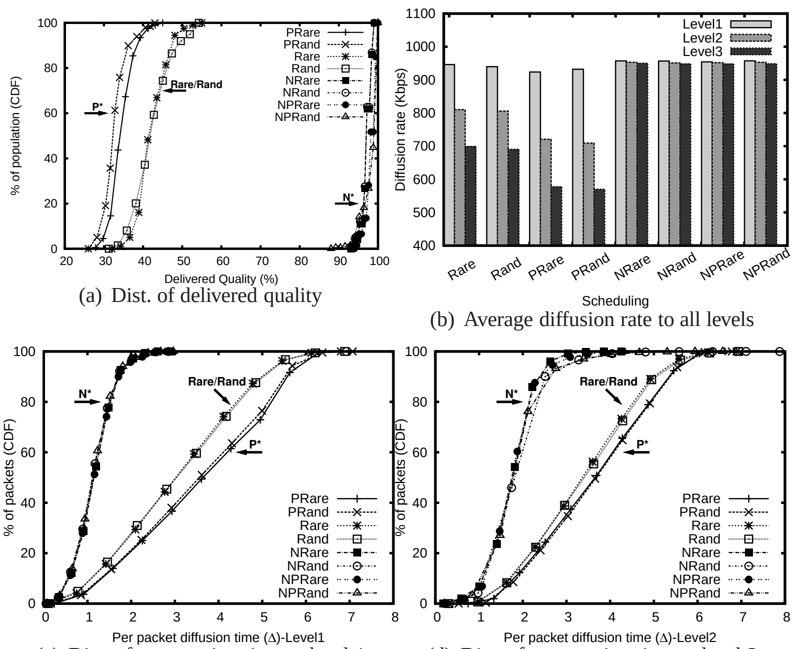
- *NRare* or *NRand*: These schemes explicitly address the diversity of requested packets by first requesting all available packets from the new region and then using the remaining budget to request a rare/random subset of missing packets from the rest of the window [t_p+Δ,t_{last}]. We collectively refer to these two techniques using *N** notation.
- *NPRare* or *NPRand:* These are hybrid schemes [1, 7] that first request the available packets from the new region, then the missing packets from the playing region, and finally use any remaining budget to request a rare/random subset of packets from the swarming region [1]. We collectively refer to these two techniques and NRare and NRand using *N** notation.

4.2 Resource-Constrained Scenario

We first consider the candidate scheduling schemes in a resource constrained scenario where source and peers do not have any excess bandwidth. Figure 4(a) shows the CDF of the delivered quality (as a percentage of maximum quality) to individual peers. This figure reveals that mean delivered quality by *N** schemes is more than 94% but it drops down to 45% and 35% for Rare/Rand and *P** schemes, respectively. To identify the underlying causes for this gap in the observed performance, Figure 4(b) depicts the diffusion rate for the top three levels of the overlay with different schemes. We can observe that all the *N** schemes achieve a high diffusion rate across all levels of the overlay. However, in other four schemes, the diffusion rate drops as we go to lower levels of the overlay. Figures 4(c) and 4(d) present the CDF of diffusion time (in terms of Δ) across all delivered packets to level 1 and level 2 for different schemes, respectively. The distribution of diffusion time for other levels follow a similar pattern and thus they are not shown. Interestingly, these figures illustrate that all the *N** schemes achieve the minimum diffusion time for the majority of packets to each level. The minimum diffusion time for delivery of new packets to the bottom level in *N** schemes provides a longer window for the peers at the bottom level to effectively swarm these packets which in turn leads to a higher delivered quality to all participating peers.

The diffusion time for both *Rare* and *Rand* schemes that purely swarm the packets (in Figures 4(c) and 4(d)) exhibits a uniform distribution across the entire window (all seven intervals) that is rather similar across different levels. This indicates that in *Rare* and *Rand* schemes new packets arrive at each level in a random order, *i.e.*, the new packets are not adequately prioritized for pulling. While all packets arrive at level 1 within 7 intervals (or ω), 10% of packets that arrive during the last interval are late and can not be requested by peers in the lower level. This in turn reduces the diffusion rate to lower levels by an extra 10% as shown in Figure 4(b). In summary, the late arrival of new packets to the top level has a ripple effect on the diffusion rate of other

[1] The other possible hybrid schemes that give higher priority to playing region (*i.e.*, *PNRand* and *PNRare*) have a performance similar to *PRare* and *PRand* and therefore are not considered.



(a) Dist. of delivered quality

(b) Average diffusion rate to all levels

(c) Dist. of propagation time to level 1

(d) Dist. of propagation time to level 2

Fig. 4 Different aspects of performance for different packet scheduling schemes in a resource-constrained setting

levels. Moreover, the diffusion time for only a portion of the delivered packets to the bottom level is short which leaves sufficient time for swarming the packets. This results in a low delivered quality as shown in Figure 4(a).

The *P** schemes perform slightly worse than $Rare/Rand$ schemes. As shown in Figure 4(c), in *P** schemes roughly 20% of packets arrive at the top-level after six intervals which in turn reduces the diffusion rate to the lower level and results in lower delivered quality to all peers. Closer examination of *P** schemes reveals that peers in the top level pull a fraction of their required packets in the playing region from source. These packets are pulled from source around six intervals after their generation time and thus can not be delivered to lower levels, resulting in drop in the diffusion rate of lower levels.

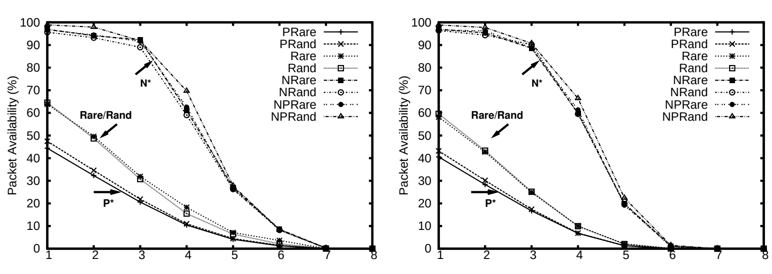
Packet Availability: We now examine the packet availability at each level of the overlay for the candidate scheduling schemes. Figures 5(a) and 5(b) depict the average packet availability across sub-windows of the buffer for peers in level 2 and level 3, respectively. These figures show that only the packet availability for *N** schemes is aligned with the signature of well-behaved content delivery (in terms of visible horizon and diagonal availability across sub-windows) that we presented in Section 3. In particular, peers in level 2 observe $\frac{100}{deg}$ (or 8.3%) of packets among their diffusion parents in sub-window 6 whereas peers in level 3 observe this amount in sub-window 5. This can be easily attributed to the high diffusion rate and low diffusion time for

*N** schemes. Note that for peers in level 3 (Figure 5(b)), the packet availability in sub-window 5 reaches around 24% which is higher than the expected value of $\frac{100}{deg}$ or 8.3%. Our examination revealed that in our simulations a non-negligible fraction of level 3 peers have multiple diffusion parents in level 2 which causes the higher average packet availability in the last sub-window than our model. *N** schemes clearly exhibit a significantly higher packet availability than other schemes especially in left sub-windows for peers at different levels. *Rare/Rand* schemes lead to a slightly better packet availability than *P** schemes. The insufficient diffusion rate coupled with longer diffusion time for *P** and $Rare/Rand$ schemes result in a much smaller slope of increase in packet availability across different sub-window among swarming parents for most peers in the overlay.

4.3 Resource-Rich Scenarios

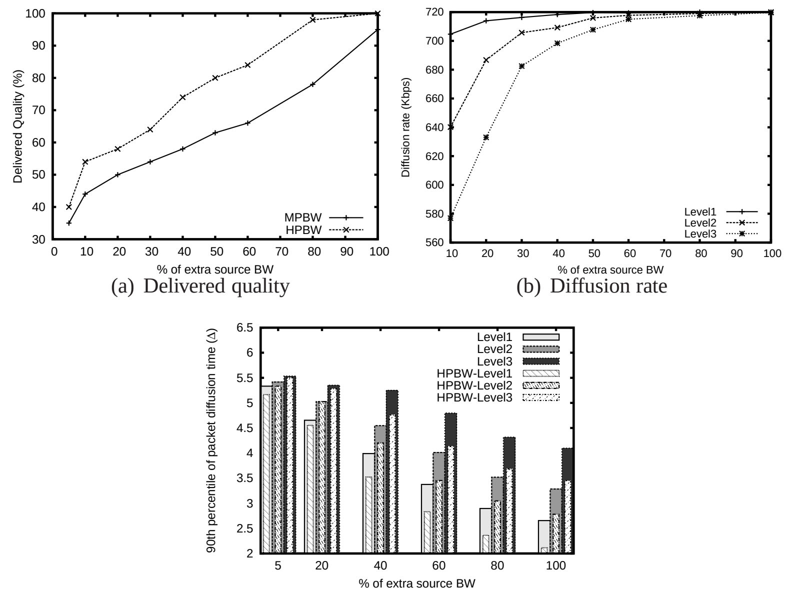
We now turn our attention to resource-rich scenarios and demonstrate the ability of our evaluation metrics to explain the effect of excess resources, namely extra source and peer bandwidth, on the performance of SPS mechanisms. We only discuss *Rare* scheduling scheme that performed poorly in the resource-constrained scenario. *Rand* exhibits a very similar behavior. We consider heterogeneous peers where half of the peers have 960 Kbps and the other half have 460 Kbps symmetric access link bandwidth. All other parameters are set to their default values.

Effect of Excess Source Bandwidth: We increase source bandwidth and adjust its out degree (*i.e.*, number of peers in level 1) accordingly. Peers do not have any excess bandwidth in this scenario and thus we refer to as minimal peer bandwidth or MPBW. The lower line in Figure 6(a) (labeled as MPBW) depicts the 5th percentile of delivered quality to all peers as a function of the extra source bandwidth for *Rare* scheduling scheme. This figure reveals that increasing source bandwidth linearly improves the 5th percentile of delivered quality to individual peers. However, even doubling source bandwidth (*i.e.*, 100% extra source bandwidth) does not lead to the delivery of full quality stream to all peers. To understand the behavior of *Rare* scheduling scheme in this scenario, we examine the diffusion rate and time. Figures 6(b) and 6(c) (only lines labeled *Level**) depict the diffusion rate and 90th percentile of diffusion time



(a) Packet availability for level 2 (b) Packet availability for level 3

Fig. 5 Packet availability for peers in level 2 and level 3 in a resource constrained setting with different packet scheduling schemes



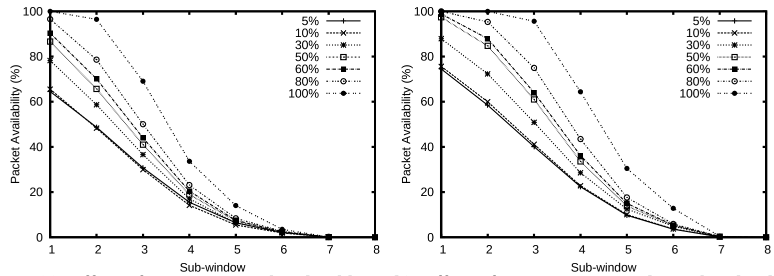
(a) Delivered quality (b) Diffusion rate

(c) Diffusion time

Fig. 6 Effect Performance of *Rare* scheduling scheme in a resource-rich scenario as a function of excess source (and excess peer) bandwidth

across peers in the top 3 levels as a function of extra source bandwidth. Figure 6(b) illustrates that the average diffusion rate to level 2 and 3 rapidly increases with extra source bandwidth until it is saturated around 720 Kbps which is less than the stream bandwidth. Figure 6(c) reveals that increasing source bandwidth monotonically decreases the diffusion time of packets to level 1 which propagates to other levels and decreases their diffusion time as well. However, the diffusion time to all levels even after doubling source bandwidth is still clearly longer than the corresponding diffusion time for *N** schemes. For example, with 60% extra source bandwidth, the 90th percentile diffusion time to level 1 and level 2 peers is 3.5 and 4 intervals whereas with *N** schemes it is 1.7 and 2.7 intervals (as shown in Figures 4(c) and 4(d)), respectively In short, with *Rare* scheme packets become available at each level more than one interval late even with 60% extra source bandwidth. This longer diffusion time limits the window of time for swarming packets and limits the delivered quality despite the major increase in source bandwidth.

Figure 7(a) depicts the packet availability for peers at level 2 as a function of extra source bandwidth. This figure shows that increasing source bandwidth improves packet availability in the left sub-windows and it becomes more similar to the packet availability for *N** schemes. However, we can clearly observe that even with 100% extra source bandwidth, packet availability for *Rare* scheme has roughly one interval shorter visible horizon. For example, only the first sub-window (instead of the first two sub-window for *N** scheme) has 100% packet availability, and the visible horizon is 5 intervals (instead of 6 for *N** schemes). More specifically, despite the addition of



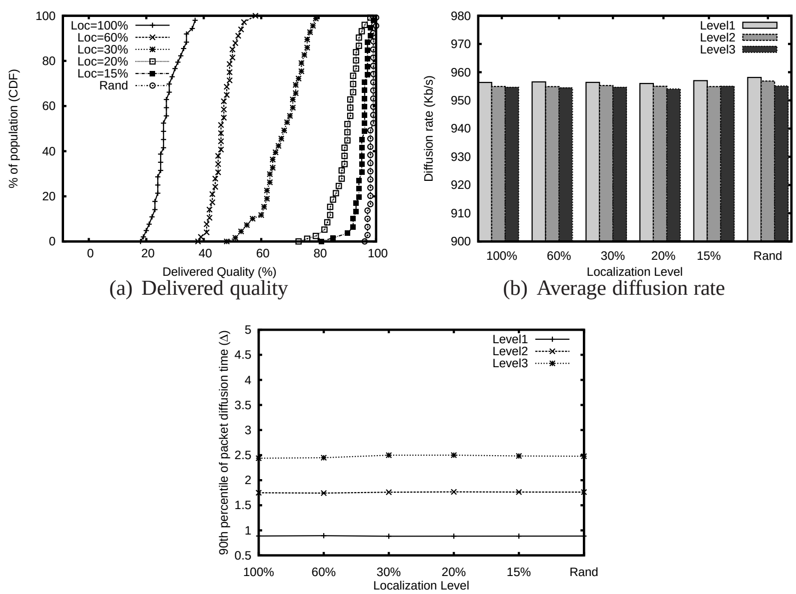
(a) Effect of excess source bandwidth (b) Effect of excess source and peer bandwidth

Fig. 7 Packet availability at level 2 as a function of excess source bandwidth with and without excess peer bandwidth for *Rare* scheduling scheme

excess bandwidth, packets are not pulled to level 1 sufficiently early since the *Rare* strategy does not properly prioritized newly generated packets to ensure sufficiently high diffusion rate and low diffusion time as we showed in Figure 6(c).

Effect of Excess Peer Bandwidth: To investigate the effect of excess peer bandwidth, we double the access link bandwidth of all peers while keeping the source bandwidth at its minimum value of 1 Mbps. Interestingly, doubling peer bandwidth does not have a measurable effect on the diffusion rate, diffusion time or packet availability of all levels. To explain this rather surprising result, we recall that having excess peer bandwidth increases the aggregate bandwidth between levels that is already abundant. But these excess resources do increase the bandwidth from source to level 1. The main performance bottleneck is the inability of the *Rare* scheduling scheme to generate sufficiently high diffusion rate and low diffusion time to level 1 which is not affected by the excess peer bandwidth.

Combined Effect of Excess Source and Peer Bandwidth: In this scenario, we double the bandwidth of all peers and gradually increase extra source bandwidth. The line labeled HPBW in Figure 6(a) depicts the 5th percentile of delivered quality as a function of the extra source bandwidth where all peers have 100% excess bandwidth. Comparing HPBW and MPBW lines in Figure 6(a) demonstrates the effect of excess peer bandwidth for each particular amount of extra source bandwidth. We observe that the combined effect of these excess resources indeed improves the delivered quality. In fact, Figure 6(a) shows that doubling peer bandwidth coupled with 80% extra source bandwidth maximizes the delivered quality for the majority of peers. Figure 6(b) presents the evolution of diffusion rate to different levels as source bandwidth increases for both MPBW and HPBW scenarios since excess peer bandwidth does not have any effect on the diffusion rate. The shaded bars in Figure 6(c) present the 90th percentile of packet diffusion time at each level for the HPBW scenario as a function of extra source bandwidth on top of the corresponding bars for the MPBW scenario. This figure reveals that the excess peer bandwidth causes a larger drop in the diffusion time to all levels as we increase source bandwidth. Figure 7(b) provides a complementary view by showing the average packet availability for peers in level 2 in the HPBW scenario as a function of source bandwidth. This figure clearly illustrates that doubling source and peer bandwidth expands the visible horizon and improves the overall packet availability at level 2 which looks very similar to *N** schemes (shown in Figure 5(a)). The observed improvement in the performance of content delivery as a result of excess resources can be explained as follows: Excess source bandwidth improves the diffusion rate to level 1 and thus overall packet availability at level 1 despite the stated weaknesses of the *Rare* scheduling scheme. Larger peer bandwidth increases the bandwidth between different levels and decreases the diffusion time at lower level which in turn leaves more time for swarming. With a sufficient amount of excess resources, the combined effect of these two factors can sufficiently drop the diffusion time at all levels to the point that accommodates effective swarming and thus good delivered quality. The results of our examinations on other scheduling schemes are available on the related technical report [5].



(a) Delivered quality (b) Average diffusion rate

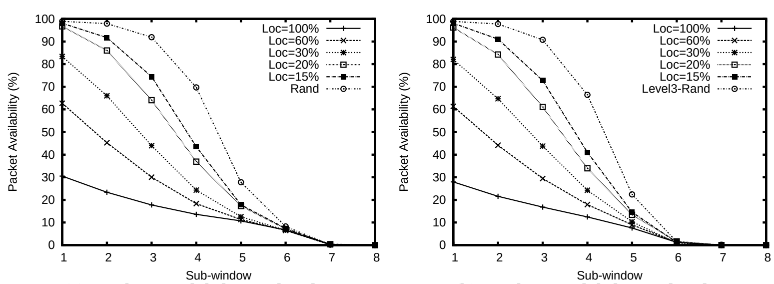
(c) 95th percentile diffusion time

Fig. 8 Effect of overlay localization on the performance of *Rare* packet scheduling

5 Case Study 2: Examining the Effect of Overlay Localization

In recent years, there has been a growing interest in the localization of overlay connectivity within each edge ISP in order to limit the amount of costly inter-ISP traffic [10, 15, 21]. The basic idea is to enable peers within each ISP to connect to each other in order to reduce the number of (incoming and outgoing) inter-ISP connections and thus the associated traffic. Given the popularity of this approach, we examine the effect of overlay localization on the performance of SPS mechanism as our second case study to demonstrate the capabilities of our evaluation methodology and provide useful insight about this approach. Toward this end, we simulate an overlay with 500 peers in a resource-constrained setting where half of peers are high bandwidth. We only focus on the delivered quality to high bandwidth peers since low bandwidth ones have more flexibility to obtain their target quality [7]. In this case study, we only consider one of the best performing packet scheduling scheme over a randomly connected overlay (namely *NPRare*) so that we can associate any change in the performance to the connectivity of overlay structure rather than the shortcomings of the scheduling scheme.

Overlay Generation: To generate an overlay, we consider 10 ISPs and evenly divide all peers among them. We quantify the level of overlay localization as the fraction of all the incoming (or outgoing) connections for all peers in an ISP that are connected to other local peers in the same ISP [10]. For example, 100% overlay localization implies that the number of incoming external connections for each ISP is equal to in-degree of high bandwidth peers (*i.e.*, the minimum incoming bandwidth to receive a full quality stream). At the other extreme, a randomly connected overlay has the lowest level of localization. The level of localization determines the number of external connections for peers in each ISP. We select random half-edges of a random set of peers (*i.e.*, one external half-edge per peer) in each ISP to determine external half-edges for that ISP. All the remaining internal half-edges of peers in each ISP are randomly paired to form internal connections for that ISP. Similarly, the external half-edges of all ISPs are randomly connected to each other and to the source to form the inter-ISP connections in the overlay.



(a) Packet availability at level 2 (b) Packet availability at level 3

Fig. 9 Effect of overlay localization on packet availability at level 2 and level 3 (with *NPRare* scheduling scheme)

Figure 8(a) depicts the distribution of the delivered quality to high bandwidth peers in an overlay with various level of localization including a randomly connected overlay (labeled as ‘Rand‘). This figure suggests that while small level of localization has a negligible effect on the delivered quality, increasing overlay localization beyond 20% significantly degrades the delivered quality to individual peers. To investigate the underlying reasons for the poor performance of content delivery over very localized overlays, we examine the average diffusion rate and 95th percentile diffusion time for the top three levels of the overlay as a function of overlay localization that are shown in Figures 8(b) and 8(c), respectively. These results indicate that all levels have a sufficiently high diffusion rate and low diffusion time (similar to the random overlay) that is not affected by the level of overlay localization. Figures 9(a) and 9(b) depict the packet availability for peers in level 2 and level 3 as we change the level of overlay localization, respectively. These figures demonstrate that peers have proper visible horizon at each level but increasing the level of localization reduces the available content in the left sub-windows which in turn adversely affects the performance of the swarming phase and thus the delivered quality to most peers in the overlay. This raises the following question: ”why is packet availability in left sub-windows low despite the proper diffusion rate and diffusion time for all levels?” A closer examination of available packets to individual peers revealed that as the overlay becomes more localized, a larger fraction of swarming parents for individual peers is likely to be located on the same diffusion subtree. This increases the overlap in the available packets in the left sub-windows of swarming parents and decreases the total available packets to individual peers. To maximize the delivered quality to individual peers, one needs to either carefully change overlay connectivity to add more connections among diffusion subtrees [6] or use an overlay-ware scheduling scheme [10].

6 Conclusions

In this paper, we presented a new methodology for performance evaluation of SPS mechanisms. Leveraging the organized view of an overlay coupled with the notion of two-phase content delivery in SPS mechanisms, we defined a set of metrics that collectively capture the behavior of each phase of content delivery and thus can be viewed as its signature. We illustrated the ability of our methodology to reveal the performance bottleneck(s) of a given SPS mechanism by conducting two detailed case studies: the performance evaluation of several packet scheduling schemes and the examination of the role of overlay localization on the performance of SPS mechanisms. Our case studies also offer a valuable insight in the behavior of targeted SPS mechanisms. Our methodology provides a unified framework for comparing different SPS mechanisms and shed an insightful light on their design.

References

1. Bonald, T., Massouliè, L., Mathieu, F., Perino, D., Twigg, A.: Epidemic live streaming: Optimal performance trade-offs. In: SIGMETRICS (2008)
2. Cohen, B.: Bittorrent. URL http://www.bittorrent.com
3. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicaqt applications. In: Proceedings of the ACM SIGCOMM (2000)
4. Liu, Z., Shen, Y., Ross, K., Panwar, S., Wang, Y.: Layerp2p: Using layered video chunks in p2p live streaming. ieee-tomm **11**(7) (2009)
5. Magharei, N., Rejaie, R.: Dissecting the performance of Live Mesh-based P2P Streaming. Tech. Rep. CIS-TR-07-05, University of Oregon (2007). URL http://mirage.cs.uoregon.edu/pub/tr07-05.pdf
6. Magharei, N., Rejaie, R.: Overlay monitoring and repair in swarm-based peer-to-peer streaming. In: ACM NOSSDAV (2009)

7. Magharei, N., Rejaie, R.: PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming. ACM/IEEE Transactions on Networking (2009)
8. Magharei, N., Rejaie, R., Guo, Y.: Mesh or Multiple-Tree: A Comparative Study of P2P Live Streaming Services. In: INFOCOM (2007)
9. Magharei, N., Rejaie, R., Guo, Y.: Incorporating contribution-awareness into mesh-based peer-to-peer streaming services. Peer-to-Peer Networking and Applications **4** (2011)
10. Magharei, N., Rejaie, R., Hilt, V., Rimac, I., Hofmann, M.: ISP-Friendly Live P2P Streaming. IEEE/ACM Transactions on Networking (2013)
11. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: An Approach to Universal Topology Generation. In: MASCOTS (2001)
12. Padmanabhan, V.N., Wang, H.J., Chou, P.A.: Resilient peer-to-peer streaming. In: ICNP (2003)
13. Pai, V., Kumar, K., Tamilmani, K., Sambamurthy, V., Mohr, A.: Chainsaw: Eliminating Trees from Overlay Multicast. In: IPTPS (2005)
14. Pianese, F., Keller, J., Biersack, E.W.: Pulse, a flexible p2p live streaming system. In: Global Internet Workshop. Barcelona, Spain (2006)
15. Picconi, F., Massouliè, L.: Isp-friend or foe? making p2p live streaming isp-aware. In: ICDCS (2009)
16. Rasti, A., Rejaie, R.: Understanding peer-level performance in bittorrent: A measurement study. In: IEEE Conference on Computer Communications and Networks (ICCCN) (2007)
17. Rejaie, R., Handley, M., Estrin, D.: RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In: INFOCOM (1999)
18. Sripanidkulchai, K., Ganjam, A., Maggs, B.: The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points. In: SIGCOMM (2004)
19. Veloso, E., Almeida, V., Meira, W., Bestavros, A., Jin, S.: A hierarchical characterization of a live streaming media workload. In: IMW (2002)
20. Vlavianos, A., Iliofotou, M., Faloutsos, M.: Bitos; enhancing bittorrent for supporting streaming applications. In: GI (2006)
21. Xie, H., Yang, R., Krishnamurthy, A., Liu, Y., Silberschatz, A.: P4p: Provider portal for (p2p) applications. In: ACM SIGCOMM (2008)
22. Zhang, M., Xiong, Y., Zhang, Q., Yang, S.: On the optimal scheduling for media streaming in data-driven overlay networks. In: GLOBECOM (2006)
23. Zhang, M., Zhang, Q., Yang, S.: Understanding the power of pull-based streaming protocol: Can we do better? JSAC (2007)
24. Zhang, X., Liu, J., Li, B., Yum, T.: Coolstreaming: A data-driven overlay network for live media streaming. In: INFOCOM (2005)
25. Zhou, Y., Chiu, D., Lui, J.: A simple model for analyzing p2p streaming protocols. In: ICNP (2007)